

Санкт-Петербургский государственный университет

**БРОДОВСКАЯ Вероника Вячеславовна**

**Выпускная квалификационная работа**

**Управление с обходом препятствий в вертикальной плоскости  
для автономного подводного аппарата**

Уровень образования: бакалавриат

Направление 01.03.02 «Прикладная математика и информатика»

Основная образовательная программа СВ.5005.2017 «Прикладная математика, фундаментальная информатика и программирование»

Профиль: «Исследование и проектирование систем управления и обработки сигналов»

Научный руководитель:

доцент кафедры компьютерных технологий и систем,

канд. физ.-мат. наук,

Жабко Н.А.

Рецензент:

ст. преп. кафедры компьютерных технологий и систем,

канд. физ.-мат. наук,

Завадский С. В.

Санкт-Петербург

2021

# Содержание

<b>Введение</b> .....	<b>3</b>
<b>Обзор литературы</b> .....	<b>5</b>
<b>1. Содержательная постановка задачи</b> .....	<b>6</b>
<b>2. Математическая постановка задачи</b> .....	<b>9</b>
2.1. Уравнения движения необитаемого подводного аппарата .....	9
2.2. Линеаризация уравнений динамики .....	11
2.3. Декомпозиция общей задачи синтеза .....	13
<b>3. Методы решения задач синтеза</b> .....	<b>15</b>
3.1. Синтез асимптотического наблюдателя в горизонтальной плоскости .....	15
3.2. Синтез скоростного регулятора на базе оценки состояния в горизонтальной плоскости.....	16
3.3. Синтез асимптотического наблюдателя в вертикальной плоскости .....	18
3.4. Синтез пропорционально-дифференциального регулятора по состоянию в вертикальной плоскости.....	19
3.5. Командный уровень на базе сонара .....	23
<b>4. Компьютерное моделирование НПА REMUS</b> .....	<b>26</b>
4.1. Компьютерное моделирование наблюдателя и управления в горизонтальной плоскости.....	26
4.2. Компьютерное моделирование наблюдателя и управления в вертикальной плоскости .....	29
4.3. Моделирование динамической системы в среде Simulink .....	31
4.4. Обход препятствий в вертикальной плоскости .....	34
<b>Выводы</b> .....	<b>38</b>
<b>Заключение</b> .....	<b>39</b>
<b>Список литературы</b> .....	<b>40</b>
<b>Приложение А</b> .....	<b>41</b>

## Введение

Подводная робототехника является одной из новейших областей развития автоматизированных робототехнических комплексов. В последние годы оснащенные широким спектром различных датчиков необитаемые подводные аппараты (НПА), именуемые в зарубежных публикациях термином «Autonomous underwater vehicle» (AUV), широко используются для решения таких задач, как исследование рельефа морского дна, мониторинг и обслуживание трубопроводов и кабельных линий, сбор биологических и геологических проб жидкостей, освоение морских нефтяных и газовых месторождений, проведение подводных аварийно-спасательных работ. Рост популярности такого рода аппаратов объясняется тем, что применение пилотируемых аппаратов может быть весьма опасно для жизни самих пилотов и несет большие эксплуатационные расходы.

Исследования в области моделирования и управления подводными роботами в конечном итоге направлены на достижение полной автоматизации процессов. Достижение данной цели представляется исключительно сложной задачей, так как требует знания и понимания большого объема теоретических и практических проблем, подлежащих всесторонним исследованиям.

В частности, система датчиков (в том числе сонаров) НПА зачастую имеет дело с неизвестной и плохо структурированной окружающей средой, исключающей применение одной только системы GPS (Global Positioning System). Кроме того, применение систем, основанных на компьютерном зрении, не обеспечивает достаточной надежности при наличии плохой видимости под водой.

В данной связи, особенно актуальной представляется проблема синтеза законов управления автономным подводным аппаратом в вертикальной плоскости с целью обхода препятствий и поддержания заданного расстояния от робота до морского дна.

В качестве подобного аппарата для проведения имитационного моделирования, синтеза закона управления и проведения компьютерных экспериментов был выбран аппарат «REMUS 100» – компактный, легковесный, автономный подводный робот, разработанный для проведения операций в прибрежных зонах на глубине до 100 метров. Данный AUV можно конфигурировать, устанавливая на него широкий диапазон стандартного и специализированного оборудования и, тем самым, реализуя необходимые для удовлетворения специфическим требованиям миссии варианты системы.

Данная работа основывается на ранее представленных знаниях об аппарате REMUS, на базе которых в среде MATLAB-Simulink может быть построен имитационный комплекс модели рассматриваемого плавучего объекта.

## Обзор литературы

Книга [1] посвящена рассмотрению современных подходов к моделированию, анализу и синтезу обратных связей для линейных управляемых систем. В книге детально освещены аспекты программной реализации изучаемых вопросов в широко распространенной среде MATLAB-Simulink.

В монографии [2] рассмотрен ряд вопросов и проблем построения моделей оптимальных управляющих устройств для динамических объектов, функционирующих под воздействием внешней среды, и предложен ряд алгоритмов для программной реализации описанных методов синтеза управления.

Моделирование автономных подводных аппаратов, в частности моделирование НПА «REMUS 100» подробно изложены в диссертации [3] и статьях [4-7]. В работах представлены различные аспекты функционирования НПА, определены численные параметры робота и предложен ряд идей для построения управляющих воздействий к рассматриваемому классу подводных аппаратов.

В работах [8] и [9] представлены различные подходы к синтезу законов управления для НПА «REMUS 100», в частности рассматриваются управляющие воздействия как в вертикальной, так и в горизонтальной плоскостях. К недостаткам, пусть и не столь существенным, можно отнести ориентацию преимущественно на линейные модели рассматриваемого аппарата.

Вопросы моделирования движений необитаемых подводных аппаратов, а также различные аспекты синтеза соответствующих им управлений подробно освещаются в книгах [10] и [11]. Кроме того, в работах представлен широкий спектр инструментов анализа НПА и материальная база описания различных свойств и режимов функционирования роботов.

## 1. Содержательная постановка задачи

В данном разделе представлено описание нелинейной механической системы, рассматриваемой в работе в качестве объекта автоматического управления. Здесь на содержательном уровне формулируются все требования, предъявляемые к динамическим процессам, протекающим в замкнутой системе, а также к структуре формируемой обратной связи.

Как уже было сказано ранее, объектом управления в рамках данной работы служит НПА REMUS 100, представленный на рис. 1. Данный аппарат относится к семейству автономных подводных аппаратов Remote Environmental Measuring UnitS, разработанных в США Океанографическим институтом в Вудс-Хоул (серийное производство компании Hydroid). Цифра в названии аппаратов соответствует максимально допустимой глубине погружения в метрах.



Рис.1. Подводный аппарат REMUS 100

В силу известности REMUS 100 и его популярности в коммерческих и военных миссиях, рассматриваемый аппарат широко используется в научной и учебной литературе [3, 4, 5] для проверки новых идей и предлагаемых подходов, связанных с синтезом обратных связей.

Для описания состояния робота, введем в рассмотрение две системы координат (рис.2) – неподвижную прямоугольную систему координат  $O_1XYZ$ , фиксированную на земле, и подвижную прямоугольную систему координат  $O_2xyz$ , закрепленную на аппарате: либо в центре масс, либо центре плавучести.

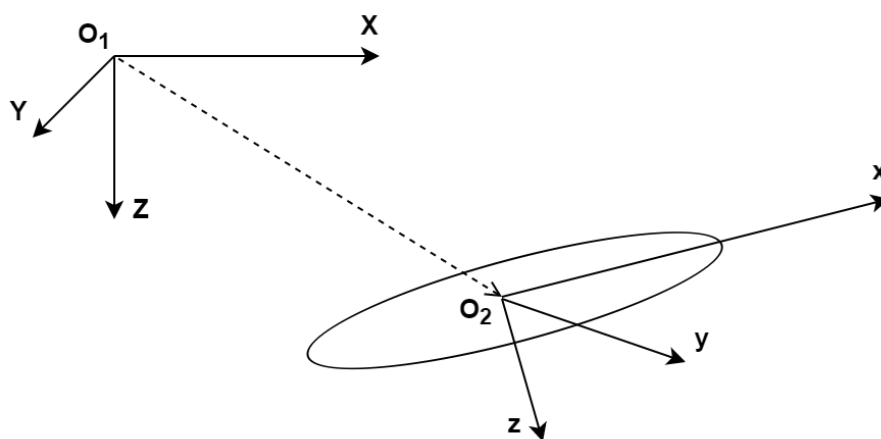


Рис.2. Неподвижная и подвижная системы координат

Для описания движений рассматриваемого объекта, нам также необходимо ввести в рассмотрение следующие переменные, представленные в таблице 1.

Таблица 1.

$u$	Продольная скорость (м/с)	$X$	Положение в направлении $x$ (м)
$v$	Боковая скорость (м/с)	$Y$	Положение в направлении $y$ (м)
$w$	Скорость подъема (м/с)	$Z$	Положение в направлении $z$ (м)
$p$	Скорость крена (рад/с)	$\varphi$	Угол крена (рад)
$q$	Скорость дифферента (рад/с)	$\theta$	Угол дифферента (рад)
$r$	Скорость рыскания (рад/с)	$\psi$	Угол курса (рад)
$\delta_r$	Угол вертикального руля (рад)	$\delta_s$	Угол горизонтального руля (рад)

Все размерности далее представляются в системе СИ.

Будем предполагать, что на работе установлен ряд датчиков (сенсоров), позволяющих в каждый момент времени измерять некоторые из рассмотренных выше переменных, а именно: глубину  $z$  и угол дифферента  $\theta$  в вертикальной плоскости движения аппарата, а также угол курса  $\psi$  в горизонтальной.

Основной целью настоящей работы является синтез многоцелевого закона управления, обеспечивающего маневрирование НПА REMUS 100 в вертикальной плоскости с целью обхода препятствий, возникающих на пути

следования аппарата, а также поддержания заданного расстояния до поверхности дна. При этом существенную роль играет стабилизация робота по курсу в условиях наличия внешних возмущений.

Найденное управление должно удовлетворять конкретным условиям для каждого из базовых режимов функционирования объекта.

1. При собственном движении, определяемом не нулевым начальным отклонением от угла курса, управление должно возвращать робота в нулевое положение относительно заданной траектории движения.

2. При движении, определяемом воздействием ступенчатых возмущений (типа морского течения), обратная связь должна удерживать систему в окрестности нулевого положения по курсу.

3. На базе командного сигнала, основанного на показаниях сонара, управление в вертикальной плоскости должно обеспечивать обход препятствий, встречаемых на пути робота, и поддерживать заданное расстояние до дна.



## 2. Математическая постановка задачи

В данной главе представлены уравнения движения объекта управления и рассмотрен вопрос их линеаризации.

### 2.1. Уравнения движения необитаемого подводного аппарата

В работе движение подводного робота рассматривается в двух плоскостях – в вертикальной и в горизонтальной. При этом основной приоритет отдается обходу препятствий по вертикали и поддержанию заданного расстояния до дна, в то время как в горизонтальной плоскости происходит лишь контроль отклонения от заданного курса и обработка ступенчатых возмущений. Однако для проведения имитационного моделирования и линеаризации уравнений робота в различных плоскостях, необходимо также ввести в рассмотрение и исходную нелинейную модель подводного аппарата.

Так как вывод нелинейных уравнения динамики НПА представляет собой самостоятельную задачу, в рамках данной главы приведем лишь конечный результат таких преобразований.

Имеют место следующие уравнения кинематики рассматриваемого робота:

$$\begin{pmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{pmatrix} = J_1(\varphi, \theta, \psi) \begin{pmatrix} u \\ v \\ w \end{pmatrix}, \quad (1)$$

$$\begin{pmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = J_2(\varphi, \theta, \psi) \begin{pmatrix} p \\ q \\ r \end{pmatrix}, \quad (2)$$

где матрицы принимают вид

$$J_1(\varphi, \theta, \psi) = \begin{pmatrix} \cos(\psi) \cos(\theta) & -\sin(\psi) \cos(\varphi) + \cos(\psi) \sin(\theta) \sin(\varphi) & \sin(\psi) \cos(\varphi) + \cos(\psi) \cos(\theta) \sin(\varphi) \\ \sin(\psi) \cos(\theta) & \cos(\psi) \cos(\varphi) + \sin(\psi) \sin(\theta) \sin(\varphi) & -\cos(\psi) \sin(\varphi) + \sin(\psi) \sin(\theta) \cos(\varphi) \\ -\sin(\theta) & \cos(\theta) \sin(\varphi) & \cos(\theta) \cos(\varphi) \end{pmatrix},$$

$$J_2(\varphi, \theta, \psi) = \begin{bmatrix} 1 & \sin(\varphi) \tan(\theta) & \cos(\varphi) \tan(\theta) \\ 0 & \cos(\varphi) & -\sin(\varphi) \\ 0 & \frac{\sin(\varphi)}{\cos(\theta)} & \frac{\cos(\varphi)}{\cos(\theta)} \end{bmatrix}.$$

Заметим при этом, что матрица  $J_2(\varphi, \theta, \psi)$  не определена для угла дифферента равного  $\pm 90$  градусов. Однако в реальных задачах это не является проблемой, так как характеристики аппарата, как правило, не допускают возможности функционирования в подобном режиме.

Кроме того, имеют следующие уравнения движения данного НПА [3].

1. Волновое уравнение движения:

$$X_f = X_{HS} + X_{u|u}|u|u| + (X_{wq} - m)wq + (X_{qq} + mx_G)q^2 + (X_{vr} + m)vr + (X_{rr} + mx_G)r^2 - my_Gpq - mz_Gpr + X_{prop}.$$

2. Колебательное уравнение движения:

$$Y_f = Y_{HS} + Y_{v|v}|v|v| + Y_{r|r}|r|r| + my_Gr^2 + (Y_{ur} - m)ur + (Y_{wp} + m)wp + (Y_{pq} - mx_G)pq + Y_{uv}uv + my_Gp^2 + mz_Gqr + Y_{uu\delta_r}u^2\delta_r.$$

3. Уравнение движения тяги:

$$Z_f = Z_{HS} + Z_{w|w}|w|w| + Z_{q|q}|q|q| + (Z_{uq} + m)uq + (Z_{vp} - m)vp + (Z_{rp} - mx_G)rp + Z_{uw}uw + mz_G(p^2 + q^2) - my_Grq + Z_{uu\delta_s}u^2\delta_s.$$

4. Уравнение движения по углу крена:

$$K_f = K_{HS} + K_{p|p}|p|p| - (I_{zz} - I_{yy})qr + m(uq - vp) - mz_G(wp - ur) + K_{prop}.$$

5. Уравнение движения по углу дифферента:

$$M_f = M_{HS} + M_{w|w}|w|w| + M_{q|q}|q|q| + (M_{uq} - mx_G)uq + (M_{vp} + mx_G)vp + (M_{rp} - (I_{xx} - I_{zz}))rp + mz_G(vr - wq) + M_{uw}uw + M_{uu\delta_s}u^2\delta_s.$$

6. Уравнение движения по углу курса:

$$N_f = N_{HS} + N_{v|v}|v|v| + N_{r|r}|r|r| + (N_{ur} - mx_G)ur + (N_{wp} + mx_G)wp + (N_{pq} - (I_{yy} - I_{xx}))pq - my_G(vr - wq) + N_{uv}uv + N_{uu\delta_r}u^2\delta_r.$$

Перечисленные выше величины  $X_f, Y_f, Z_f, K_f, M_f, N_f$  являются суммами всех внешних сил, действующих в фиксированном направлении, а все рассматриваемые при этом параметры, не объявленные ранее, представляют собой некоторые численные значения, определяемые на этапе вывода уравне-

ний динамики робота. В рамках настоящей работы данные параметры были взяты из диссертации [3].

Первые три представленных уравнения моделируют поступательные движения подводного аппарата, вторые три – вращательные движения. Все вместе они могут быть представлены в общей матричной форме, более удобной как для их анализа, так для работы, например, в среде Matlab:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 & 0 & mz_G & -my_G \\ 0 & m - Y_{\dot{v}} & 0 & -mz_G & 0 & -Y_{\dot{r}} \\ 0 & 0 & m - Z_{\dot{v}} & my_G & -mx_G - Z_{\dot{q}} & 0 \\ 0 & -mz_G & my_G & I_{xx} - K_{\dot{p}} & 0 & 0 \\ mz_G & 0 & -mx_G - M_{\dot{w}} & 0 & I_{yy} - M_{\dot{q}} & 0 \\ -my_{CG} & mx_G - N_{\dot{v}} & 0 & 0 & 0 & I_{zz} - N_{\dot{r}} \end{bmatrix}^{-1} \begin{bmatrix} X_f \\ Y_f \\ Z_f \\ K_f \\ M_f \\ N_f \end{bmatrix}. \quad (3)$$

Представленные выше уравнения (1), (2) и (3) вкуче представляют полные исходные уравнения динамики аппарата REMUS 100 с вектором состояния 12-ой размерности и двумя управляющими рулями.

## 2.2. Линеаризация уравнений динамики

Работа с исходными нелинейными уравнениями динамики представляет собой крайне сложную задачу, поэтому для последующего анализа поведения рассматриваемого объекта, а также для синтеза стабилизирующих законов управления будем использовать соответствующие линеаризованные модели, свойства и особенности которых могут быть подробно изучены.

На базе рассмотренных выше уравнений динамики представим объект управления как нелинейную систему обыкновенных дифференциальных уравнений вида

$$\dot{x} = F(x, \delta, d), \quad (4)$$

где  $x \in E^{12}$  – вектор состояния,  $\delta \in E^2$  – вектор управляющих воздействий,  $d \in E^l$  – внешнего возмущения, а  $F$  – векторная функция, непрерывно-дифференцируемая по совокупности своих аргументов.

В рамках поставленной задачи нас интересует управление аппаратом только в вертикальной и горизонтальной плоскостях, тем самым, не умаляя

общности, будем предполагать, что на работе уже реализован некоторый закон поддержания постоянной продольной скорости движения  $u$ , которую примем как  $u_0 = 1.51$  м/с, что соответствует стандартному режиму функционирования рассматриваемого аппарата.

С учетом свойств функций  $F$  и характера рассматриваемой задачи линеаризуем систему (4) в окрестности положения робота  $x^*$ , которому соответствуют продольная скорость  $u = u_0$  и нули по остальным переменным состояния и управляющим рулям. В результате имеем линейную систему ОДУ:

$$\dot{x} = A(t)x + B(t)\delta, \quad (5)$$

где матрицы

$$A(t) = \left\{ \frac{\partial F_i}{\partial x_j} \right\}_{x=x^*, \delta=0}, \quad i, j = \overline{1,12};$$

$$B(t) = \left\{ \frac{\partial F_i}{\partial \delta_j} \right\}_{x=x^*, \delta=0}, \quad i = \overline{1,12}, j = \overline{1,2}.$$

Как было отмечено выше, в рассматриваемой задаче вектор состояния системы не подлежит непосредственному измерению полностью, поэтому к системе уравнений (5) добавим уравнение измерения:

$$y = C(t)x. \quad (6)$$

Так как в рассматриваемой задаче частные производные не зависят от  $t$ , то матрицы  $A, B, C$  в уравнениях (5), (6) константные. Таким образом, линеаризованная модель объекта является стационарной (ЛТИ-системой) и имеет вид:

$$\dot{x} = Ax + B\delta, \quad (7)$$

$$y = Cx, \quad (8)$$

Кроме того, в рамках данной работы, будем рассматривать также уравнения, моделирующие привод рулей в линейном варианте, тем самым система (7-8) будет дополнена двумя уравнениями:

$$\dot{\delta}_s = u_s, \quad (9)$$

$$\dot{\delta}_r = u_r. \quad (10)$$

Заметим, что система (7-10) является системой 14-ого порядка, однако в рамках поставленных задач такая размерность является избыточной. В самом деле, рассматривая движения объекта исключительно в вертикальной плоскости, нам достаточно рассматривать лишь те коэффициенты матрицы  $A$ , которые соответствуют скорости подъема  $w$ , скорости дифферента  $q$ , глубине  $z$  и углу дифферента  $\theta$ , и столбец матрицы  $B$ , соответствующий углу вертикального руля  $\delta_s$ .

Аналогичная ситуация обстоит и с движениями робота в горизонтальной плоскости, где пренебрегая скоростью крена  $p$  и углом крена  $\varphi$  для описания процессов достаточно знать боковую скорость  $v$ , скорость рыскания  $r$ , угол курса  $\psi$  и угол горизонтального руля  $\delta_r$ .

С учетом данных замечаний, в дальнейшем будем рассматривать систему (7-10), подразумевая под ней соответствующую систему меньшего порядка, дополнительно оговаривая в какой плоскости происходит движение.

### 2.3. Декомпозиция общей задачи синтеза

С учетом рассмотренных ранее нелинейных уравнений динамики НПА и их линейного приближения, а также сформулированных в рамках содержательной постановки задачи требований к управляющим воздействиям, определим следующий последовательный алгоритм решения задачи синтеза.

1. На базе линейных уравнений (7-10), рассматриваемых в горизонтальной плоскости, необходимо синтезировать асимптотический наблюдатель для рассматриваемой системы и базовый регулятор по состоянию, стабилизирующий НПА на заданном курсе.
2. Предполагая, что в горизонтальной плоскости на аппарат могут действовать внешние возмущения (например, морские течения), необходимо перейти от базового регулятора к скоростному закону управления, обеспечивающему астатизм замкнутой системы по углу курса.

3. По аналогии с первым пунктом на базе линейных уравнений (7-10), но рассматриваемых в вертикальной плоскости, синтезировать асимптотический наблюдатель, восстанавливающий вектор скоростей робота по глубине и углу дифферента.
4. Для линеаризованной в вертикальной плоскости системы необходимо синтезировать пропорционально-дифференциальный регулятор, выводящий аппарат на заданный командный уровень.
5. На базе моделируемого сигнала с сонара необходимо разработать закон управления глубиной с целью обхода видимых препятствий и поддержанием заданного расстояния до дна.

### 3. Методы решения задач синтеза

#### 3.1. Синтез асимптотического наблюдателя в горизонтальной плоскости

Как было отмечено выше, в рассматриваемой модели в канале измерения происходит потеря информации, так как размерность измеряемого сигнала меньше размерности вектора состояния. По этой причине один из распространенных аналитических подходов к синтезу законов управления состоит в том, что на базе получаемого измеренного сигнала  $y(t)$  модели (8) необходимо сформировать асимптотический наблюдатель, который по некоторым правилам строит приближение  $z(t)$  к вектору состояния. Затем полученный вектор оценок используется для формирования управляющего сигнала.

Как было отмечено ранее в рассматриваемом случае синтеза управления в горизонтальной плоскости достаточно знать боковую скорость  $v$ , скорость рыскания  $r$ , угол курса  $\psi$  и угол горизонтального руля  $\delta_r$ . С учетом этого уравнения (7-8) примут вид

$$\dot{x} = Ax + B\delta_r, \quad (11)$$

$$y_r = Cx, \quad (12)$$

где матрица  $A \in E^{3 \times 3}$ , столбец  $B \in E^3$  и строка  $C \in E^3$  соответствуют линеаризованной модели (7-10) рассматриваемого НПА, но обрезанные в соответствии с рассматриваемыми переменными.

Модель асимптотического наблюдателя для системы (10-11) следующая

$$\dot{z} = Az + B\delta_r + L(y_r - Cz), \quad (13)$$

где столбец  $L \in E^3$  выбирается из условия асимптотической сходимости оценки к рассматриваемому вектору состояния, т.е. исходя из гурвицевости матрицы  $A - LC$ .

Для обеспечения устойчивости наблюдателя можно применить различные идеологии, например оптимальную фильтрацию по Калману, биномиальный синтез или распределение полюсов Баттерворта. В нашем рассматриваемом случае остановимся на последнем варианте. Тогда, с учетом размерности системы, собственные значения матрицы  $A - LC$  будут следующими:

$$\begin{aligned} & -\rho, \\ & -\rho * (\cos(\pi/3) + i * \sin(\pi/3)), \\ & -\rho * (\cos(\pi/3) - i * \sin(\pi/3)), \end{aligned}$$

где  $\rho$  – параметр, определяющий степень устойчивости системы.

Динамические процессы при таком распределении собственных чисел имеют более колебательный характер в сравнении с биномиальным синтезом, однако длительность таких процессов, как правило, меньше.

### **3.2. Синтез скоростного регулятора на базе оценки состояния в горизонтальной плоскости**

Обратимся к построению базового управления в виде

$$u_r = K \begin{pmatrix} x \\ \delta_r \end{pmatrix} = K_x x + k_\delta \delta_r,$$

стабилизирующего нулевое положение по углу курса, где  $K$  – числовая матрица, подлежащая поиску. Существует множество регулярных способов решения данной задачи для линейной модели (10, 11), однако в рамках настоящей работы будем использовать синтез линейно-квадратичного (LQR) регулятора.

Напомним, что LQR-регулятор [1], кроме стабилизации замкнутой системы, обеспечивает минимум интегрального квадратичного функционала

$$J = \int_0^{\infty} \left( (x^T, \delta) Q \begin{pmatrix} x \\ \delta \end{pmatrix} + u^T R u + 2(x^T, \delta) N u \right) dt, \quad (14)$$

где  $Q \geq 0, R > 0$  и  $N > 0$  - заданные симметрические матрицы.



Отметим, что рассматриваемый регулятор обеспечивает глобальную асимптотическую устойчивость нулевого положения равновесия линейной системы (10-11), однако, в случае замыкания им исходного нелинейного объекта (1-3), обеспечивается, вообще говоря, лишь локальная асимптотическая устойчивость в некоторой окрестности нуля.

Кроме того, как ранее уже было сказано, предполагается, что на рассматриваемый НПА могут действовать различные внешние возмущения, например, ступенчатые возмущения типа морского течения, т. е. уравнение (11) принимает вид

$$\dot{x} = Ax + B\delta_r + hd,$$

где столбец  $h \in E^3$ , а  $d = d(t)$  – единичная ступенька.

В этой связи, для успешного выполнения поставленной перед регулятором задачи, замкнутая им система должна обладать свойствами астатизма. Известно [2], что подобные свойства в замкнутой системе может обеспечить скоростной регулятор вида

$$u = \mu\dot{x} + \nu y$$

где строка  $\mu$  и число  $\nu$  подлежат поиску.

В ряде же случаев, существует возможность перехода от базового регулятора к его скоростному эквиваленту. Покажем, что в нашем частном случае такой переход возможен.

Можно показать, что при линеаризации исходных уравнений (1-3) и дальнейшем их обрезании до 3-ей размерности в рамках текущей задачи, уравнения (11-12) имеют следующую структуру:

$$\begin{aligned}\dot{x}_1 &= a_{11}x_1 + a_{12}x_2 + b_1\delta, \\ \dot{x}_2 &= a_{21}x_1 + a_{22}x_2 + b_2\delta, \\ \dot{x}_3 &= x_2, \\ \dot{\delta} &= u, \\ y &= x_3 = \psi\end{aligned}$$

Тогда с учетом несложных преобразований [2] имеет место:

$$u = K_x x + k_\delta \delta_r = (K_a S^{-1}, k_2 - K_a S^{-1} M) \dot{x} + k_3 y,$$

где матрицы

$$S = \begin{pmatrix} a_{11} & b_1 \\ a_{21} & b_2 \end{pmatrix}, M = \begin{pmatrix} a_{12} \\ a_{22} \end{pmatrix}, K_x = (k_1 \ k_2 \ k_3), K_a = (k_1 \ k_\delta).$$

В случае отсутствия внешних возмущений, полученный скоростной закон управления будет полностью аналогичен тому, что был найден как решение задачи LQR-синтеза. Однако, ситуация существенно меняется в случае, если возмущения есть, т.к. исходный LQR-регулятор в отличие от скоростного, вообще говоря, астатизм обеспечить не может.

Наконец, в силу того, что на деле производные вектора состояния недоступны для использования в построении управляющего воздействия, заменим их на производные оценки вектора состояния, полученные в предыдущем подразделе. Итого, управление, обеспечивающее асимптотическую устойчивость и астатизм замкнутой линейной системы (11-12) на базе линейного асимптотического наблюдателя (13) имеет вид

$$u_r = \mu \dot{z} + \nu y_r. \quad (15)$$

Заметим, что именно такой закон управления (15), сформированный указанным способом, подлежит непосредственной реализации в контуре обратной связи рассматриваемого НПА.

### **3.3. Синтез асимптотического наблюдателя в вертикальной плоскости**

Обратимся теперь к оценке вектора состояния, состоящего из скорости подъема  $w$ , скорости дифферента  $q$ , глубины  $z$  и угла дифферента  $\theta$ . Постановка текущей задачи и вывод формул полностью аналогичен подразделу 3.1, за тем лишь исключением, что уравнения динамики (11-12) и асимптотический наблюдатель (13) примут вид

$$\dot{x} = Ax + B\delta_s, \quad (16)$$

$$y_s = Cx, \quad (17)$$

$$\dot{z} = Az + B\delta_s + L(y_s - Cz), \quad (18)$$

где матрица  $A \in E^{4 \times 4}$ , столбец  $B \in E^4$  и матрица  $C \in E^{2 \times 4}$  также соответствуют линеаризованной модели (7-10), а матрица  $L \in E^{4 \times 2}$  выбирается исходя из гурвицевости матрицы  $A - LC$ .

Однако в данном случае выбор матрицы  $L$  в уравнении (18) будем осуществляться исходя из идеологии биномиального синтеза так, чтобы выполнялось тождество  $\Delta_s(s) = (s + \rho)^3$ , где  $\Delta_s(s)$  – характеристический полином матрицы  $A - LC$ , а  $\rho$  – требуемая степень устойчивости. Биномиальное распределение, как правило, дает динамические процессы монотонного характера, что особенно важно в рассматриваемой задаче обхода препятствий по вертикали, т.к. излишняя колебательность системы может приводить к критическим последствиям для объекта управления.

### **3.4. Синтез пропорционально-дифференциального регулятора по состоянию в вертикальной плоскости**

Обратимся теперь к построению управляющего воздействия для рассматриваемого НПА в вертикальной плоскости. При этом будем синтезировать управление в виде PD-регулятора

$$\begin{aligned} u_s &= K_p \left( \begin{pmatrix} x \\ \delta_s \end{pmatrix} - \begin{pmatrix} \xi \\ 0 \end{pmatrix} \right) + K_d \left( \begin{pmatrix} \dot{x} \\ \dot{\delta}_s \end{pmatrix} - \begin{pmatrix} \dot{\xi} \\ 0 \end{pmatrix} \right) = \\ &= K_{p_x}(x - \xi) + k_{p_\delta} \delta_s + K_{d_x}(\dot{x} - \dot{\xi}) + k_{d_\delta} \dot{\delta}_s, \end{aligned} \quad (19)$$

где строки  $K_p$  и  $K_d$  подлежат выбору, а  $\xi$  – командный уровень, который, вообще говоря, является функцией времени.

Необходимость добавления дифференциальной части в уравнении (19) обуславливается тем фактом, что в рассматриваемой задаче обхода препятствий по вертикали и поддержания заданного расстояния до дна ключевую роль играет характеристика перерегулирования замкнутой системы. Как будет показано далее в главе 4, использование в контуре обратной связи одной

только пропорциональной составляющей в большинстве случаев не способно обеспечить должный уровень качества управления.

Выбор матрицы  $K_p$  в рамках настоящей работы осуществляется как поиск решения задачи LQR-синтеза (по аналогии с задачей (14), рассмотренной ранее в подразделе 3.2).

Более существенный интерес представляет выбор матрицы  $K_d$  дифференциальной части управления. Но прежде обратимся к формуле (19) и заметим, что с учетом уравнения привода вертикального руля (9) в рассматриваемом законе управления образуется алгебраическая петля:

$$\dot{\delta}_s = u_s = K_{p_x}(x - \xi) + k_{p_\delta}\delta_s + K_{d_x}(\dot{x} - \dot{\xi}) + k_{d_\delta}\dot{\delta}_s.$$

Наличие алгебраической петли в контуре обратной связи негативно сказывается на точности и скорости численного решения уравнений замкнутой системы, так как на каждом шаге интегрирования необходимо решать алгебраическое уравнение относительно неизвестной  $\dot{\delta}_s$ .

Для решения рассматриваемой проблемы перепишем формулу (19), сгруппировав коэффициенты при  $\dot{\delta}_s$  в левой части формулы и разделив на них правую часть:

$$u_s = \dot{\delta}_s = \tilde{K}_{p_x}(x - \xi) + \tilde{k}_{p_\delta}\delta_s + \tilde{K}_{d_x}(\dot{x} - \dot{\xi}), \quad (20)$$

где матрицы

$$\tilde{K}_{p_x} = \frac{1}{1 - k_{d_\delta}}K_{p_x}, \quad \tilde{k}_{p_\delta} = \frac{1}{1 - k_{d_\delta}}k_{p_\delta}, \quad \tilde{K}_{d_x} = \frac{1}{1 - k_{d_\delta}}K_{d_x}.$$

Рассматриваемый регулятор (20) на базе асимптотического наблюдателя (18) должен обеспечивать асимптотическую устойчивость линейной системы (16-17), т. е. собственные значения общей матрицы системы (16-18, 20) должны лежать в левой комплексной полуплоскости.

Формализуем указанное замечание. С этой целью рассмотрим управление (20) на базе асимптотического наблюдателя, полагая, что командный уровень равен нулю:

$$u_s = \tilde{K}_{p_x} z + \tilde{k}_{p_\delta} \delta_s + \tilde{K}_{d_x} \dot{z},$$

что, в силу уравнений (17-18), можно записать как

$$\begin{aligned} u_s &= \tilde{K}_{p_x} z + \tilde{k}_{p_\delta} \delta_s + \tilde{K}_{d_x} (Az + B\delta_s + L(Cx - Cz)) = \\ &= \tilde{K}_{d_x} LCx + (\tilde{K}_{p_x} + \tilde{K}_{d_x} A - \tilde{K}_{d_x} LC)z + (\tilde{k}_{p_\delta} + \tilde{K}_{d_x} B)\delta_s. \end{aligned}$$

Тогда вкупе с уравнениями (16) и (18) имеем следующее уравнение характеристического полинома замкнутой системы:

$$\begin{aligned} \Delta_r(s) &= \det \begin{pmatrix} Es - A & \mathbb{O} & -B \\ -LC & Es - A + LC & -B \\ -\tilde{K}_{d_x} LC & -\tilde{K}_{p_x} - \tilde{K}_{d_x} A + \tilde{K}_{d_x} LC & s - \tilde{k}_{p_\delta} - \tilde{K}_{d_x} B \end{pmatrix} = \\ &= \det \begin{pmatrix} Es - A & Es - A & -B \\ -LC & Es - A & -B \\ -\tilde{K}_{d_x} LC & -\tilde{K}_{p_x} - \tilde{K}_{d_x} A & s - \tilde{k}_{p_\delta} - \tilde{K}_{d_x} B \end{pmatrix} = \\ &= \det \begin{pmatrix} Es - A + LC & 0 & 0 \\ -LC & Es - A & -B \\ -\tilde{K}_{d_x} LC & -\tilde{K}_{p_x} - \tilde{K}_{d_x} A & s - \tilde{k}_{p_\delta} - \tilde{K}_{d_x} B \end{pmatrix}. \end{aligned}$$

Откуда по теореме об определителе блочно-диагональной матрицы следует

$$\Delta_r(s) = \det(Es - A + LC) \det \begin{pmatrix} Es - A & -B \\ -\tilde{K}_{p_x} - \tilde{K}_{d_x} A & s - \tilde{k}_{p_\delta} - \tilde{K}_{d_x} B \end{pmatrix}.$$

Тем самым выбор строки  $K_d$  должен осуществляться исходя из выполнения условий на собственные числа

$$Re s_k \begin{pmatrix} A & B \\ \tilde{K}_{p_x} + \tilde{K}_{d_x} A & \tilde{k}_{p_\delta} + \tilde{K}_{d_x} B \end{pmatrix} < 0, \quad k = \overline{1,5}. \quad (21)$$

В практическом случае выбор строки  $K_d$  дифференциальной составляющей регулятора может быть осуществлен программно с помощью перебора на конечной сетке. При осуществлении перебора, ключевую роль играют характеристики качества линейной системы (15-16), замкнутой управлением (20) на базе асимптотического наблюдателя (18).

Для формализации данного подхода будем рассматривать задачу выхода объекта на требуемый командный уровень по глубине. С этой целью положим  $y_s = \tilde{y}_s(t)$  – выходной сигнал (16), формируемый с учетом управления (20) с некоторой заданной матрицей  $K_d$ , удовлетворяющей условиям (21), а величина  $y_{se}$  – некоторое установившееся значение выходного сигнала. Положим также

$$\rho(t) = \|\tilde{y}_s(t)\|, \quad \rho_0 = \|y_{se}\|.$$

С учетом данных обозначений могут быть введены следующие типовые характеристики качества.

### 1. Перерегулирование

$$J_p = \frac{\rho_m - \rho_0}{\rho_0} \cdot 100\%,$$

где  $\rho_m$  – точная верхняя граница функции  $\rho(t)$ . Смысл использования данной характеристики состоит в том, что при обходе препятствий излишний выброс траектории робота по глубине должен быть минимизирован, так как в противном случае может привести к аварийным ситуациям.

### 2. Время переходного процесса

$$T_p = \inf\{t_m: \rho(t) \in M(\rho_0, \Delta), \forall t \geq t_m\},$$

где  $M(\rho_0, \Delta) = \{\rho: |\rho - \rho_0|/\rho_0 \leq \Delta\}$  – заданная  $\Delta$ -окрестность числа  $\rho_0$ . Аппарату необходимо быстро выходить на заданный командный уровень, так как в условиях непредсказуемого морского дна препятствие может быть замечено в последний момент.

### 3. Колебательность

$$N_k = \left\lceil \frac{n_r}{2} \right\rceil,$$

где  $n_r$  – это количество вещественных корней нелинейного уравнения  $\rho(t) = \rho_0$  на временном отрезке моделирования. При минимизации предыдущих двух характеристик качества за счет выбора матрицы  $K_d$  необходимо также учитывать колебательность процесса, так как чрезмерная колебательность может приводить к невозможности проведения тех операций, для кото-

рых робот был запущен (например, картографирование рельефа дна или осмотр подводных трубопроводов).

### 3.5. Командный уровень на базе сонара

Текущая постановка задачи управления НПА подразумевает обход препятствий, возникающих по курсу движения робота, в вертикальной плоскости. Для обнаружения препятствий на аппарате установлен сонар. В данном разделе будут сформулированы основные допущения, позволяющие моделировать работу сонара в рамках настоящей работы.

Будем считать, что выполнены следующие условия.

1. Поверхность дна является достаточно гладкой для того, чтобы можно было моделировать ее в виде конечного набора точек в плоскости  $O_1XZ$ .
2. НПА движется с постоянной скоростью по курсу  $u = u_0$  на всем временном отрезке моделирования.
3. Отклонения по углу курса малы в достаточной мере, чтобы считать, что траектория движения аппарата целиком лежит в малой окрестности плоскости  $O_1XZ$ .

Выполнение данных обстоятельств позволяет рассматривать координату  $X$  аппарата как  $X = u_0t$  и тем самым описывать положение робота в вертикальной плоскости в координатах  $(t, Z)$ .

С учетом вышесказанного, отбор точек, доступных для наблюдения сонара, может быть организован следующим образом. Пусть  $P_i = (t_i, Z_i)$  – рассматриваемые точки поверхности, а  $R = R(t, Z_R)$  – текущее положение НПА REMUS в системе координат. В силу того, что нас не интересуют точки поверхности, оставшиеся позади, рассматриваем только те, значения  $t_i$  которых больше текущего времени  $t$ .

Введем точку  $N = R(t, Z_R + 1)$  и вектор единичной длины  $\overrightarrow{RN}$ . Рассмотрим также последовательность векторов  $\overrightarrow{RP}_i$ . Пусть из текущего положения НПА видна некоторая точка  $P_k$ .

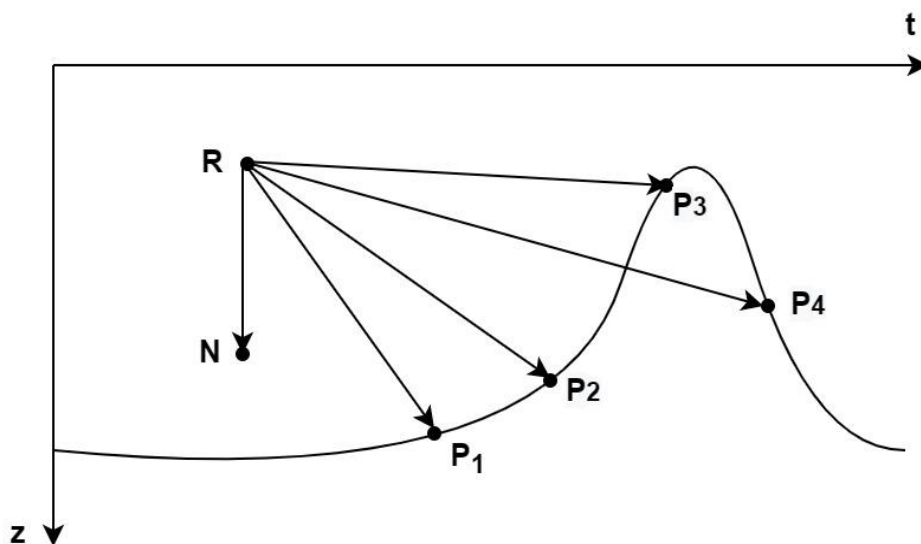


Рис.3. Моделирование работы сонара

Заметим следующий факт, который наглядно продемонстрирован на рисунке 3: в случае когда, из текущего положения НПА видна следующая точка поверхности  $P_{k+1}$ , угол между векторами  $\overrightarrow{RN}$  и  $\overrightarrow{RP_{k+1}}$  должен быть больше угла, образуемого векторами  $\overrightarrow{RN}$  и  $\overrightarrow{RP_k}$ . И наоборот, если некоторая точка не видна из текущего положения REMUS'a, то угол между указанными векторами меньше угла, образованного последней видимой точкой.

При этом измерение угла между векторами можно заменить на вычисление косинусного расстояния, которое существенно проще может быть реализовано программно. Напомним, что косинусное расстояние определяется следующей формулой:

$$\rho_{\text{cosine}}(u, v) = 1 - \frac{\langle u, v \rangle}{\|u\| \cdot \|v\|},$$

где под  $\langle u, v \rangle$  понимается скалярное произведение векторов  $u$  и  $v$ .

По отобранным указанным способом точкам можно восстановить приближенный вид поверхности дна, например, используя сплайны, и получить тем самым некоторую функцию  $\eta = \eta(\tau, t, Z_R)$ , описывающую наблюдаемый рельеф в момент времени  $t$  на глубине  $Z_R$ .

Заметим, что в подавляющем большинстве практических ситуаций с учетом невысокой скорости движения рассматриваемого НПА, для построе-



ния управляющего воздействия на базе сонара, аппарату необходимо наблюдать поверхность лишь на несколько метров вперед. Если, кроме того, поверхность не содержит резких перепадов высот, то на всей траектории пути робота ему будет доступна для наблюдения та часть поверхности, которая необходима для корректировки командного уровня.

Командный уровень в данном случае можно определить как

$$\xi = \eta(t + t_0) - Z_0,$$

где  $\eta$  – кусочно-гладкая функция времени, приближающая рельеф поверхности,  $t_0$  – время, необходимое аппарату и управляющим приводам, чтобы отработать командный сигнал, а  $Z_0$  – заданное расстояние до дна, которое необходимо поддерживать.

## 4. Компьютерное моделирование НПА REMUS

В данном разделе приводятся результаты применения описанного подхода к синтезу обратной связи для управления НПА REMUS 100 в вертикальной плоскости с целью обхода препятствий и поддержания нулевого положения по углу курса в горизонтальной плоскости. Проведение всех необходимых вычислений и имитационного моделирования процессов управления осуществляется в среде MATLAB с привлечением прикладного пакета Simulink.

### 4.1. Компьютерное моделирование наблюдателя и управления в горизонтальной плоскости

Обратимся к линейной системе (11-12) движения аппарата в горизонтальной плоскости. В среде Matlab-Simulink процесс нахождения линейного приближения к рассматриваемому объекту сводится к использованию встроенной функции `linmod`:

$$[A\_full, B\_full] = \text{linmod}('Remus', X, U).$$

Здесь “Remus” – Simulink-модель, представленная на рис. 4 и построенная на базе m-функции “remus.m”, в которой реализована правая часть нелинейных уравнений динамики робота. Данная функция также приведена в приложении А настоящей работы.

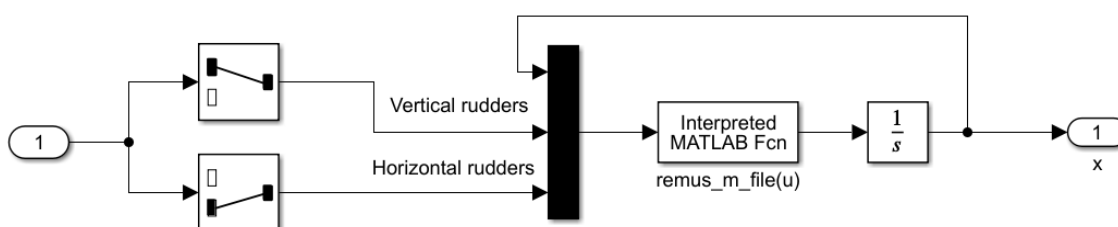


Рис.4. Модель для линеаризации

Как было отмечено ранее, матрица  $L$  асимптотического наблюдателя (13) может быть найдена в виде решения задачи модального синтеза с рас-

пределением полюсов Баттерворта. В рассматриваемом практическом примере выберем значение параметра  $\rho = 3$ . Тогда с использованием функции `place` пакета прикладных программ `Control Systems Toolbox` имеем:

```
L = place(A', C', [-3, ...
                  -3*(cos(pi/3)+1i*sin(pi/3)), ...
                  -3*(cos(pi/3)-1i*sin(pi/3))])';
```

С учетом произведенных вычислений имеем следующие матрицы в уравнениях (11-13):

$$A = \begin{pmatrix} -0.805 & -0.5721 & 0 \\ -4.5371 & -0.4952 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0.2923 \\ -1.6157 \\ 0 \end{pmatrix},$$

$$C = (0 \quad 0 \quad 1), \quad L = \begin{pmatrix} -5.7272 \\ 14.086 \\ 4.6998 \end{pmatrix}.$$

Обратимся теперь к синтезу скоростного закона управления. Как было отмечено ранее, для начала необходимо синтезировать базовый регулятор по состоянию. В среде `MATLAB` это можно сделать с использованием `lqr`:

```
K = -lqr([A, B; zeros(1, 4)], [0; 0; 0; 1], ...
         diag([1, 1, 10, 1]), 1);
```

где матрицы `Q` и `R` интегрального функционала и результат функции применения функции `lqr` имеют вид

$$Q = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad R = 1, \quad K = (-8.51 \quad 4.65 \quad 3.16 \quad 4.58).$$

Такой выбор коэффициентов рассматриваемых матриц `Q` и `R` обусловлен желанием вывести НПА на желаемый угол курса наиболее быстрым способом.

На основе полученного базового регулятора по состоянию построим его скоростной эквивалент

$$u_r = \mu \dot{z} + \nu y_r,$$

используя соответствующие формулы из подраздела 3.2. Результатом преобразований станут матрицы

$$\mu = (-2.6792 \quad 2.3522 \quad 4.2835), \quad \nu = 3.1623.$$

Экспериментально покажем, что полученная линейная система (11-12), замкнутая скоростным регулятором (15) на базе оценки вектора состояния (13), обладает астатизмом. Для этого на вход линейной модели подадим ступенчатое возмущение по углу курса  $\psi$  величиной в 10 град, которое активируется в пятую секунду процесса симуляции. Результат обработки возмущения представлен на рисунке 5.

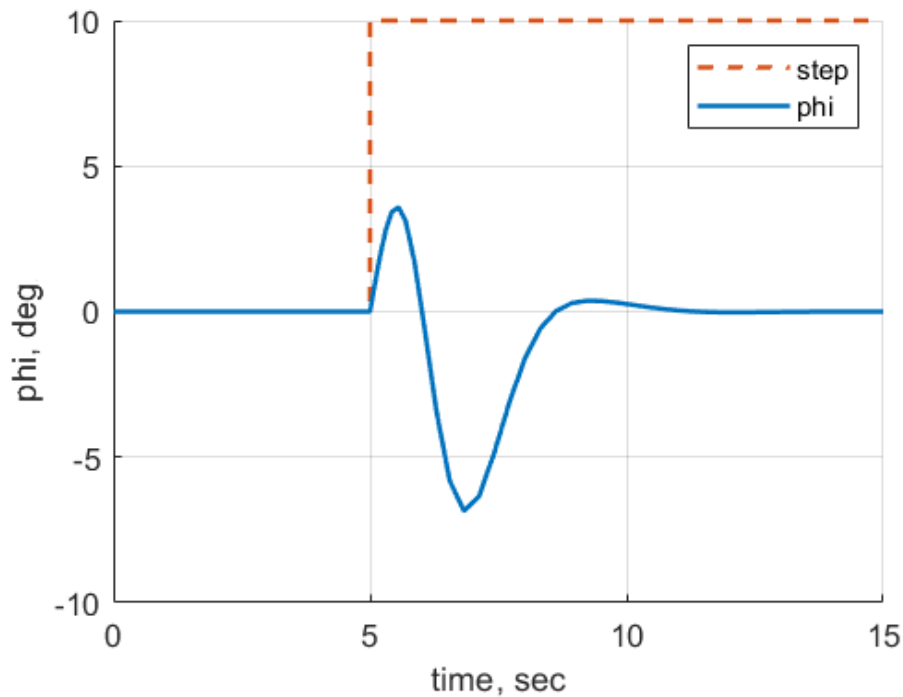


Рис.5. Обработка внешних ступенчатых возмущений

Как можно видеть, синтезированное управление отлично справляется с поставленной перед ним задачей.

## 4.2. Компьютерное моделирование наблюдателя и управления в вертикальной плоскости

Перейдем к рассмотрению вопросов синтеза управления в вертикальной плоскости. Для этого вновь обратимся к функции `linmod`, представленной ранее.

Результатом линеаризации в рамках рассматриваемого вертикального случая (16-17) станут матрицы

$$A = \begin{pmatrix} -0.7875 & 0.5927 & 0 & 0.021 \\ 4.5392 & -0.5008 & 0 & -0.7163 \\ 1 & 0 & 0 & -1.51 \\ 0 & 1 & 0 & 0 \end{pmatrix},$$
$$B = \begin{pmatrix} -0.2858 \\ -1.6194 \\ 0 \\ 0 \end{pmatrix}, \quad C = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

И вновь по аналогии с горизонтальным случаем синтезируем асимптотический наблюдатель (18) на базе модального синтеза с биномиальным распределением полюсов матрицы наблюдателя:

$$L = \text{place}(A', C', [-3.41, -3.42, -3.43, -3.44])';$$

Результатом вычисления данной функции будет матрица

$$L = \begin{pmatrix} 9.5915 & 3.3545 \\ 25.1158 & 10.6208 \\ 6.0456 & -0.9071 \\ 4.5210 & 6.3661 \end{pmatrix}.$$

Обратимся к синтезу пропорциональной составляющей управления (19) на базе решения задачи LQR-синтеза:

$$Kp = -\text{lqr}(A1, B1, \text{diag}([1, 1, 30, 1, 1]), 1);$$

где  $A1$  и  $B1$  представляют собой матрицы расширенной системы (системы (15-16) с учетом уравнения динамики привода (9)). Коэффициенты матриц  $Q$

и  $R$  были выбраны из требования наиболее строгого следования глубины НПА текущему командному уровню. Матрица  $K_p$  при этом имеет вид

$$K_p = (8.6065 \quad 8.3488 \quad -5.4772 \quad 11.2346 \quad -5.741).$$

Рассмотрим теперь вопрос выбора коэффициентов матрицы  $K_d$  дифференциальной части управления (19). С этой целью в рамках настоящей работы была реализована компьютерная программа перебора коэффициентов матрицы  $K_d$  на конечной сетке. На вход программе подаются матрицы  $A$ ,  $B$  линейной системы (16), матрица пропорциональной части  $K_p$  управления (19) и некоторое начальное приближение к матрице  $K_d$ , которое может быть получено вручную исходя из собственных соображений. Так же в программе указываются максимально допустимая величина перерегулирования и требуемое время переходного процесса, после чего запускается процесс перебора. При этом на каждом шаге вычисления матрицы  $K_d$  происходит проверка условий (21) асимптотической устойчивости замкнутой системы, изложенных в подразделе 3.4.

По достижении поставленных условий, в случае, если такое возможно, программа выдает матрицу  $K_d$  оптимальную в смысле поставленной задачи. В частности, была получена следующая матрица коэффициентов дифференциальной части:

$$K_d = (-2.33 \quad 6.4 \quad -12.41 \quad 19.36 \quad -1.57).$$

Обратимся к управлению (20) и пересчитаем матрицы управления для избавления от алгебраической петли. С учетом значений строк  $K_p$  и  $K_d$  имеем

$$\tilde{K}_{p_x} = (3.3488 \quad 3.2485 \quad -2.1312 \quad 4.3714),$$

$$\tilde{k}_{p_\delta} = -2.2338,$$

$$\tilde{K}_{d_x} = (-0.9066 \quad 2.4903 \quad -4.8288 \quad 7.5331).$$

Для демонстрации необходимости использования дифференциальной части в уравнении управляющего воздействия, вычислим выход системы (16-

18) на заданный командный уровень по глубине  $Z^* = 3$  при исходных нулевых начальных условиях. При этом будем замыкать систему управлением P и PD-регуляторами на базе управления (20). Результаты данного сравнения представлены на рисунке 6.

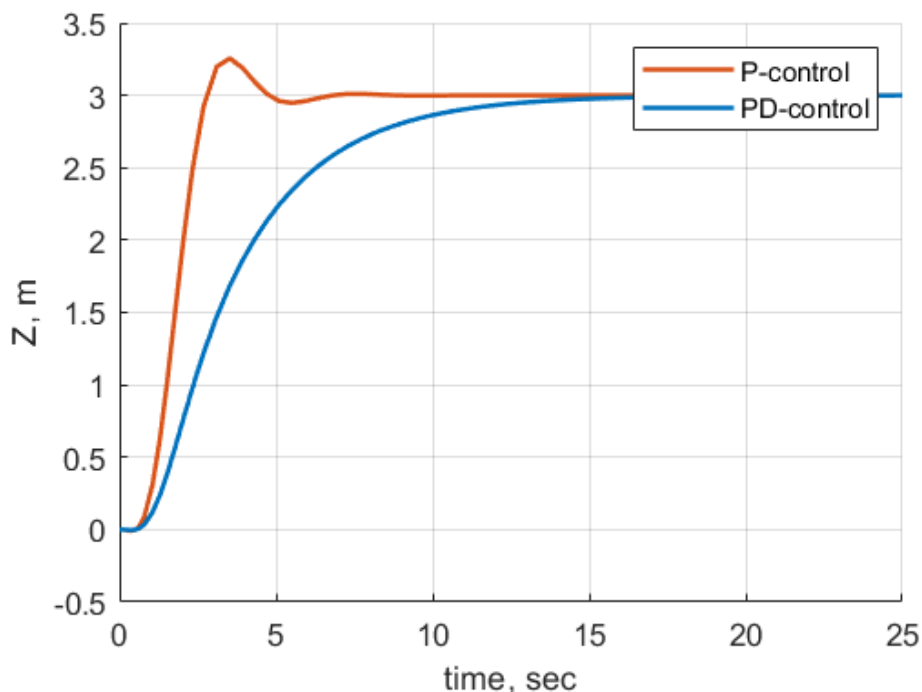


Рис.6. Сравнение регуляторов

Несмотря на кажущуюся незначительность перерегулирования в линейной системе, замкнутой P-регулятором, в нелинейной системе за счет не точности наблюдателя и сложной исходной динамики объекта, перерегулирование может быть существенно больше, тогда как PD-регулятор будет показывать стабильно хороший с точки зрения характеристик качества результат.

### 4.3. Моделирование динамической системы в среде Simulink

Перейдем к реализации контура обратной связи в исходных нелинейных уравнениях рассматриваемого НПА REMUS 100.

Компьютерная модель, построенная в соответствии с нелинейными уравнениями (1-3), наблюдателями (13) и (18), управляющими воздействиями (15) и (20) представлена на рисунке 7.

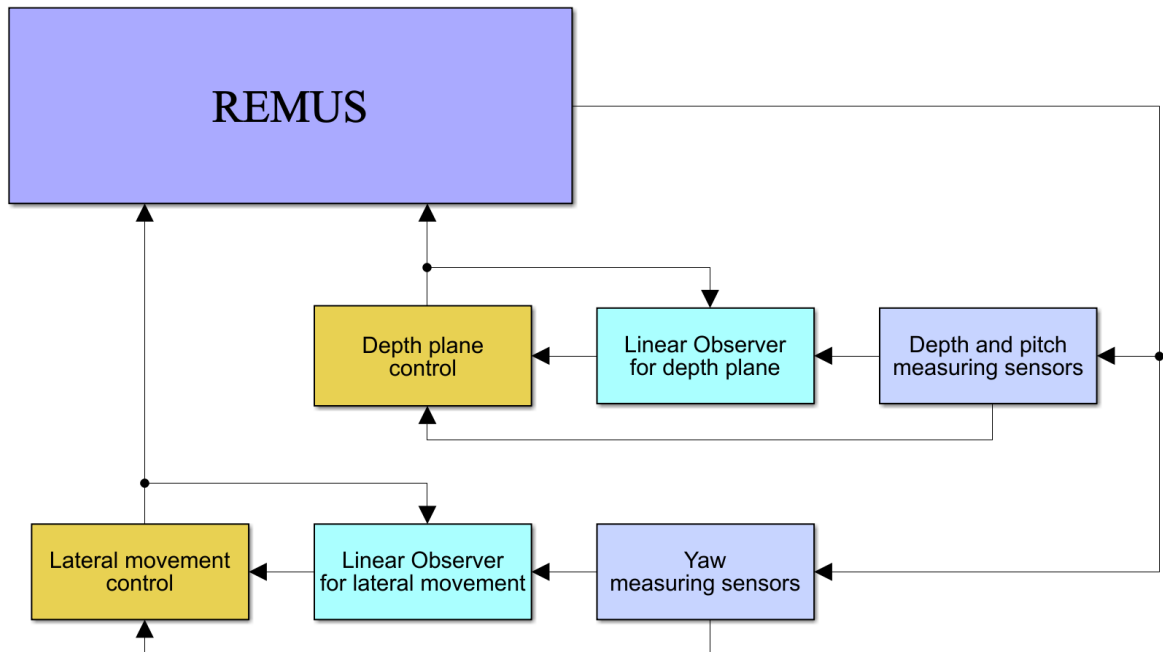


Рис.7. Simulink- модель системы управления НПА REMUS 100

В рассматриваемой схеме крайние правые блоки отвечают за измерение наблюдаемых переменных: верхний измеряет глубину и угол дифферента НПА, нижний – угол курса.

Линейные наблюдатели сформированы в соответствие со следующими блок-схемами, представленными на рисунках 8 и 9.

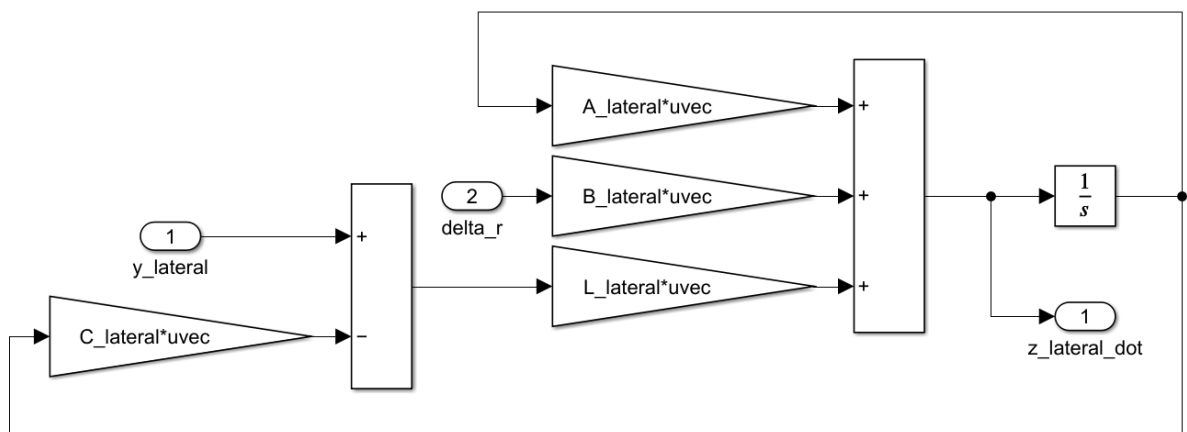


Рис.8. Блок-схема наблюдателя в горизонтальной плоскости



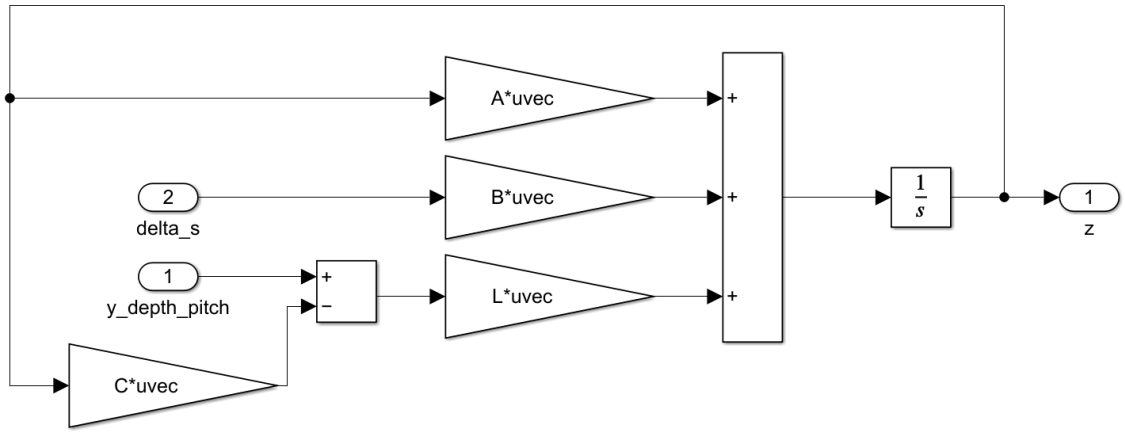


Рис.9. Блок-схема наблюдателя в вертикальной плоскости

Блоки управления по принципу обратной связи, реализованные в соответствии с уравнениями (15) и (20) изображены рисунках 10 и 11.

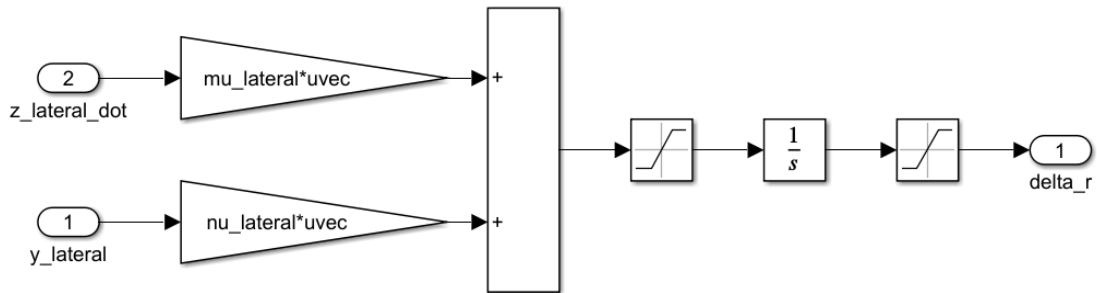


Рис.10. Блок-схема управления в горизонтальной плоскости

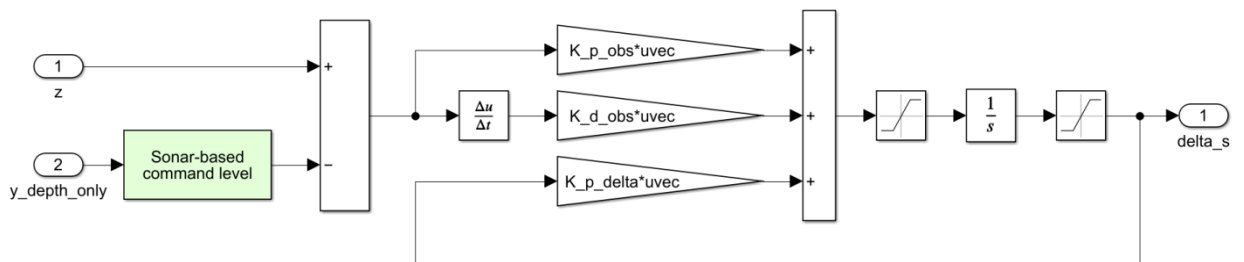


Рис.11. Блок-схема управления в вертикальной плоскости

Здесь стоит заметить, что помимо структуры управлений, описанной уравнениями (15) и (20), в рамках данной нелинейной модели также учитываются ограничения на переменные:  $\pm 30$  градусов на отклонение вертикальных и горизонтальных рулей и  $\pm 5$  градусов в секунду на скорости соответствующих рулей.

Кроме того, в блок-схеме, представленной на рисунке 11 можно заметить блок командного сигнала, основанного на показателях сонара. Данный блок реализован на базе функции “sonar.m”, разработанный в соответствии с моделированием сонара в подразделе 3.5. Исходный код данной функции представлен в приложении А настоящей работы.

#### 4.4. Обход препятствий в вертикальной плоскости

Перейдем, наконец, к тестированию построенной нелинейной модели (1-3), замкнутой управляющими воздействиями (15) и (20) на базе оценок векторов состояния линейными наблюдателями (13) и (18). С этой целью сформируем две тестовые поверхности, описываемые в среде MATLAB следующими формулами.

```
% тестовая матрица точек, задающая поверхность №1
x1 = 0:2:38;
z1 = 3*sqrt((x1+7)/5);
x2 = 47:2:75;
z2 = -(sin(0.18 * (x2 - 47)) - 9);
points_matrix = [x1', z1'; x2', z2'];

% тестовая матрица точек, задающая поверхность №2
x1 = 8:2:44;
z1 = -(sin(0.18 * x1) + cos(0.433 * x1)) / 2 + 5;
x2 = 56:2:74;
z2 = -(sin(0.18 * (x2 - 10)) + ...
       cos(0.433 * (x2 - 10))) / 2 + 5;
points_matrix = [x1', z1'; 50, 4; x2', z2'];
```

Будем предполагать, что в начальный момент времени НПА находится на глубине 3 метров. Заданное расстояние до дна  $Z_0$ , которое необходимо поддерживать на протяжении всего процесса моделирования, примем равным одному метру.

В результате многочисленных экспериментов было также получено, что для корректной отработки заданного командного уровня аппарату необходимо время  $t_0$  равное в среднем двум секундам, что соответствует полю видимости сонара не менее трех метров.

С учетом указанных фактов, на тестовой поверхности №1 НПА имеет следующую траекторию движения по глубине, представленную на рис. 12.

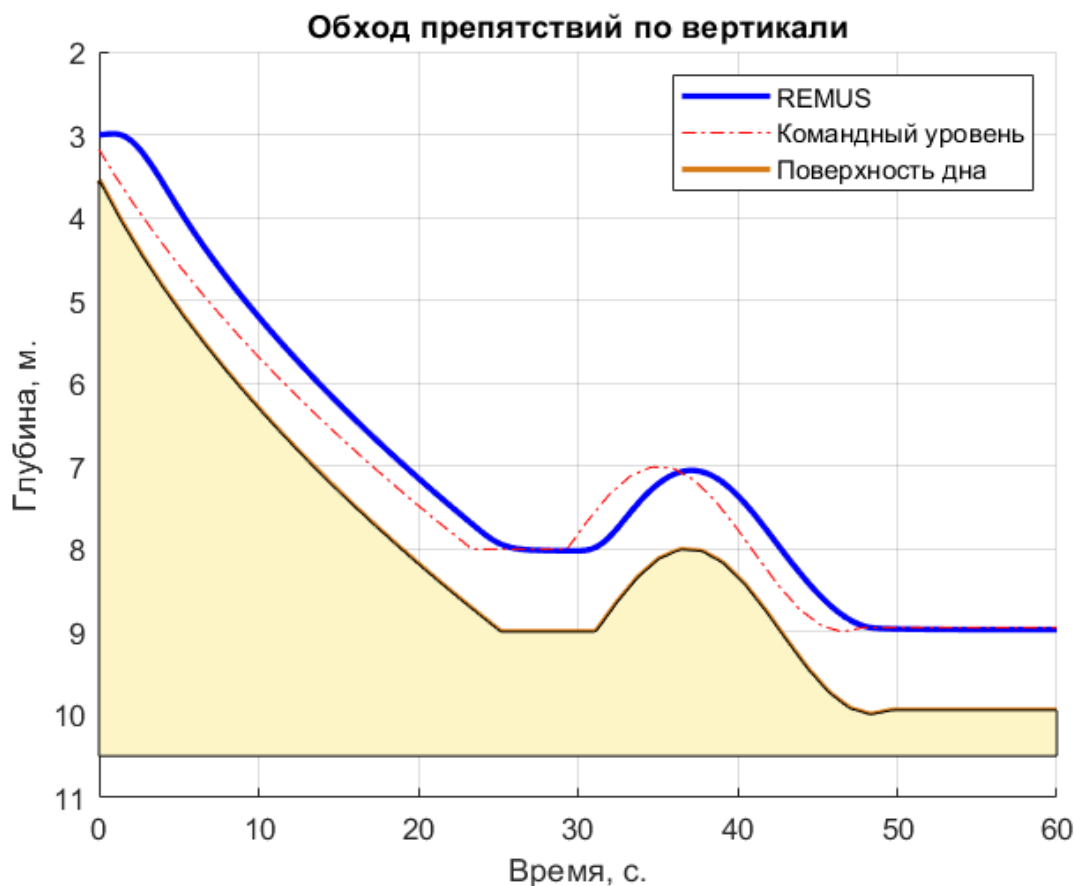


Рис.12. Результаты работы управления на тестовой поверхности дна №1

Среднее расстояние до дна в рассматриваемом первом случае составляет 1.0235 метра, медианное – 1.0243 метра, что является отличными показателями в рамках поставленной задачи.

Перейдем теперь к рассмотрению более сложной тестовой поверхности дна. Все также считаем, что в начальный момент времени НПА находится на глубине трех метров,  $Z_0 = 1$  метр, а  $t_0 = 2$  секунды.

Результат работы управления представлен на рисунке 13.



Рис.13. Результаты работы управления на тестовой поверхности дна №2

Как можно заметить из представленного графика траектории движения НПА, управление отлично справляется с поставленной задачей. Среднее расстояние до дна во втором случае составляет 1.0668 метра, медианное – 0.99019 метра, что также можно считать приемлемыми показателями качества синтезированного управления.

Графики по другим рассматриваемым переменным состояния в рамках данного случая представлены на рисунках 14 и 15.

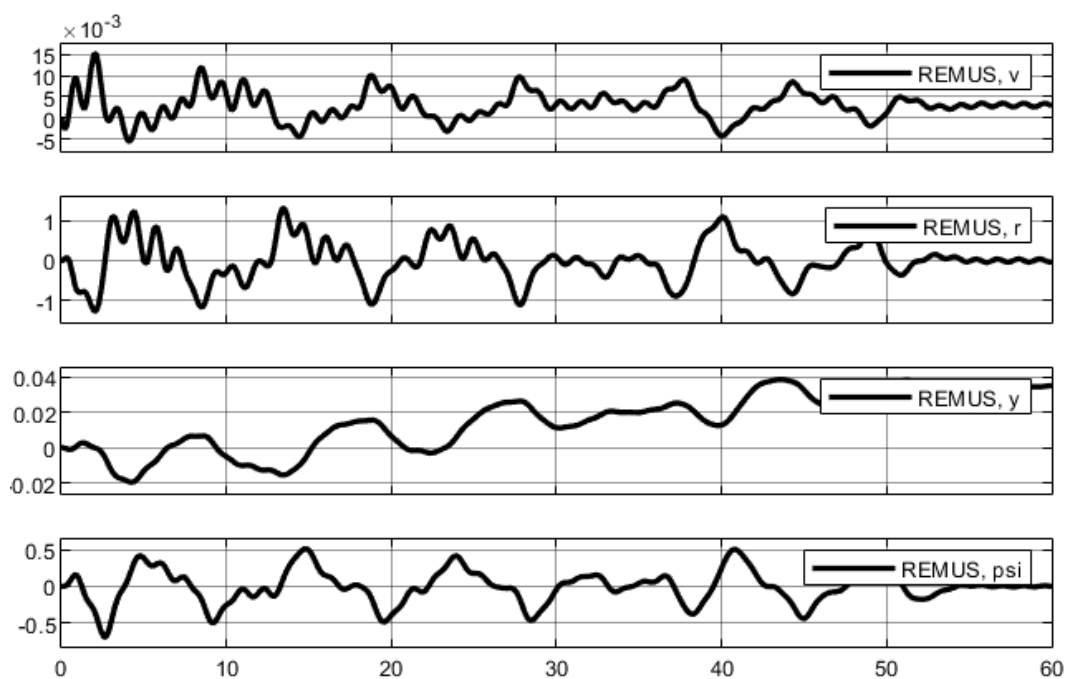


Рис.14. Переменные в горизонтальной плоскости на тестовом режиме №2

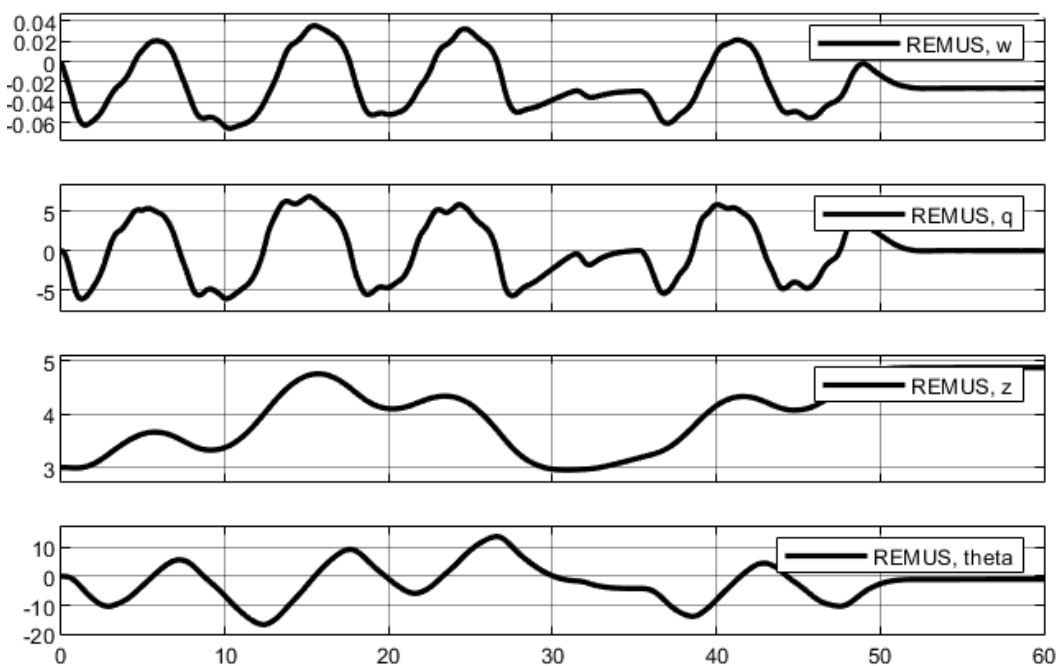


Рис.15. Переменные в вертикальной плоскости на тестовом режиме №2

Из представленных графиков заметим, что существует небольшое отклонение от курса, равное на момент завершения моделирования 0.04 метра, что тем не менее является приемлемым результатом с учетом пройденного за это время пути, составляющего порядка 90 метров.

## Выводы

Существенной особенностью поставленной задачи синтеза управляющего воздействия для необитаемого подводного аппарата REMUS 100 является необходимость обхода препятствий в вертикальной плоскости. Данное требование накладывает серьезные ограничения на выбор инструментов синтеза как управлений, так и асимптотических наблюдателей.

В ходе проделанной работы были рассмотрены различные аспекты разработки алгоритмов управления, большое внимание при этом уделено возможным режимам функционирования робота как в горизонтальной, так и вертикальной плоскостях.

Так, в горизонтальной плоскости была произведена не только настройка базового закона управления как решения задачи LQR-оптимизации, но и осуществлен переход к его скоростному эквиваленту, обеспечивающему астатизм замкнутой на базе асимптотического наблюдателя линейной системы, что наглядно было представлено в рамках главы компьютерного моделирования.

В вертикальной же плоскости был синтезирован PD-регулятор, где обе составляющие, как пропорциональная, так и дифференциальная, выбирались из обеспечения наиболее оптимальных характеристик замкнутой системы. По аналогии с горизонтальной плоскостью, здесь также был разработан асимптотический наблюдатель, обеспечивающий высокую точность оценки вектора состояния.

И наконец, в рамках главы компьютерного моделирования были объединены оба рассматриваемых управляющих комплекса, которые на базе сигнала сонара обеспечивают высокую точность и надежность управления в условиях непредсказуемого рельефа дна.

## Заключение

В процессе проведенного исследования получены следующие результаты, которые выносятся на защиту:

1. Сформулирована задача синтеза многоцелевой структуры управления НПА REMUS 100 и предложены способы нахождения ее элементов с обеспечением желаемой динамики процессов.
2. Сформирован имитационно-моделирующий комплекс в среде MATLAB с привлечением прикладного пакета Simulink для компьютерного моделирования рассматриваемых систем и проведения вычислительных экспериментов.
3. Проведено численное решение задачи синтеза регуляторов для рассматриваемого аппарата, показавшее работоспособность и эффективность принятого подхода.

## Список литературы

1. Веремей Е.И. Линейные системы с обратной связью. СПб.: Лань, 2013. 448 с.
2. Веремей Е.И. Среднеквадратичная многоцелевая оптимизация. СПб: Изд-во С.-Петербур. ун-та, 2016. 408 с.
3. Timothy Prester. Verification of a Six-Degree of Freedom Simulation Model for the REMUS Autonomous Underwater Vehicle // Massachusetts institute of technology, 2001. 127 с.
4. Jose Alberti. Modeling and model identification of autonomous underwater vehicles // Calhoun: The NPS Institutional Archive, 2015. 87 с.
5. John Gornowich. Tracking control for a formation of Autonomous Underwater vehicles // George Mason University, 2008. 108с.
6. Tyler H. Furukawa. Reactive obstacle avoidance for the REMUS Autonomous Underwater Vehicle utilizing a forward looking sonar // Calhoun: The NPS Institutional Archive, 2006. 79 с.
7. Matthew D. Phaneuf. Experiments with the REMUS AUV // Calhoun: The NPS Institutional Archive, 2004. 77 с.
8. Sean Michael Doherty. Cross body thruster control and modeling of a body of revolution Autonomous Underwater Vehicle // Calhoun: The NPS Institutional Archive, 2011. 115 с.
9. Christopher D. Chuhuran. Obstacle avoidance control in the vertical plane for the REMUS autonomous underwater vehicle // Calhoun: The NPS Institutional Archive, 2003. 63 с.
10. Thor I. Fossen. Guidance and Control of Ocean Vehicles // John Wiley & Sons, 1994. 480 с.
11. Thor I. Fossen. Handbook of Marine Craft Hydrodynamics and Motion Control // John Wiley & Sons, 2011. 575 с.



## Приложение А

Ниже представлен исходный код вспомогательных функций, рассматриваемых в рамках моделирования управляющего воздействия для необитаемого подводного аппарата REMUS 100.

### 1. Функция «remus.m» вычисления правых частей нелинейных уравнений:

```
% Нелинейная модель AUV Remus 100
%
function output = remus(input, U_init)
% Входные данные:
% Вектор состояния:
% x = [u v w p q r xpos ypos zpos phi theta psi]'
% где:
% u = продольная скорость (м/с)
% v = скорость боковая (м/с)
% w = скорость подъема (м/с)
% p = скорость крена (рад/с)
% q = скорость дифферента (рад/с)
% r = скорость рыскания (рад/с)
% xpos = положение в направлении x (м)
% ypos = положение в направлении y (м)
% zpos = положение в направлении z (м)
% phi = угол крена (рад)
% theta = угол дифферента (рад)
% psi = угол курса (рад)
%
% Вектор управления:
% [delta_r delta_s]'
% где
% delta_r = угол вертикального руля (градусы)
% delta_s = кормовой порт и правый борт – горизонтальные рули
(градусы)

% Входные переменные
u = input(1); v = input(2); w = input(3);
p = input(4); q = input(5); r = input(6);
ksi = input(7); dzeta = input(8); eta = input(9);
phi = input(10); theta = input(11); psi = input(12);

delta_r = input(13); % вертикальные рули для горизонтальной
плоскости
delta_s = input(14); % горизонтальные рули для вертикальной
плоскости

% delta_r = pi/180 * input(13); % вертикальные рули для горизон-
тальной плоскости
% delta_s = pi/180 * input(14); % горизонтальные рули для верти-
кальной плоскости

% Физические параметры и гидродинамические производные
```

```

L = 1.3327; ge = 9.8; xG = 0; yG = 0; zg = 19.6*10^(-3);
xB = 0; yB = 0; zB = 0; rho = 1030; m = 30.48;
W = m*ge; B = W + (0.75*4.44822162); Ixx = 0.177;
Iyy = 3.45; Izz = 3.45; Ixy = 0; Iyz = 0; Ixz = 0;
Tnn = 6.279e-004; Tnu = 0; Qnn = -1.121e-005; Qnu = 0;

```

#### % Гидродинамические производные

```

Xwq = -35.5; Xqq = -1.93; Xvr = 35.5; Xrr = -1.93;
Yvv = -1310.0; Yrr = 0.632; Yuv = -28.6; Yur = 5.22;
Ywp = 35.5; Ypq = 1.93; Yuudr = 9.64; Zww = -1310.0;
Zqq = -0.632; Zuw = -28.6; Zuq = -5.22; Zvp = -35.5;
Zrp = 1.93; Zuuds = -9.64; Kpp = -0.130; Mww = 3.18;
Mqq = -188; Muw = 24.0; Muq = -2.0; Mvp = -1.93;
Mrp = 4.86; Muuds = -6.15; Nvv = -3.18; Nrr = -94.0;
Nuv = -24.0; Nur = -2.0; Nwp = -1.93; Npq = -4.86;
Nuudr = -6.15; Kpdot = -0.0704; cdu = 0.2;
Af = 0.0285; Xuu = -0.5*rho*cdu*Af; Xu = -rho*cdu*Af*U_init;

```

#### % Коэффициенты добавленной массы

```

Xudot = -0.93; Yvdot = -35.5; Yrdot = 1.93;
Zwdot = -35.5; Zqdot = -1.93; Mwdot = -1.93;
Mqdot = -4.88; Nvdot = 1.93; Nrdot = -4.88;

```

#### % Члены добавленной массы

```

Zwc = -15.7; Zqc = 0.12; Mwc = -0.403;
Mqc = -2.16; mq = 0.3; taup = 0.1;

```

#### % Связь добавленных масс

```

Xqa = Zqdot*mq; Zqa = -Xudot*U_init;
Mwa = -(Zwdot - Xudot)*U_init; Mqa = -Zqdot*U_init;

```

#### % Матрица массы

```

Mass_Matr(1,1) = m - Xudot; Mass_Matr(1,5) = m*zg;
Mass_Matr(1,6) = 0; Mass_Matr(2,2) = m - Yvdot;
Mass_Matr(2,4) = -m*zg; Mass_Matr(2,6) = - Yrdot;
Mass_Matr(3,3) = m - Zwdot; Mass_Matr(3,4) = 0;
Mass_Matr(3,5) = - Zqdot; Mass_Matr(4,2) = -m*zg;
Mass_Matr(4,3) = 0; Mass_Matr(4,4) = Ixx - Kpdot;
Mass_Matr(5,1) = m*zg; Mass_Matr(5,3) = - Mwdot;
Mass_Matr(5,5) = Iyy - Mqdot; Mass_Matr(6,1) = 0;
Mass_Matr(6,2) = - Nvdot; Mass_Matr(6,6) = Izz - Nrdot;

```

```

Xf = -(W-B)*sin(theta) + Xuu*u*abs(u) + (Xwq-
m)*w*q + Xqq*q*q + (Xvr+m)*v*r + Xrr*r*r - m*zg*p*r -
Xuu*U_init*abs(U_init);

```

```

Yf = (W-
B)*cos(theta)*sin(phi) + Yuv*u*v + Yvv*v*abs(v) + Yrr*r*abs(r) + (Yur-
m)*u*r + (Ywp+m)*w*p + Ypq*p*q + Yuudr*u*u*delta_r - m*zg*q*r;

```

```

Zf = (W-B)*cos(theta)*cos(phi) + (Zuq+m)*u*q + (Zvp-
m)*v*p + Zrp*r*p + Zww*w*abs(w) + Zqq*q*abs(q) + Zuw*u*w + Zuuds*u*u*delta
_s + m*zg*(p*p + q*q);

```

```

Kf = -W*zg*cos(theta)*sin(phi)+Kpp*p*abs(p)-(Izz-
Iyy)*q*r+m*zg*(u*r-w*p)+0.995*(yG*W-yB*B)- 1*0.093*(zg*W-zB*B);

Mf = -
W*zg*sin(theta)+Muw*u*w+Mvp*v*p+Mrp*r*p+Muq*u*q+Mww*w*abs(w)+Mqq
*q*abs(q)+Muuds*u*u*delta_s-(Ixx-Izz)*r*p-m*zg*(w*q-v*r);

Nf =
Nuv*u*v+Nwp*w*p+Npq*p*q+Nur*u*r+Nvv*v*abs(v)+Nrr*r*abs(r)+Nuudr*
u*u*delta_r-(Iyy-Ixx)*p*q;

Forces = [Xf Yf Zf Kf Mf Nf]';

xdot1 = Mass_Matr\Forces;

xdot2 = [u*cos(psi)*cos(theta)+v*(-
sin(psi)*cos(phi)+cos(psi)*sin(theta)*sin(phi))+w*(sin(psi)*sin(
phi)+cos(psi)*sin(theta)*cos(phi))
u*sin(psi)*cos(theta)+v*(cos(psi)*cos(phi)+sin(psi)*sin(theta)*s
in(phi))+w*(-cos(psi)*sin(phi)+sin(psi)*sin(theta)*cos(phi))
-u*sin(theta)+v*cos(theta)*sin(phi)+w*cos(theta)*cos(phi)
p+q*sin(phi)*tan(theta)+r*cos(phi)*tan(theta)
q*cos(phi)-r*sin(phi)
q*sin(phi)/cos(theta)+r*cos(phi)/cos(theta)];

output = [xdot1; xdot2];

```

2. Функция нахождения косинусного расстояния, используемая для определения точек, видимых аппарату из текущего положения по глубине.

```

function distance = cosine(u, v)
distance = 1 - dot(u, v) / (norm(u, 2) * norm(v, 2));

```

3. Функция «sonar.m» моделирующая работу сонара и выдающая командный уровень.

```

function target_depth = sonar(current_time, current_depth,
points_matrix, time_lag, target_level, U_init)
% current_time - текущее время моделирования, сек
% current_depth - глубина НПА в данный момент, м
% time_lag - время, необходимое НПА для выхода на уровень, сек
% target_level - заданное расстояние до дна, м
% points_matrix - матрица точек, задающих поверхность дна
% U_init - постоянная скорость движения вперед, м/с

% количество точек поверхности
points_num = length(points_matrix(:, 1));

% текущий пройденный путь
current_distance = U_init * current_time;
% текущее положение в СК
remus_cord = [current_distance, current_depth];

```

```

% ищем текущее положение между точками
k = 1;
last = remus_cord;
next = points_matrix(k, :);
while (current_distance > next(1)) & (k < points_num)
    k = k + 1;
    last = next;
    next = points_matrix(k, :);
end

% путь с учетом time_lag
lag_distance = current_distance + U_init * time_lag;

% фильтруем оставшиеся точки, т.е. рассматриваем то, что можем
видеть
cosine_last = cosine([0, 1], next - remus_cord); % <- тут
while (lag_distance > next(1)) & (k < points_num)
    k = k + 1;
    cosine_next = cosine([0, 1], points_matrix(k, :) - re-
mus_cord);
    if cosine_next > cosine_last
        cosine_last = cosine_next;
        last = next;
        next = points_matrix(k, :);
    end
end

% обработка возможных случаев
if k == 1
    target_depth = next(2) - target_level;
elseif (k == points_num) & (lag_distance > next(1))
    target_depth = next(2) - target_level;
else
    % тангенс наклона уровня
    rate = (next(2) - last(2)) / (next(1) - last(1));
    % командный сигнал по глубине
    target_depth = rate * (lag_distance - last(1)) - tar-
get_level + last(2);
end

```