

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И СИСТЕМ

ЯКОВЛЕВ АЛЕКСЕЙ АЛЕКСАНДРОВИЧ

Выпускная квалификационная работа бакалавра

**СИНТЕЗ АЛГОРИТМОВ УПРАВЛЕНИЯ
ДЛЯ СТАБИЛИЗАЦИИ ПРОГРАММНОГО ДВИЖЕНИЯ
МОРСКОГО СУДНА**

Направление 01.03.02

Прикладная математика и информатика

Научный руководитель,
доктор физ.-мат. наук,
профессор
Сотникова М. В.

Санкт-Петербург
2018

Содержание

Введение	3
Постановка задачи	4
Обзор литературы	6
Глава 1. Описание математических моделей	8
1.1. Математическая модель движения судна	8
1.2. Математическая модель бокового ветра	12
Глава 2. Синтез закона управления	14
2.1. Линеаризация математической модели	14
2.2. Синтез базового закона управления	16
2.2.1. Описание предлагаемого подхода	16
2.2.2. Построение асимптотического наблюдателя	17
2.2.3. Синтез регулятора	20
2.3. Компенсация постоянного внешнего возмущения	22
Глава 3. Имитационное моделирование процессов управления	25
3.1. Программный комплекс	25
3.2. Примеры имитационного моделирования	30
3.2.1. Стабилизация при отклонении начальных данных	31
3.2.2. Компенсация воздействия бокового ветра	34
Выводы	37
Заключение	39
Список литературы	40
Приложения	42

Введение

В наше время, как и на протяжении большей части истории человечества, надводные морские подвижные объекты являются одним из наиболее эффективных средств перевозки пассажиров и различных грузов.

Однако, несмотря на значительный прогресс в их технической модернизации, из-за специфики среды, в которой они функционируют, на морские суда зачастую воздействуют внешние факторы, которые могут препятствовать достижению стоящих перед ними целей. В частности, одной из таких целей является недопущение отклонения от заданного движения.

В связи с этим подобная задача, называемая задачей стабилизации программного движения [1] морского судна, актуальности не теряет. Также она нередко является нетривиальной. Её сложность зависит от сложности модели, описывающей подвижный объект, от природы возмущений, оказывающих на него влияние, а также от специфики программного движения.

Разумеется, можно попытаться решить эту задачу «вручную», как это и происходило до относительно недавнего времени. Тем не менее очевидно, что сейчас, располагая развитым аппаратом математической теории управления, равно как и техническими возможностями его реализации, значительно более разумным и оптимальным во многих смыслах подходом является создание системы управления, автоматически стабилизирующей морской подвижный объект.

Однако динамика реальных судов в большинстве случаев описывается сложными нелинейными системами дифференциальных уравнений, к работе с которыми нет универсального подхода, поэтому нередко приходится прибегать к их упрощению путём линейного приближения. Но даже так, если линеаризованная в окрестности заданного программного движения модель динамики объекта оказывается нестационарной (то есть её параметры зависят от времени), то снова возникает недостаточность методологии.

В данной работе в рамках описанной ситуации предлагается подход к решению задачи стабилизации программного движения морского судна, применяющийся затем для работы с моделью реального объекта, а также осуществляется компьютерное моделирование процессов управления.

Постановка задачи

Объектом управления в решаемой задаче является морское судно, рассматривается его движение в горизонтальной плоскости. Цель управления состоит в стабилизации программного движения данного судна при наличии неполной информации о векторе состояния объекта и ограничениях на управление, которые обусловлены особенностями конструкции привода. Также объект управления подвержен воздействию внешнего возмущения — бокового ветра. Таким образом, возникает ряд подзадач:

- поиск математических моделей объекта управления и внешнего возмущения;
- разработка подхода к упрощению уравнений движения морского судна в горизонтальной плоскости;
- построение оценки вектора состояния объекта;
- синтез базового закона управления и его модификация для компенсации воздействия внешнего возмущения на определённые компоненты вектора состояния;
- реализация имитационного компьютерного моделирования процессов управления.

Для формализации цели управления рассматривается система, описывающая движение морского судна в горизонтальной плоскости в общем виде:

$$\dot{x} = F(t, x, \bar{x}, \delta_v, d), \quad (1)$$

где t — время, $x(t)$ — фактическое движение морского судна в горизонтальной плоскости, $\bar{x}(t)$ — не характеризующие движение в горизонтальной плоскости компоненты вектора состояния объекта управления, $\delta_v(t)$ — управляющее воздействие, $d(t)$ — внешнее возмущение, $F(\cdot)$ — нелинейная вектор-функция.

Решением системы (1), находящейся под воздействием некоторого программного управления $\tilde{\delta}_v(t)$, при возмущении $d(t) \equiv \mathbf{0}$ и начальном состоянии \tilde{x}_0 является программное движение $\tilde{x}(t)$.

Теперь цель управления может быть записана в виде:

$$x^1(t) \longrightarrow \tilde{x}^1(t) \text{ при } t \longrightarrow +\infty,$$

$$x_0 \in \Omega_\varepsilon(\tilde{x}_0).$$

Здесь $x^1(t)$ — часть вектора $x(t)$, воздействие внешнего возмущения на компоненты которой требуется нивелировать (при этом $x^1(t) = x(t)$, если $d(t) \equiv \mathbf{0}$), x_0 — фактическое начальное состояние объекта управления, $\Omega_\varepsilon(\tilde{x}_0)$ — некоторая окрестность \tilde{x}_0 .

В процессе решения поставленной задачи управление $\delta_v(t)$ формируется следующим образом:

$$\delta_v(t) = \tilde{\delta}_v(t) + \hat{\delta}_v(t),$$

где $\hat{\delta}_v(t)$ — искомое стабилизирующее управляющее воздействие. Упомянутые ранее ограничения на управление имеют вид:

$$|\dot{\tilde{\delta}}_v(t)| \leq a, \quad |\tilde{\delta}_v(t)| \leq b, \quad |\dot{\delta}_v(t)| \leq a, \quad |\delta_v(t)| \leq b, \quad (2)$$

параметры a и b определяются конкретной моделью судна.

Обеспечение определённого качества управления достигается в процессе синтеза регулятора.

Обзор литературы

При написании данной работы используется как различная учебно-методическая литература, так и узконаправленные научные труды, посвящённые передовым разработкам в области автоматической теории управления (в частности ориентированные конкретно на морские подвижные объекты).

Книга [1] выступает в качестве источника базовых сведений, касающихся решения проблемы стабилизации программных движений и методов синтеза управлений.

Математическое моделирование бокового ветра и динамики морского судна выполняется на основе работ [2, 3, 4]. Изложенный в них материал также способствует пониманию природы процессов, происходящих во время движения морского подвижного объекта, и воздействий, на него оказываемых.

Труд [5] используется как источник информации о различных фундаментальных математических понятиях, в частности о разложении функций многих переменных в степенные ряды, которое используется для упрощения модели динамики морского судна.

В учебном пособии [6] изложен материал, касающийся управления динамическими объектами в условиях неопределённости (отсутствия информации о точных значениях параметров объекта), который напрямую не относится к решаемой задаче, однако помогает оценить некоторые возможности для модификации предлагаемого подхода.

На основании результатов, полученных в диссертации [7], возможно проанализировать перспективы для поиска теоретического обоснования разработанного метода решения поставленной задачи.

В книге [8], помимо доступно изложенных принципов теории устойчивости движения, также содержится большое количество полезных сведений о синтезе законов управления для нелинейных систем.

Основным источником информации о синтезе законов управления (основанных на принципе обратной связи) для линейных стационарных систем является работа [9]. В ней подробно излагаются как базовые для

данной области понятия, так и различные современные методы, к которым приведены примеры их компьютерной реализации в среде MATLAB.

Полезные сведения по методам анализа и синтеза систем автоматического управления, подверженных детерминированным и случайным воздействиям, изложены в работе [10].

Построение многоцелевого закона управления осуществляется на основании материала, изложенного в статьях [11, 12]. В процессе построения закона используются работы [13, 14].

Результаты имитационного компьютерного моделирования помогла оценить (с позиции теории устойчивости) информация, изложенная в работе [17].

Ознакомление с трудами [18, 19, 20, 21, 22] позволило расширить понимание современных тенденций в области автоматического управления морскими подвижными объектами.

Глава 1. Описание математических моделей

Создание рентабельной системы управления во многом зависит от качества математической модели, описывающей как объект управления, так и воздействия, которым он подвергается. В «хорошей» модели должен быть достигнут определённый баланс между точностью и простотой. Очевидно, что без достаточной точности математической модели созданная для неё система управления не будет подходить к работе в реальных условиях. Однако излишне сложная модель может создать непреодолимые препятствия при разработке управляющей системы. Тем не менее, ввиду существования способов, позволяющих синтезированному для упрощённой модели закону управления сообщать нужные свойства модели исходной, первоначально имеет смысл моделировать реальный объект наиболее точно.

1.1. Математическая модель движения судна

Математическое моделирование динамики таких сложных подвижных объектов как морское судно является трудоёмким и нетривиальным процессом, особенно если к модели предъявляются высокие требования соответствия реальному объекту. Таким образом, её создание потребовало бы глубокого изучения теоретического аспекта вопроса и заняло большое количество времени. Поэтому в дальнейшем будет использована, с некоторыми модификациями, готовая высококачественная модель движения морского судна [2, 3].

Для её описания вводится жёстко связанная с судном декартова прямоугольная система координат $Oxyz$. Начало координат расположено в центре масс объекта, направление оси Ox совпадает с направлением движения судна, ось Oy перпендикулярна оси Ox в плоскости движения, а ось Oz перпендикулярна двум предыдущим. Орты осей системы координат представляют собой правую тройку векторов. Теперь можно ввести необходимые обозначения:

- V_x, V_y — проекции вектора линейной скорости на оси Ox, Oy ;
- ω_x, ω_z — проекции вектора угловой скорости на оси Ox, Oz ;
- ξ — продольное (по направлению движения) смещение центра масс;

- ζ — боковое смещение центра масс;
- θ — угол крена;
- φ — угол курса — угол между ортом оси Ox и ортом оси Ox_0 некоторой неподвижной системы координат;
- Q_x, Q_y, N_x, N_z — проекции совокупности сил и моментов, действующих на судно, на соответствующие оси связанной системы координат;
- F_x, F_y, M_x, M_z — проекции векторов внешней возмущающей силы и её момента на соответствующие оси связанной системы координат (эта сила порождена как волнением, так и воздействиями аэродинамической природы);
- X_H, Y_H, M_{xH}, M_{zH} — проекции гидродинамических сил и момента вязкостной природы, действующих на корпус судна;
- X_R, Y_R, M_{xR}, M_{zR} — проекции сил и моментов, обусловленных перекладкой вертикального руля;
- Ω — безразмерная угловая скорость рыскания;
- T_v, ν — тяга и число оборотов гребного винта;
- β — угол дрейфа;
- δ_v — угол перекладки вертикального руля,

при этом вектором состояния объекта управления является вектор

$$\bar{x} = \left(V_x \ V_y \ \omega_x \ \omega_z \ \xi \ \zeta \ \theta \ \varphi \ \delta_v \right)^T.$$

В модели фигурируют физические постоянные, а также константы, характеризующие конкретное судно:

- $g = 9,81$ — ускорение свободного падения;
- $m = 402,68$ — масса судна;
- $L = 115$ — длина судна;
- $L_R = 52,75$ — плечо руля;
- $\mu(\theta_0) = 0,34$ — коэффициент демпфирования бортовой качки;

- $V_p = 15,2$ — скорость хода в установившемся режиме (максимальная);
- $h_0 = 1,22$ — поперечная метацентрическая высота;
- $z_k = 3,3$ — плечо сил инерции;
- $J_{xx} = 6573,5$, $J_{zz} = 266270$ — моменты инерции относительно осей Ox, Oz ;
- $k_{11} = 0,02$, $k_{22} = 0,616$ — коэффициенты присоединённой массы вдоль осей Ox, Oy ;
- $k_{24} = 0,31$ — коэффициент присоединённой массы вдоль оси Oy и Oz ;
- $k_{44} = 0,503$, $k_{66} = 0,527$ — коэффициенты присоединённого момента инерции относительно осей Ox, Oz .

Динамика объекта управления представлена системой из восьми нелинейных дифференциальных уравнений первого порядка, а также уравнений привода вертикального руля:

$$\left\{ \begin{array}{l} \dot{V}_x = (m + k_{11}m)^{-1}Q_x, \\ \dot{V}_y = (J_{xx}(m + k_{22}m)(1 + k_{44}) - m^2k_{24}^2)^{-1}(Q_y(1 + k_{44})J_{xx} - N_xk_{24}m), \\ \dot{\omega}_x = (J_{xx}(m + k_{22}m)(1 + k_{44}) - m^2k_{24}^2)^{-1}(N_x(m + k_{22}m) - Q_yk_{24}m), \\ \dot{\omega}_z = (J_{zz} + k_{66}J_{zz})^{-1}N_z, \\ \dot{\xi} = V_x \cos\varphi + V_y \cos\theta \sin\varphi, \\ \dot{\zeta} = -V_x \sin\varphi + V_y \cos\theta \cos\varphi, \\ \dot{\theta} = \omega_x, \\ \dot{\varphi} = \omega_z, \\ \delta = \begin{cases} 3 \operatorname{sign}(u), & |u| > 3^\circ/c, \\ u, & |u| \leq 3^\circ/c, \end{cases} \quad \delta_v = \begin{cases} 30 \operatorname{sign}(\delta), & |\delta| > 30^\circ, \\ \delta, & |\delta| \leq 30^\circ. \end{cases} \end{array} \right.$$

Здесь u — управляющее воздействие, подающееся на привод судна,

$$Q_x = 1,8T_v - X_H - X_R + (m + k_{22}m)\omega_z V_y + F_x,$$

$$Q_y = Y_H + Y_R + F_y,$$

$$N_x = -\mu\omega_x J_{xx} - mgh_0\theta + z_k m\omega_z V + M_{xH} + M_{xR} + M_x,$$

$$N_z = M_{zH} + M_{zR} + M_z,$$

$$\begin{aligned}
X_H &= 0,5775(1 - \Omega^2)V_L^2, \\
X_R &= 0,462\alpha^2V_{LR}^2, \\
Y_H &= 4,89V_L^2\beta - 4,39V_L^2\Omega\sqrt{1 - \Omega^2} + 23,7V_L^2\beta|\beta|, \\
Y_R &= 1,236V_{LR}^2\beta + 0,567V_{LR}^2\omega - 1,236V_{LR}^2\delta_v, \\
M_{xH} &= -11,39V_L^2\beta + 10,23V_L^2\Omega\sqrt{1 - \Omega^2} - 55,23V_L^2\beta|\beta|, \\
M_{xR} &= -3,91V_{LR}^2\beta - 1,79V_{LR}^2\omega + 3,91V_{LR}^2\delta_v, \\
M_{zH} &= 260V^2\beta - 223,6V_L^2\Omega + 69,1V_L^2\Omega|\beta| - 160,8V_L^2\Omega|\Omega|, \\
M_{zR} &= -65,2V_{LR}^2\beta - 29,9V_{LR}^2\omega + 65,2V_{LR}^2\delta_v, \\
T_v &= 9,74\nu^2 - 2,23\nu V,
\end{aligned}$$

где, в свою очередь,

$$\begin{aligned}
V &= \sqrt{V_x^2 + V_y^2}, \quad V_L = \sqrt{V^2 + L^2\omega_z^2}, \quad \beta = -\arctg\frac{V_y}{V_x}, \quad \nu = \frac{V_p}{3,039}, \\
\Omega &= \frac{\omega_z L}{V_L}, \quad V_{LR} = \sqrt{V^2 + L_R^2\omega_z^2}, \quad \omega = \frac{\omega_z L}{V_{LR}}, \quad \alpha = \delta_v - \frac{\omega_z L_R}{V} - \beta.
\end{aligned}$$

Несмотря на то, что в имитационном моделировании используется вышеуказанная модель, для дальнейшей работы требуется лишь та её часть, которая описывает движение судна в горизонтальной плоскости:

$$\begin{cases}
\dot{V}_y = \frac{Q_y(1 + k_{44})J_{xx} - N_x k_{24}m}{J_{xx}(m + k_{22}m)(1 + k_{44}) - m^2 k_{24}^2}, \\
\dot{\omega}_z = (J_{zz} + k_{66}J_{zz})^{-1}N_z, \\
\dot{\varphi} = \omega_z, \\
\dot{\delta} = \begin{cases} 3\text{sign}(u), & |u| > 3^\circ/c, \\ u, & |u| \leq 3^\circ/c, \end{cases} \quad \delta_v = \begin{cases} 30\text{sign}(\delta), & |\delta| > 30^\circ, \\ \delta, & |\delta| \leq 30^\circ \end{cases}
\end{cases} \quad (3)$$

В качестве вектора состояния рассматривается, соответственно, вектор

$$x = \left(V_y \quad \omega_z \quad \varphi \quad \delta_v \right)^T \quad (4)$$

Стоит также упомянуть, что включать в систему (3) динамику, связанную с креном, не имеет смысла, так как она достаточно слабо влияет на стабилизацию движения морского судна по курсу.

1.2. Математическая модель бокового ветра

Построение математической модели бокового ветра [2, 4] осуществляется исходя из физических соображений и, разумеется, является намного более простой задачей, чем моделирование сложной динамической системы наподобие морского судна.

Боковой ветер характеризуется двумя параметрами: V_w и A_w — величиной скорости ветра и углом между ортом оси Ox_0 неподвижной системы координат и вектором $-\vec{V}_w$.

Влияние бокового ветра на движение морского судна в горизонтальной плоскости выражается через проекцию на ось Oy силы, с которой он действует на корпус судна, и проекцию создаваемого этой силой момента на ось Oz . Эти проекции ранее были обозначены как F_y и M_z соответственно и имеют следующий вид:

$$F_y = 0,076\Theta \sin\Gamma, \quad M_z = -8,72\Theta \sin\Gamma \left(0,273 - \frac{\Gamma}{2\pi}\right).$$

Здесь

$$\begin{aligned}\Theta &= V_w^2 + V^2 + 2V_w V \cos(A_c + \beta), \\ \Gamma &= \arctg \frac{V \sin\beta - V_w \sin A_c}{V \cos\beta + V_w \cos A_c}, \\ \beta &= -\arcsin \frac{V_y}{V}, \quad A_c = A_w - \varphi.\end{aligned}$$

Суммарный поток ветра на корпусе морского судна показан на рис. 1. Используются следующие обозначения:

- Θ — квадрат модуля вектора \vec{V}_Γ , где \vec{V}_Γ — вектор скорости потока воздуха относительно корпуса судна;
- Γ — угол встречи с набегающим потоком, то есть угол между векторами \vec{V}_x и $-\vec{V}_\Gamma$;
- β — угол дрейфа, то есть угол между ортом оси Ox и вектором скорости судна \vec{V} ;
- A_c — угол между векторами \vec{V}_x и $-\vec{V}_w$.

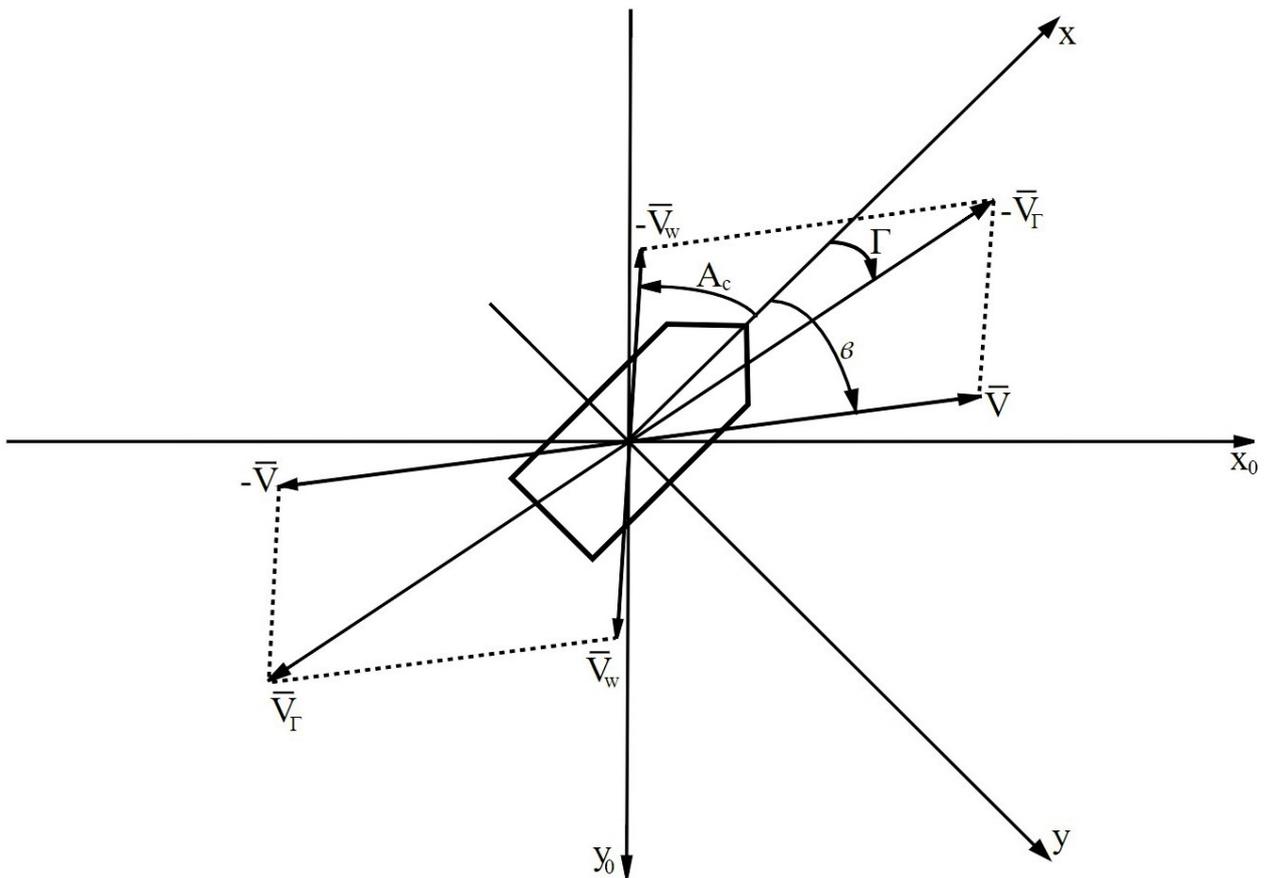


Рис. 1. Суммарный воздушный поток на корпусе морского судна

Все остальные величины, задействованные при моделировании, имеют ранее указанный смысл.

Нетрудно заметить, что выразив момент M_z , создаваемый возмущающей силой F_y , через саму F_y , можно несколько упростить модель:

$$F_y = 0,076\Theta \sin\Gamma, \quad M_z = -\frac{8,72}{0,076} \left(0,273 - \frac{\Gamma}{2\pi}\right) F_y = h(\Gamma) F_y = h(t) F_y. \quad (5)$$

Итак, описание математической модели объекта управления, равно как и воздействующих на него внешних возмущений, можно считать выполненным.

Глава 2. Синтез закона управления

Как уже было сказано при постановке задачи, необходимо стабилизировать программное движение морского судна в горизонтальной плоскости. Для моделирования рассматриваемого типа движения используется система (3). Вектором состояния данной системы является вектор (4), однако для измерения доступны только угол курса φ и угол перекладки вертикального руля δ_v (за исключением начального момента времени, в который известна вся информация о состоянии судна). Таким образом, доступную информацию о состоянии объекта управления можно представить в виде:

$$y = \left(\begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} V_y \\ \omega_z \\ \varphi \end{pmatrix}, \delta_v \right) = \left(C \begin{pmatrix} V_y \\ \omega_z \\ \varphi \end{pmatrix}, \delta_v \right) = (y_p \ \delta_v). \quad (6)$$

Программный вектор состояния объекта \tilde{x} полностью известен в каждый момент рассматриваемого отрезка времени.

Влияние бокового ветра на морское судно моделируется с помощью уравнений (5). Так как в рассматриваемой задаче объект управления не подвержен другим внешним воздействиям, то силы и моменты, не порождаемые боковым ветром, можно считать нулевыми, то есть $F_x = 0, M_x = 0$. В итоге вектор внешних возмущений имеет вид:

$$D = \begin{pmatrix} 0 & F_y & 0 & M_z \end{pmatrix}^T, \quad \text{или, иначе:} \quad D = \begin{pmatrix} F_y & M_z \end{pmatrix}^T. \quad (7)$$

Компенсация возмущения (7) будет осуществляться для угла курса φ (что влечёт за собой и компенсацию для угловой скорости $\omega_z = \dot{\varphi}$) как для наиболее существенной характеристики движения морского судна в горизонтальной плоскости.

Стоит также упомянуть, что программное движение $\tilde{x}(t)$ является, разумеется, невозмущённым, то есть $\tilde{D} = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}^T$.

2.1. Линеаризация математической модели

Работать с системой (3) в силу её сложной нелинейной структуры крайне затруднительно, поэтому разумно будет произвести упрощение дан-

ной модели при помощи её линеаризации в окрестности программного движения $\tilde{x}(t)$, то есть совершить переход от системы (3) к линейной системе в отклонениях от программного движения $\tilde{x}(t)$.

Для этого правые части уравнений системы (3), представленные как функции от компонент вектора состояния x и вектора возмущений D , раскладываются в ряд Тейлора для функции многих переменных [5] в окрестности программного движения $\tilde{x}(t)$ и программного возмущения \tilde{D} , затем от ряда отбрасываются все члены с частными производными порядка большего, чем первый.

После описанных действий полученная система в отклонениях имеет следующий вид:

$$\dot{\hat{x}} = A(t)\hat{x} + B\hat{u} + MD. \quad (8)$$

Здесь $\hat{x} = x - \tilde{x}$, $\hat{u} = u - \tilde{u}$,

$$A(t) = \left(\begin{array}{cccc} \frac{\partial f_1}{\partial V_y} & \frac{\partial f_1}{\partial \omega_z} & 0 & \frac{\partial f_1}{\partial \delta_v} \\ \frac{\partial f_2}{\partial V_y} & \frac{\partial f_2}{\partial \omega_z} & 0 & \frac{\partial f_2}{\partial \delta_v} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array} \right)_{x=\tilde{x}}, \quad B = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \quad M = \begin{pmatrix} \frac{\partial f_1}{\partial F_y} & 0 \\ 0 & \frac{\partial f_2}{\partial M_z} \\ 0 & 0 \\ 0 & 0 \end{pmatrix},$$

где, в свою очередь, f_i — обозначение для правой части i -ого уравнения системы (3).

Воспользовавшись видом уравнений (5), вектор возмущений (7) можно заменить на скаляр $d = F_y$, изменив матрицу M соответствующим образом:

$$M(t) = \left(\begin{array}{cccc} \frac{\partial f_1}{\partial F_y} & \frac{\partial f_2}{\partial M_z} h(t) & 0 & 0 \end{array} \right)^T. \quad (9)$$

В итоге система (8), дополненная уравнением выхода (6), принимает вид:

$$\begin{cases} \dot{\hat{x}} = A(t)\hat{x} + B\hat{u} + M(t)d, \\ \hat{y} = \left((C \ 0)\hat{x} \ \delta_v - \tilde{\delta}_v \right) = \left(\hat{y}_p \ \hat{\delta}_v \right). \end{cases} \quad (10)$$

Итак, для дальнейшей работы по синтезу закона управления в качестве модели движения судна в горизонтальной плоскости будет рассматриваться система (10). Полученные результаты, однако, будут применены для стабилизации программного движения модели (3).

2.2. Синтез базового закона управления

Базовый закон управления призван стабилизировать программное движение морского судна в случае отклонения (в некоторых пределах) фактического начального состояния от программного. Другими словами, такой закон должен обеспечить локальную асимптотическую устойчивость [1] системы (3) при отсутствии внешних возмущений.

2.2.1. Описание предлагаемого подхода

Проблема в рассматриваемой задаче заключается в том, что даже после упрощения модели (3) с помощью линейного приближения, полученная модель (10) является нестационарной, то есть изменяется с течением времени. К сожалению, в настоящий момент отсутствуют методы со строгим теоретическим обоснованием, позволяющие синтезировать требуемый закон в подобной ситуации.

В данной работе предлагается следующий подход:

1. рассматривать в отдельно взятый момент времени модель (3) (а соответственно, и модель (10)) в качестве стационарной;
2. синтезировать для полученной стационарной системы необходимый закон управления;
3. замыкать найденным законом объект управления;
4. повторять пункты 1 — 3 после перехода к следующему моменту времени.

Так как модель, описывающая такой массивный тяжёлый подвижный объект как фрегат, является робастной [6] (вариация параметров этой модели в определённых пределах не меняет её свойств), то у закона управления будет присутствовать некоторый запас устойчивости, поэтому рассматриваемые моменты времени не обязательно должны быть соседними. Однако в данной работе закон управления синтезируется в каждый момент времени ввиду наличия в этом случае перспектив для поиска теоретического обоснования [7].

Описанный метод позволяет, согласно теореме Ляпунова А. М. об устойчивости по первому приближению [8], свести синтез базового закона

управления к поиску в каждый конкретный момент времени управляющего воздействия, обеспечивающего расположение в левой открытой полуплоскости комплексной плоскости всех собственных чисел матрицы системы следующего вида:

$$\dot{\hat{x}} = A\hat{x} + B\hat{u}, \quad (11)$$

то есть рассмотренной в конкретный момент времени системы (10) при возмущении $d = 0$.

Закон управления будет строиться по принципу обратной связи, то есть зависеть от состояния объекта управления:

$$\hat{u} = -K\hat{x}, \quad (12)$$

где матрица $K = \begin{pmatrix} k_1 & k_2 & k_3 & k_4 \end{pmatrix}$. Такой подход позволяет управляющей системе реагировать в режиме реального времени на нежелательное изменение вектора состояния судна.

Итак, в каждый момент времени необходимо найти матрицу K , обеспечивающую расположение в левой открытой полуплоскости комплексной плоскости всех собственных чисел матрицы системы

$$\dot{\hat{x}} = (A - BK)\hat{x}, \quad (13)$$

полученной замыканием системы (11) законом управления (12).

2.2.2. Построение асимптотического наблюдателя

Как видно из уравнения (12), для синтеза закона управления необходимо знать вектор $\hat{x} = x - \tilde{x}$. Программный вектор состояния \tilde{x} известен в каждый момент рассматриваемого отрезка времени, однако информация о векторе x — фактическом состоянии объекта — имеется только в виде (6), то есть боковая скорость V_y и угловая скорость ω_z не измеряются.

Эффективным аналитическим подходом к преодолению данного затруднения является построение оценки вектора \hat{x} — вектора \hat{z} , при этом на отклонение реальных значений от приближённых накладывается условие:

$$\hat{\varepsilon} = \hat{x} - \hat{z} \longrightarrow \mathbf{0} \text{ при } t \longrightarrow +\infty. \quad (14)$$

Описанную оценку позволяет осуществить специальная линейная стационарная система — асимптотический наблюдатель состояния [9] системы

(11). Достаточно, однако, искать приближение не всего вектора \hat{x} , а только первых трёх его компонент, а к получившемуся вектору оценки — вектору \hat{z}_p — добавлять измеряемую четвёртую компоненту вектора \hat{x} , то есть

$$\hat{z} = \begin{pmatrix} \hat{z}_p & \hat{\delta}_v \end{pmatrix}^T$$

Таким образом, асимптотический наблюдатель будет строиться не для системы (11), а для её части, описывающей динамику первых трёх компонент вектора \hat{x} :

$$\dot{\hat{x}}_p = A_p \hat{x}_p + B_p \hat{\delta}_v. \quad (15)$$

Здесь

$$A_p = \begin{pmatrix} \frac{\partial f_1}{\partial V_y} & \frac{\partial f_1}{\partial \omega_z} & 0 \\ \frac{\partial f_2}{\partial V_y} & \frac{\partial f_2}{\partial \omega_z} & 0 \\ 0 & 1 & 0 \end{pmatrix}_{x=\tilde{x}, t=t_k}, \quad B_p = \begin{pmatrix} \frac{\partial f_1}{\partial \delta_v} \\ \frac{\partial f_2}{\partial \delta_v} \\ 0 \end{pmatrix}_{x=\tilde{x}, t=t_k}, \quad \hat{x}_p = \begin{pmatrix} V_y - \tilde{V}_y \\ \omega_z - \tilde{\omega}_z \\ \varphi - \tilde{\varphi} \end{pmatrix},$$

где, в свою очередь, f_i — обозначение для правой части i -ого уравнения системы (3), t_k — рассматриваемый момент времени.

Динамические уравнения наблюдателя состояния системы (15) имеют следующий вид:

$$\dot{\hat{z}}_p = A_p \hat{z}_p + B_p \hat{\delta}_v + G(\hat{y}_p - C\hat{z}_p), \quad (16)$$

где $C = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}$, $\hat{y}_p = C\hat{x}_p$, $G = \begin{pmatrix} g_1 & g_2 & g_3 \end{pmatrix}^T$. Матрица G строится таким образом, чтобы выполнялось требование:

$$\hat{\varepsilon}_p = \hat{x}_p - \hat{z}_p \longrightarrow \mathbf{0} \text{ при } t \longrightarrow +\infty. \quad (17)$$

Очевидно, что если оценка \hat{z}_p удовлетворяет условию (17), то оценка \hat{z} удовлетворяет условию (14).

Вычитанием из уравнений (15) уравнений (16) получается система, описывающая динамику ошибки $\hat{\varepsilon}_p$:

$$\dot{\hat{\varepsilon}}_p = (A_p - GC)\hat{\varepsilon}_p. \quad (18)$$

Из вида данной системы следует, что требование (17) эквивалентно требованию асимптотической устойчивости системы (18), поэтому матрица G

строится таким образом, чтобы обеспечить расположение в левой открытой полуплоскости комплексной плоскости всех собственных чисел (полюсов) матрицы $A_p - GC$.

Указанное расположение полюсов достигается с использованием метода, называемого «модальный синтез» [9]:

1. вычисляется характеристический полином матрицы $A_p - GC$:

$$\Delta(s) = (Es - A_p + GC), \quad (19)$$

здесь s — скалярный аргумент, E — единичная матрица с размерностью такой же, как и у матрицы A_p ;

2. задаётся полином $\delta(s)$, расположение корней которого на комплексной плоскости совпадает с желаемым расположением полюсов матрицы $A_p - GC$:

$$\delta(s) = (s - p_1)(s - p_2)(s - p_3),$$

где p_i — желаемое значение полюса. Нужные значения собственных чисел задаются масштабированием вглубь левой полуплоскости комплексной плоскости (то есть умножением на некоторое положительное вещественное число, большее единицы) распределения полюсов, полученного в ходе синтеза регулятора;

3. составляется система уравнений, полученных приравниванием коэффициентов при одинаковых степенях переменной s у полиномов $\Delta(s)$ и $\delta(s)$. В матричной форме она имеет следующий вид:

$$MG = L.$$

Здесь

$$M = \begin{pmatrix} c_1 & c_2 & c_3 \\ a_{21}c_2 - a_{22}c_1 & c_3 - a_{11}c_2 + a_{12}c_1 & -a_{11}c_3 - a_{22}c_3 \\ a_{21}c_3 & -a_{11}c_3 & a_{11}a_{22}c_3 - a_{12}a_{21}c_3 \end{pmatrix},$$

$$L = \begin{pmatrix} -p_3 - p_2 - p_1 + a_{22} + a_{11} \\ p_2p_3 + p_1p_3 + p_1p_2 - a_{11}a_{22} + a_{12}a_{21} \\ -p_1p_2p_3 \end{pmatrix},$$

где, в свою очередь, a_{ij} — компонента матрицы A_p с индексом $\{i, j\}$, c_i — компонента матрицы C с индексом $\{i\}$;

4. указанная система уравнений разрешается относительно компонент матрицы G :

$$G = M^{-1}L.$$

Полученная матрица G подставляется в систему (16), в результате интегрирования которой с нулевыми начальными условиями получается искомая оценка \hat{z}_p , дополняемая затем до \hat{z} — приближения вектора \hat{x} в рассматриваемый момент времени.

Построение матрицы G описанным методом эффективно с точки зрения экономии вычислительных ресурсов: достаточно один раз найти компоненты матрицы G в явном виде как функции от компонент матриц A_p , C и корней полинома $\delta(s)$, а затем в процессе компьютерного моделирования просто подставлять в эти функции текущие значения их аргументов.

2.2.3. Синтез регулятора

Имея удовлетворяющую условию (14) оценку \hat{z} вектора \hat{x} , для синтеза базового закона управления (12) осталось построить матрицу K , обеспечивающую расположение в левой открытой полуплоскости комплексной плоскости всех собственных чисел матрицы $A - BK$.

От выбранного при данном построении распределения полюсов зависят свойства полученного закона. Чаще всего характеристиками качества управления являются время достижения желаемой цели и количество ресурсов, затрачиваемых в процессе.

Одним из подходов к выбору расположения собственных чисел является их подбор в ходе ряда экспериментов, также существует некоторое количество стандартных форм распределения полюсов, масштабированием которых можно достичь удовлетворительного качества управления.

Существует, однако, метод, позволяющий при построении регулятора целенаправленно влиять на определённые характеристики закона управления и при этом добиваться оптимального решения в рамках высказанных предпочтений. В данном подходе, называемом задачей линейной интегральной квадратичной оптимизации [9] (LQR-оптимизации, построения

LQR-регулятора), рассматривается интегральный линейно-квадратичный функционал качества следующего вида:

$$J = J(\hat{u}) = J(K) = \int_0^{+\infty} (\hat{z}^T Q \hat{z} + \hat{u}^T R \hat{u}) dt. \quad (20)$$

Здесь матрица $Q \geq 0$, называемая матрицей штрафа за отклонение от программного движения, — знакоположительная, а $R > 0$ — матрица штрафа за использование управления — положительно определённая и симметрическая.

Матрица K находится минимизацией функционала (20) на множестве матриц требуемой размерности с постоянными вещественными коэффициентами, обеспечивающих расположение полюсов матрицы $A - BK$ в левой открытой полуплоскости комплексной плоскости:

$$J(\hat{u}) \xrightarrow{K \in \Omega_K} \min.$$

Указанная минимизация сводится к разрешению матричного алгебраического уравнения Риккати

$$PA + A^T P - PBR^{-1}B^T P + Q = 0$$

относительно матрицы P , после нахождения которой матрица K строится следующим образом:

$$K = -R^{-1}B^T P.$$

Желаемые свойства регулятора достигаются выбором матриц Q и R . Например, при увеличении коэффициентов матрицы R (которая в рассматриваемой задаче является скаляром, так как скалярно управление) происходит сдвиг от увеличения скорости стабилизации к уменьшению затрат на управление.

Итак, после построения приближения вектора \hat{x} и нахождения с помощью LQR-оптимизации матрицы K , задачу синтеза базового закона управления (12) в рассматриваемый момент времени можно считать полностью решённой.

2.3. Компенсация постоянного внешнего возмущения

В данном параграфе систему (10), рассмотренную в конкретный момент времени в качестве стационарной, удобно записать в виде:

$$\begin{cases} \dot{\hat{x}}_p = A_p \hat{x}_p + B_p \hat{\delta}_v + M_p d, \\ \hat{y}_p = C \hat{x}_p, \\ \dot{\hat{\delta}}_v = \hat{u}, \end{cases} \quad (21)$$

где M_p — вектор, состоящий из первых трёх компонент вектора M (вектора коэффициентов при возмущении d). Так как $C = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}$, то выход \hat{y}_p представляет собой отклонение фактического значения угла курса φ от программного.

Базовый закон управления (12) не способен обеспечить астатизм [10] (свойство системы автоматического управления приводить ошибку регулирования к нулю при постоянном внешнем воздействии на данную систему) угла курса φ , поэтому необходима его модификация — создание многоцелевого закона управления [11, 12].

Новый закон управления, обеспечивающий как стабилизацию программного движения $\tilde{x}(t)$, так и компенсацию воздействия бокового ветра на угол курса φ , будет строиться добавлением к закону (12) невязки (умноженной на некоторый искомый скаляр) между выходом \hat{y}_p системы (21) и данными асимптотического наблюдателя, соответствующими этому выходу:

$$\hat{u} = -K_p \hat{z}_p - k_4 \hat{\delta}_v + K_\Delta (\hat{y}_p - C \hat{z}_p). \quad (22)$$

Здесь K_p — вектор, состоящий из первых трёх компонент найденного при построении базового закона управления вектора K , k_4 — четвёртая компонента вектора K .

Коэффициент K_Δ ищется из условия астатизма управляемой системы (21) по отношению к возмущению d : передаточная функция [9] от возмущения к выходу должна быть устойчивой, а её числитель должен иметь нулевой корень. Формально это условие записывается в следующем виде:

$$\hat{y}_p = P(s)d, \quad P(0) = 0, \quad (23)$$

где $P(s)$ — передаточная функция, s — переменная Лапласа.

Таким образом, задача поиска коэффициента K_Δ сводится к построению передаточной функции $P(s)$ с дальнейшим выражением из неё K_Δ на основании условия (23).

Выход \widehat{y}_p можно представить в следующей форме:

$$\widehat{y}_p = C\widehat{x}_p = C(\widehat{\varepsilon}_p + \widehat{z}_p). \quad (24)$$

Здесь $\widehat{\varepsilon}_p$ — погрешность наблюдения. Из равенства (24) видно, что передаточная функция $P(s)$ может быть представлена как сумма передаточных матриц от входа d к $\widehat{\varepsilon}_p$ и \widehat{z}_p , умноженная на матрицу C , поэтому условие (23) принимает вид:

$$\widehat{y}_p = P(s)d = C(F(s) + W(s))d, \quad C(F(0) + W(0)) = 0.$$

С учётом наличия внешнего возмущения d уравнения динамики погрешности наблюдения $\widehat{\varepsilon}_p$ имеют следующий вид:

$$\dot{\widehat{\varepsilon}}_p = (A_p - GC)\widehat{\varepsilon}_p + M_p d. \quad (25)$$

После применения к данной системе преобразования Лапласа [13] она представляется в tf-форме (tf — transfer function — передаточная функция):

$$\widehat{\varepsilon}_p = (Es - A_p + GC)^{-1}M_p d = F(s)d, \quad (26)$$

где E — единичная матрица соответствующей размерности.

Динамика асимптотического наблюдателя (16) и уравнение регулятора (22) могут быть записаны в виде:

$$\begin{cases} \dot{\widehat{z}}_p = A_p \widehat{z}_p + B_p \widehat{\delta}_v + GC \widehat{\varepsilon}_p, \\ \dot{\widehat{\delta}}_v = -K_p \widehat{z}_p - k_4 \widehat{\delta}_v + K_\Delta C \widehat{\varepsilon}_p, \end{cases}$$

Таким образом, динамика компонент вектора \widehat{z} — приближения вектора \widehat{x} — может быть представлена как

$$\dot{\widehat{z}} = A_z \widehat{z} + B_z \widehat{\varepsilon}_p, \quad (27)$$

где

$$A_z = \begin{pmatrix} A_p & B_p \\ -K_p & -k_4 \end{pmatrix}, \quad B_z = \begin{pmatrix} GC \\ K_\Delta C \end{pmatrix}.$$

После применения к системе (27) преобразования Лапласа она также представляется в tf-форме:

$$\widehat{z} = (Es - A_z)^{-1} B_z \widehat{\varepsilon}_p = T(s) B_z \widehat{\varepsilon}_p. \quad (28)$$

Здесь E — единичная матрица размерности такой же, как размерность матрицы A_z . Матрицу $T(s)$ можно представить как блочную:

$$T(s) = \begin{pmatrix} Es - A_p & -B_p \\ K_p & s + k_4 \end{pmatrix}^{-1} = \begin{pmatrix} T_{11}(s) & T_{12}(s) \\ T_{21}(s) & T_{22}(s) \end{pmatrix}, \quad (29)$$

где E — единичная матрица размерности такой же, как размерность матрицы A_p .

После подстановки (26) и (29) в систему (28) можно выписать уравнения для компонент вектора \widehat{z}_p :

$$\widehat{z}_p = (T_{11}(s)GC + T_{12}(s)K_\Delta C)F(s)d = W(s)d. \quad (30)$$

Блоки $T_{11}(s)$ и $T_{12}(s)$ находятся по формуле Фробениуса для обращения блочных матриц [14]:

$$\begin{aligned} T_{11}(s) &= (Es - A_p)^{-1} + (Es - A_p)^{-1}(-B_p)H_f^{-1}K_p(Es - A_p)^{-1}, \\ T_{12}(s) &= -(Es - A_p)^{-1}(-B_p)H_f^{-1}, \end{aligned}$$

где $H_f = (s + k_4) - K_p(Es - A_p)^{-1}(-B_p)$.

После подстановки (26) и (30) в выражение (24) оно принимает следующий вид:

$$\widehat{y}_p = C(E + T_{11}(s)GC + T_{12}(s)K_\Delta C)F(s)d = P(s)d,$$

а условие астатизма (23) может быть переписано как

$$C(E + T_{11}(0)GC + T_{12}(0)CK_\Delta)F(0) = 0,$$

откуда следует, что

$$K_\Delta = -[CEF(0) + CT_{11}(0)GCF(0)][CT_{12}(0)CF(0)]^{-1}. \quad (31)$$

Итак, синтез многоцелевого закона управления (22), стабилизирующего программное движение морского судна при отсутствии полной информации о его состоянии и компенсирующего воздействие внешнего возмущения на угол курса φ , можно считать завершённым.

Глава 3. Имитационное моделирование процессов управления

Имитационное компьютерное моделирование процессов управления реализовано с использованием пакета программного обеспечения MATLAB и его подсистемы Simulink, которые предоставляют широкие возможности для проектирования и анализа систем управления, визуализации полученных результатов, а также обширный набор алгоритмов для работы с различными математическими объектами (функциями, матрицами и так далее).

3.1. Программный комплекс

С целью осуществления имитационного компьютерного моделирования процессов управления создана Simulink-модель, представленная на рисунках 2 – 11.

Общий вид модели в Simulink показан на рис. 2. Параметры модели, такие как программное отклонение вертикального руля $\tilde{\delta}_v(t)$, характеристики внешнего возмущения, начальные условия для программного и фактического движений, наличие или отсутствие возможности измерения компонент вектора состояния объекта управления (матрица C), а также параметры, задающие желаемые свойства регулятора, инициализируются в скрипте из Приложения 1.

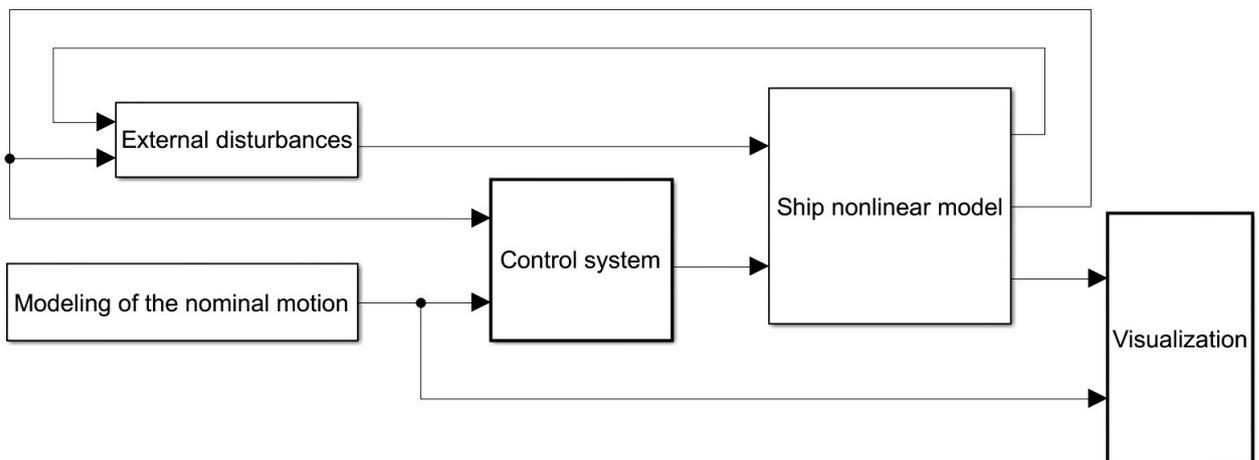


Рис. 2. Общий вид модели в Simulink

Моделирование программного движения $\tilde{x}(t)$, а также линеаризация в его окрестности уравнений системы (3) (вычисление компонент матриц

$A_p(t), B_p(t)$), реализуются в подсистеме «Modeling of the nominal motion», представленной на рис. 3.

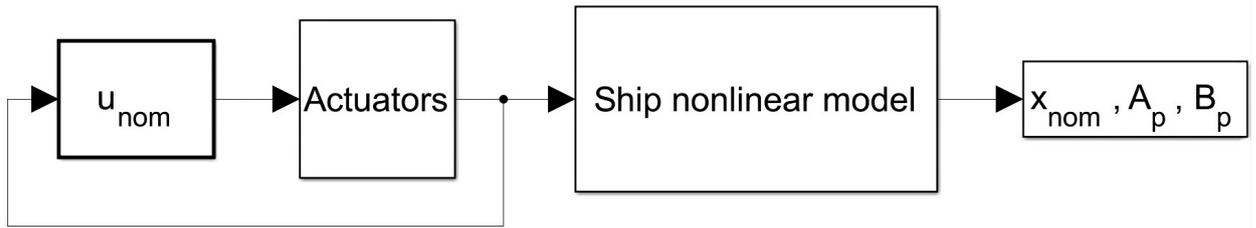


Рис. 3. Моделирование программного движения $\tilde{x}(t)$ и построение матриц $A_p(t), B_p(t)$

Подсистема « u_{nom} », показанная на рис. 4, отвечает за формирование программного управляющего воздействия $\tilde{u}(t)$, получаемого численным дифференцированием программного отклонения вертикального руля $\tilde{\delta}_v(t)$ с последующим добавлением $\tilde{\delta}_v(t)$ для компенсации инерционности привода.

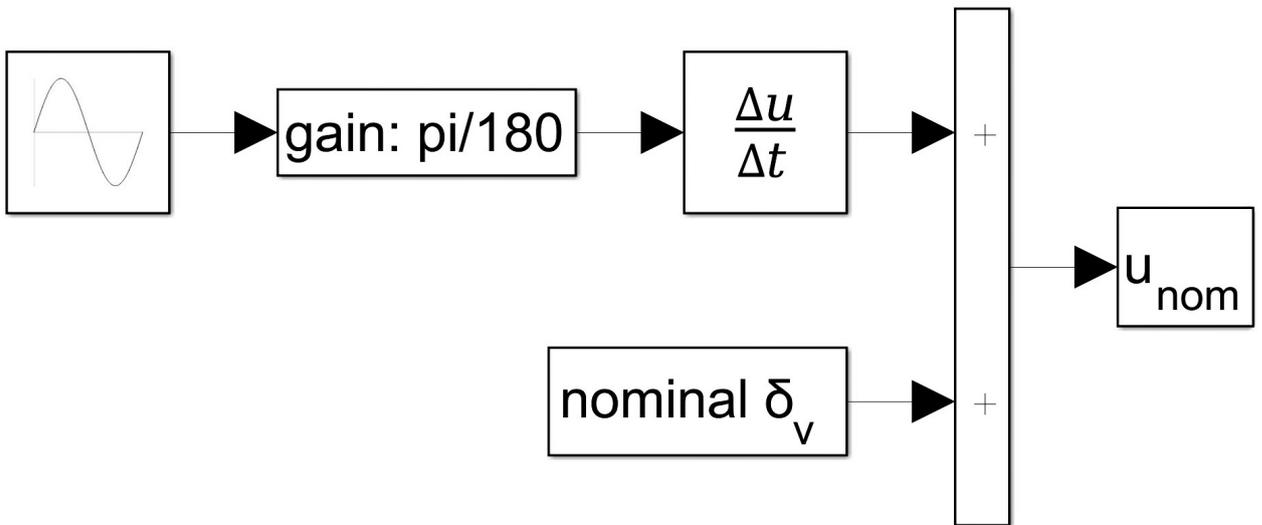


Рис. 4. Формирование программного управления $\tilde{u}(t)$

Привод вертикального руля моделируется в подсистеме «Actuators», представленной на рис. 5.

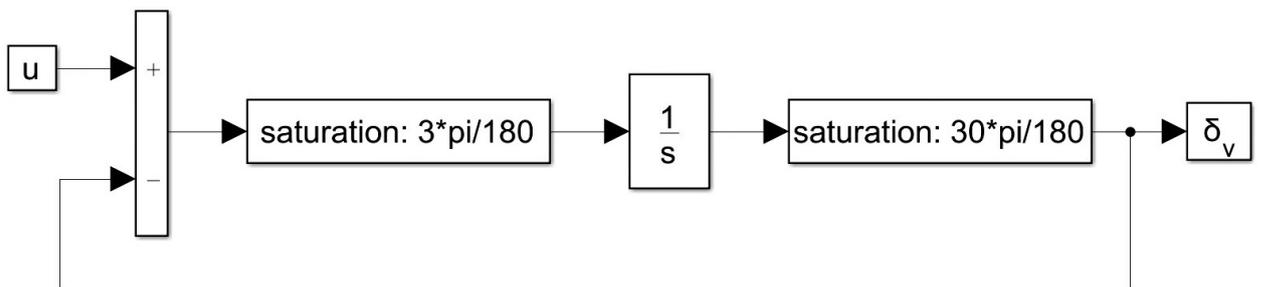


Рис. 5. Модель привода вертикального руля

Особенности конструкции привода порождают ограничения на управление: ограничивается скорость перекладки вертикального руля $u(t)$ ($\tilde{u}(t)$), а также его отклонение $\delta_v(t)$ ($\tilde{\delta}_v(t)$).

Подсистема «Ship nonlinear model» (та её версия, что находится в подсистеме «Modeling of the nominal motion»), показанная на рис. 6, отвечает непосредственно за моделирование программного движения $\tilde{x}(t)$ и линеаризацию в его окрестности системы (3).

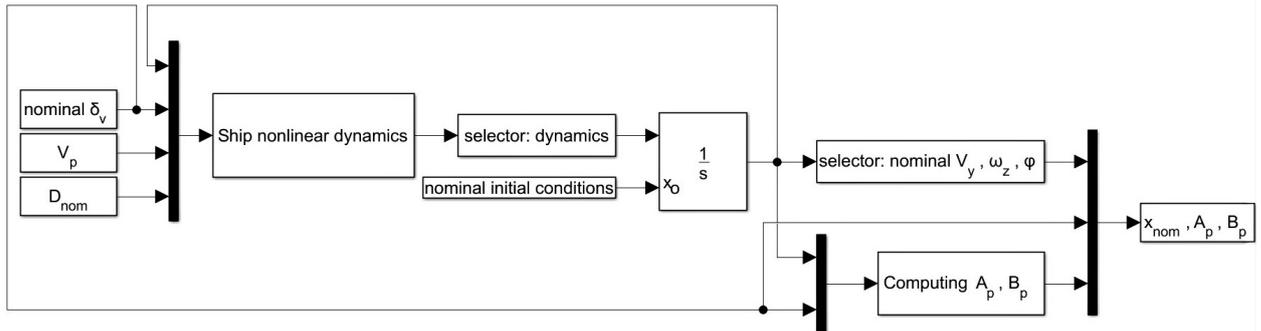


Рис. 6. Модель морского судна при формировании $\tilde{x}(t)$, $A_p(t)$, $B_p(t)$

Блок «Ship nonlinear dynamics» содержит MATLAB-функцию из Приложения 2, в которой реализуется модель динамики морского судна. В блоке «Computing A_p , B_p » осуществляется вычисление текущих значений компонент матриц $A_p(t)$, $B_p(t)$. При этом частные производные уравнений системы (3) заранее находятся в явном виде как функции, зависящие от компонент вектора состояния (4) и вектора возмущения (7), с помощью метода «diff» [15], реализующего алгоритм символьного дифференцирования.

В подсистеме «External disturbances», представленной на рис. 7, происходит моделирование воздействия внешних возмущений на морское судно. Блок «Side wind force and momentum» содержит MATLAB-функцию из Приложения 3, в которой реализуется модель бокового ветра.

Подсистема «Ship nonlinear model», показанная на рис. 8, отвечает за моделирование фактического движения $x(t)$. Сигнал, идущий с первого её выхода (« V_x , V_y »), используется только в подсистеме «External disturbances» для моделирования бокового ветра (так как, вообще говоря, информация о первых двух компонентах вектора состояния $x(t)$ недоступна для измерения). Второй выход (« u_p ») представляет собой измеряемую информацию о состоянии объекта управления. Третий выход (« x , β ») используется исклю-

чительно для визуализации результата моделирования.

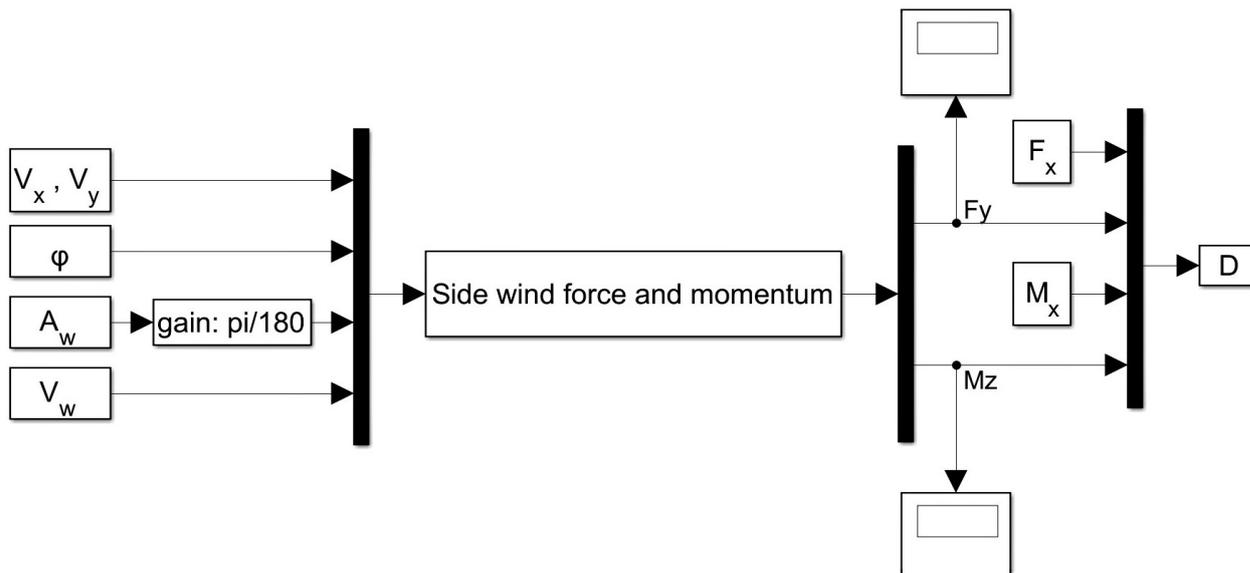


Рис. 7. Моделирование внешнего возмущения $D(t)$

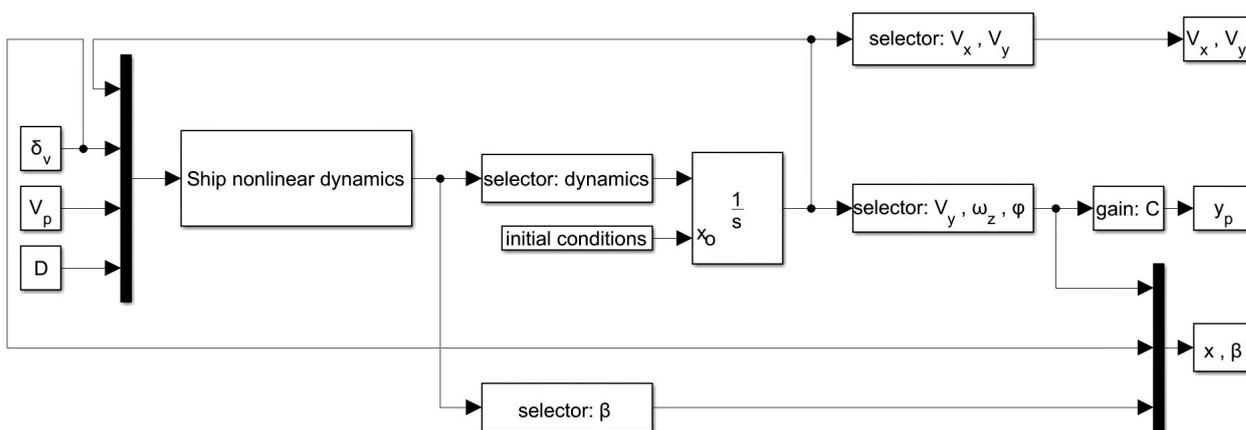


Рис. 8. Simulink-модель морского судна

В подсистеме «Control system», представленной на рис. 9, реализуется система автоматического управления морским судном. Выходом подсистемы является угол перекладки вертикального руля $\delta_v(t)$, подающийся в качестве управления на модель морского судна (подсистему «Ship nonlinear model»).

Стабилизирующее управляющее воздействие $\hat{u}(t)$ находится в подсистеме «Control signal», показанной на рис. 10. Блок «LQR» содержит MATLAB-функцию из Приложения 4, в которой реализован многоцелевой закон управления (22), при этом матрица K находится с помощью LQR-синтеза с использованием метода «lqr» [16]. Блок «Pole placement regulator» также реализует многоцелевой закон управления (22), однако для построения матрицы K используется модальный синтез.

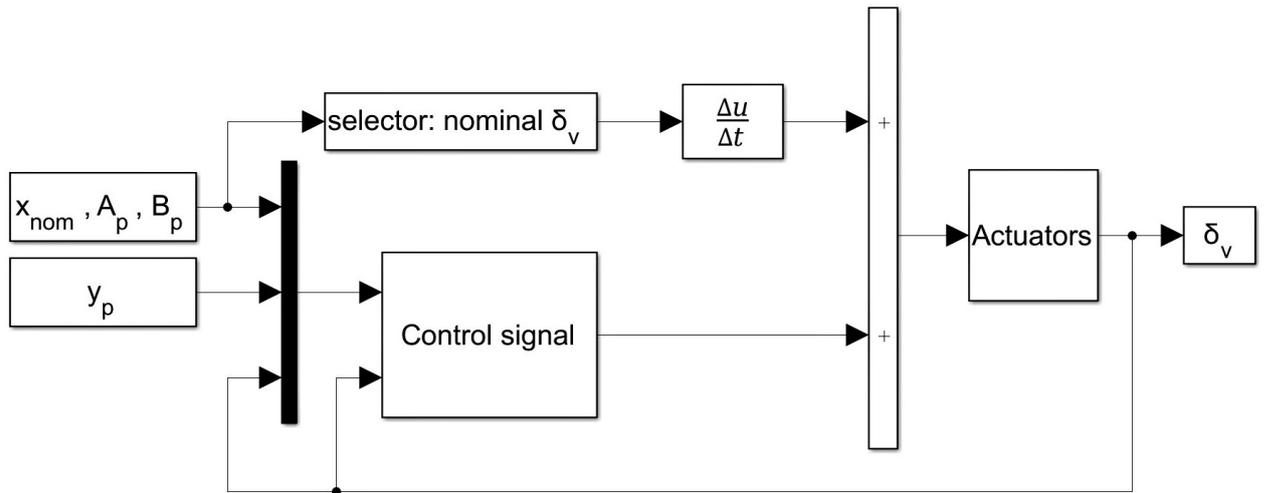


Рис. 9. Система управления (формирование $\delta_v(t)$)

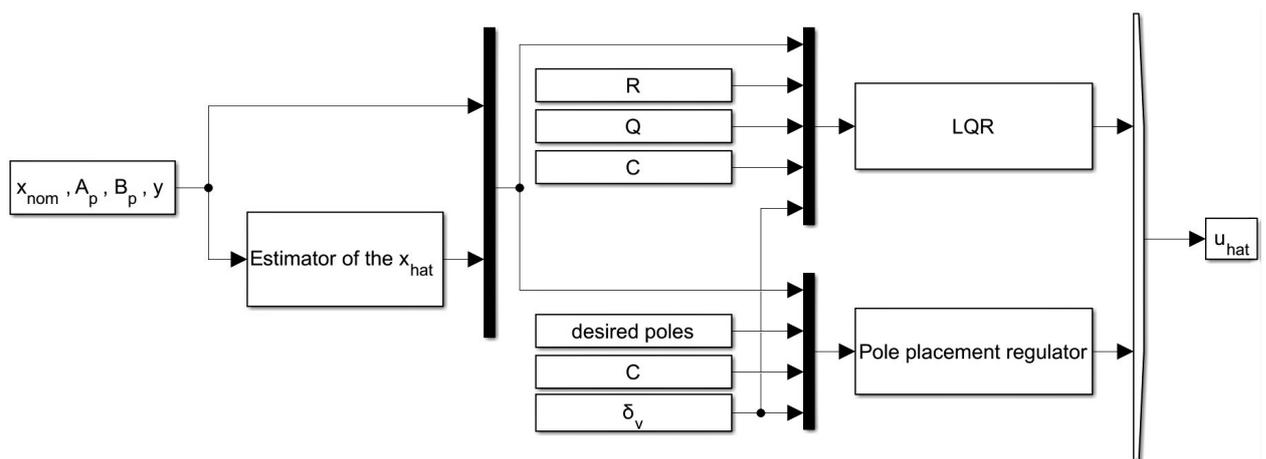


Рис. 10. Синтез стабилизирующего управления $\hat{u}(t)$

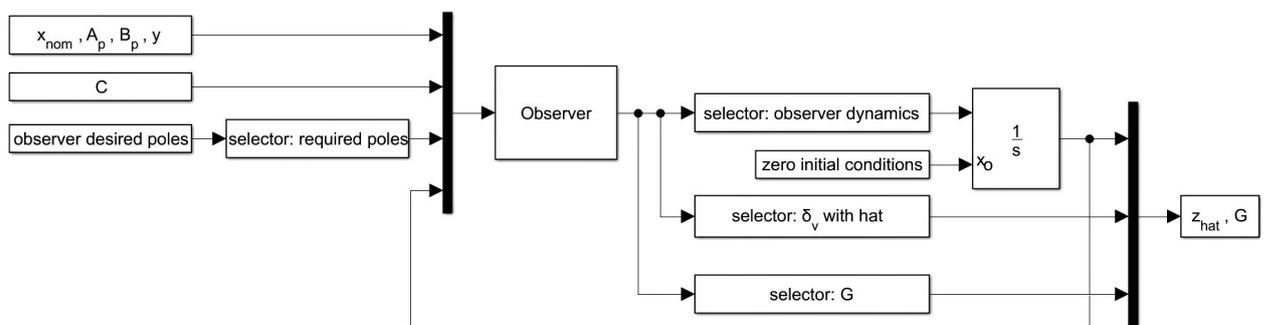


Рис. 11. Построение оценки вектора $\hat{x}(t)$

При формировании стабилизирующего управляющего воздействия $\hat{u}(t)$ в функции из Приложения 4 также учтено свойство инерционности привода вертикального руля (путём прибавления $\delta_v(t)$ к $\hat{u}(t)$).

В подсистеме «Estimator of the x_{hat} », представленной на рис. 11, формируется оценка вектора $\hat{x}(t)$. Блок «Observer» содержит MATLAB-функцию из Приложения 5, в которой реализованы уравнения динамики асимптотического наблюдателя (16).

На этом обзор существенных частей разработанной компьютерной модели можно считать завершённым.

3.2. Примеры имитационного моделирования

Для объективной проверки рентабельности созданной системы автоматического управления необходимо в качестве программного движения выбрать достаточно сложный, нестационарный, но в то же время реалистичный режим функционирования морского судна. В данной работе рассматривается периодическое движение типа «зигзаг», порождаемое программным управлением $\tilde{\delta}_v(t)$ следующего вида:

$$\tilde{\delta}_v(t) = A \cdot \sin(f \cdot t), \quad (32)$$

где A — амплитуда переключки вертикального руля, f — частота переключки вертикального руля, t — время, « \cdot » — символ умножения.

Движение, схожее с описанным, может, например, иметь место при маневрировании морского судна между рифами, айсбергами или иными препятствиями на пути следования объекта.

При синтезе LQR-регулятора весовые матрицы R и Q в функционале качества (20) выбраны следующим образом:

$$R = 0,01, \quad Q = \begin{pmatrix} 0,0001 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Наибольший штраф ($100\hat{\omega}_z^2$) накладывается за отклонение фактической угловой скорости ω_z от программной, также сильное влияние ($\hat{\varphi}^2$) на значение подынтегрального выражения в функционале (20) имеет отклонение

угла курса φ . Величина отклонения фактического угла перекладки вертикального руля δ_v от программного не учитывается вовсе, однако, пусть и не слишком значительный $(0,01\hat{u}^2)$, накладывается штраф за разницу в скорости его перекладки. Отклонение фактической величины боковой скорости V_y от программной оказывает наименьшее влияние $(0,0001\hat{V}_y^2)$ на качество синтезированного закона управления.

В результате анализа распределения собственных чисел матрицы $A - BK$, получаемого LQR-синтезом, расположение полюсов, используемое для модального синтеза в ходе построения матрицы G в асимптотическом наблюдателе (16), было выбрано следующим образом:

$$\begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} = 5 \begin{pmatrix} -1 \\ -1 \\ -0,2 \end{pmatrix}.$$

3.2.1. Стабилизация при отклонении начальных данных

Для проверки эффективности базового закона управления (12) моделируется ситуация, когда начальное состояние для фактического движения (x_0) находится в некоторой окрестности начального состояния для программного движения (\tilde{x}_0), то есть имеет место отклонение начальных условий. При этом для обеспечения чистоты эксперимента внешние возмущения полагаются нулевыми:

$$\begin{aligned} V_w &= 0, & A_w &= 0, \\ \tilde{x}(0) &= \tilde{x}_0 = \left(15,2 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ \tilde{\delta}_v(0) \right)^T, \\ x(0) &= x_0 = \left(15,2 \ 0,5 \ 0,5 \ 0,5 \ 0,5 \ 0,5 \ 0,5 \ 0,5 \ 0,5 \ \delta_v(0) \right)^T, \\ A &= 20, & f &= 0,2. \end{aligned}$$

Переходные процессы по компонентам вектора состояния объекта управления, характеризующим движение в горизонтальной плоскости, показаны на рисунках 12 – 15. На них (и далее) пунктирная линия соответствует программному движению, а сплошная — фактическому.

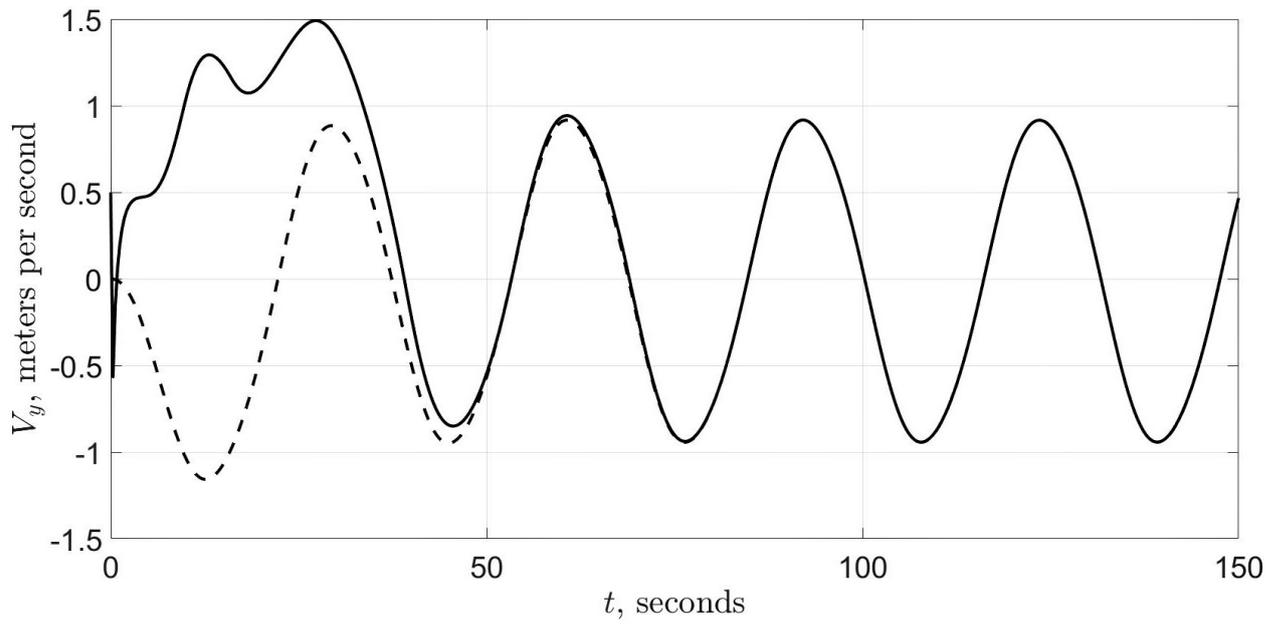


Рис. 12. Переходный процесс по $V_y(t)$

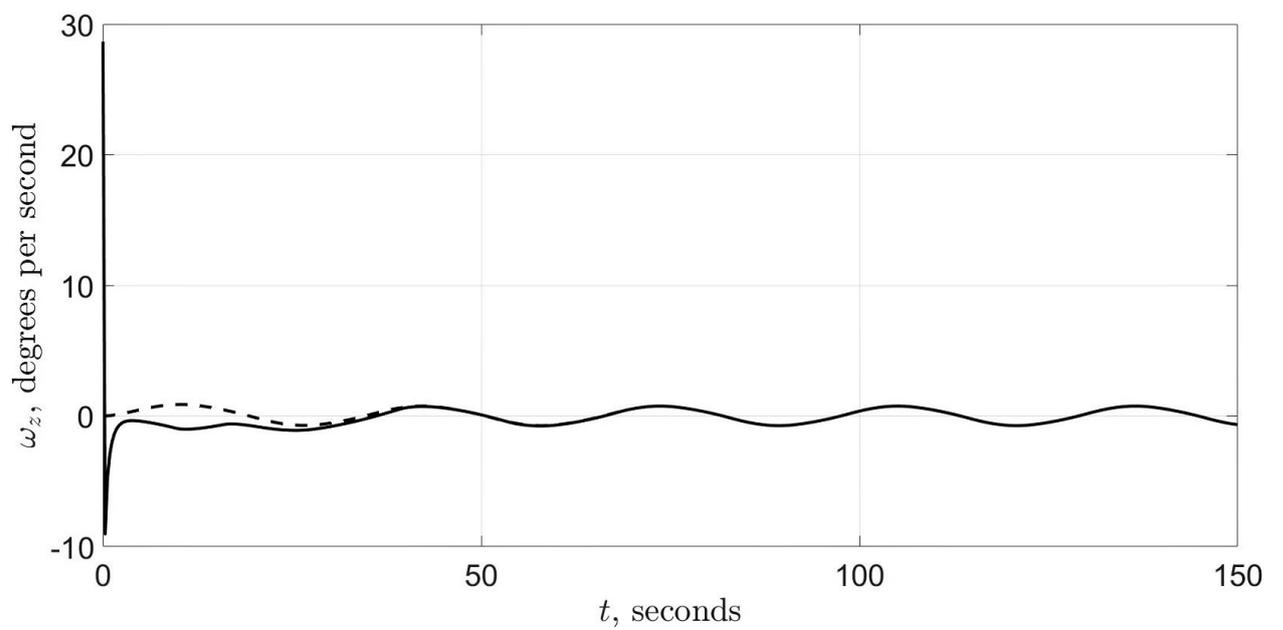


Рис. 13. Переходный процесс по $\omega_z(t)$

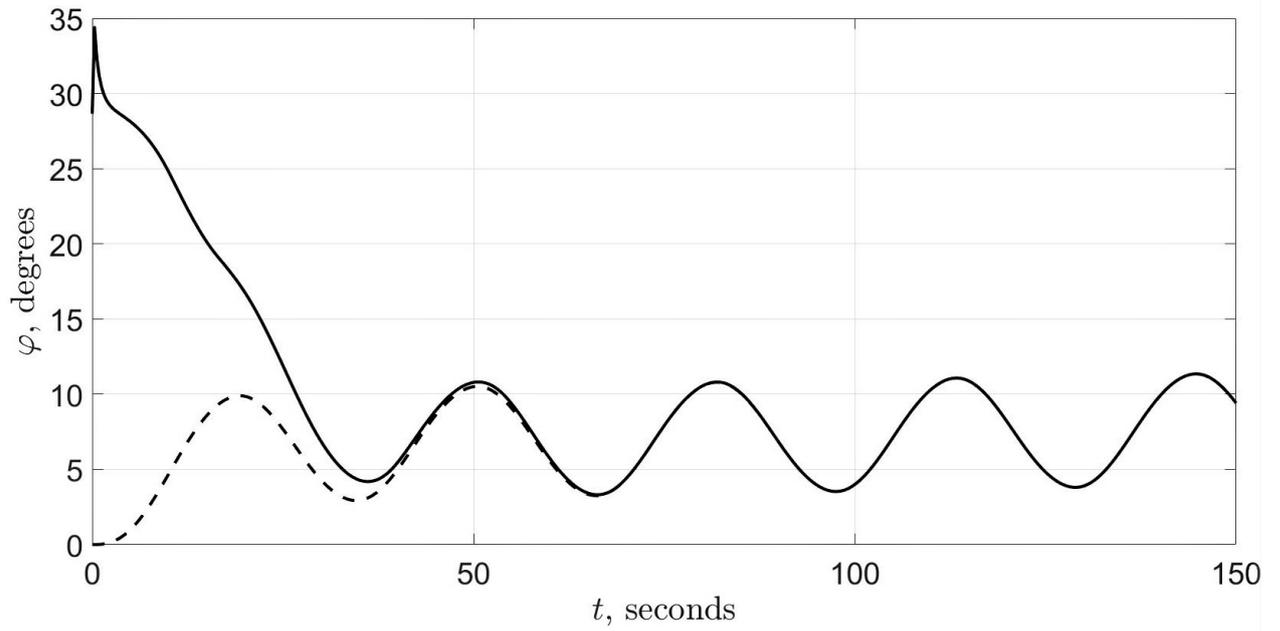


Рис. 14. Переходный процесс по $\varphi(t)$

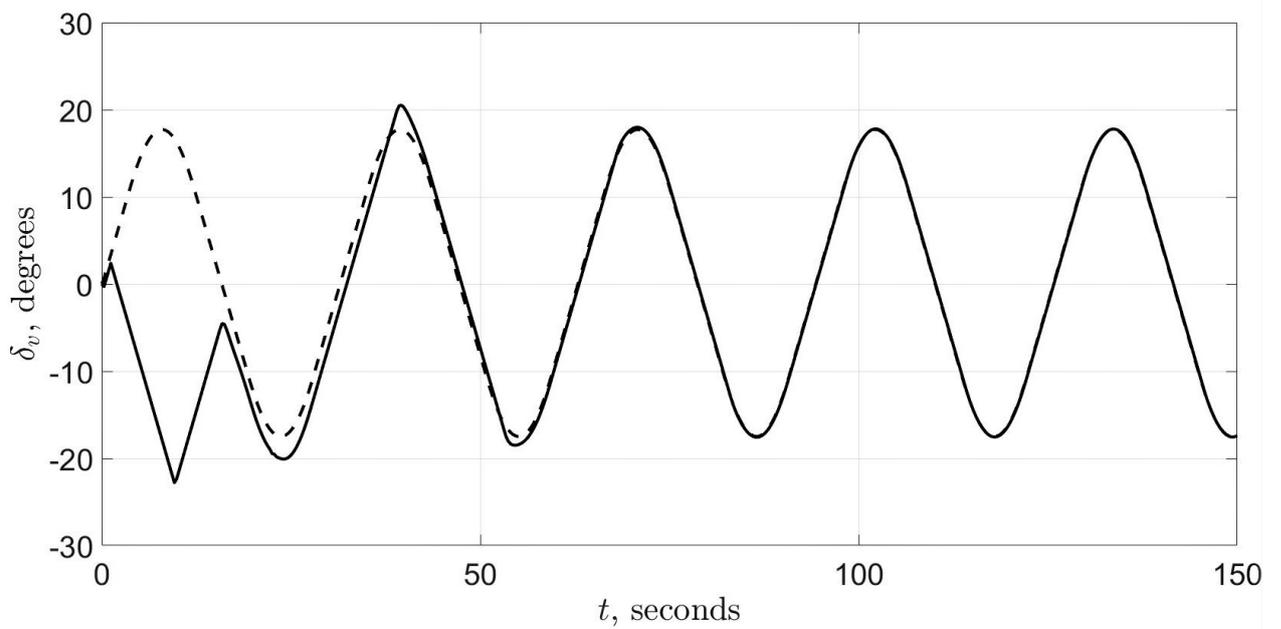


Рис. 15. Переходный процесс по $\delta_v(t)$

На основании приведённых результатов можно заключить, что базовый закон управления (12) способен стабилизировать программное движение морского судна в горизонтальной плоскости в случае отклонения начального состояния x_0 от \tilde{x}_0 . Однако, если задать слишком большое отклонение (начиная, примерно, с $x_0 = \left(15,2 \ 0,6 \ 0,6 \ 0,6 \ 0,6 \ 0,6 \ 0,6 \ 0,6 \ \delta_v(0)\right)^T$ при $\tilde{x}_0 = \left(15,2 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ \tilde{\delta}_v(0)\right)^T$), то, в связи с недостаточной мощностью управления $\delta_v(t)$ ввиду наложенных на него ограничений (2), x_0 окажется вне области асимптотической устойчивости [17] движения $\tilde{x}(t)$, что приведёт к неспособности закона управления выполнить поставленную перед ним задачу.

3.2.2. Компенсация воздействия бокового ветра

Для проверки эффективности многоцелевого закона управления (22) моделируется ситуация, аналогичная описанной выше, однако теперь морское судно оказывается подверженным воздействию бокового ветра:

$$V_w = 20, \quad A_w = 45.$$

Переходные процессы по компонентам вектора состояния объекта управления, характеризующим движение в горизонтальной плоскости, показаны на рисунках 16 – 19.

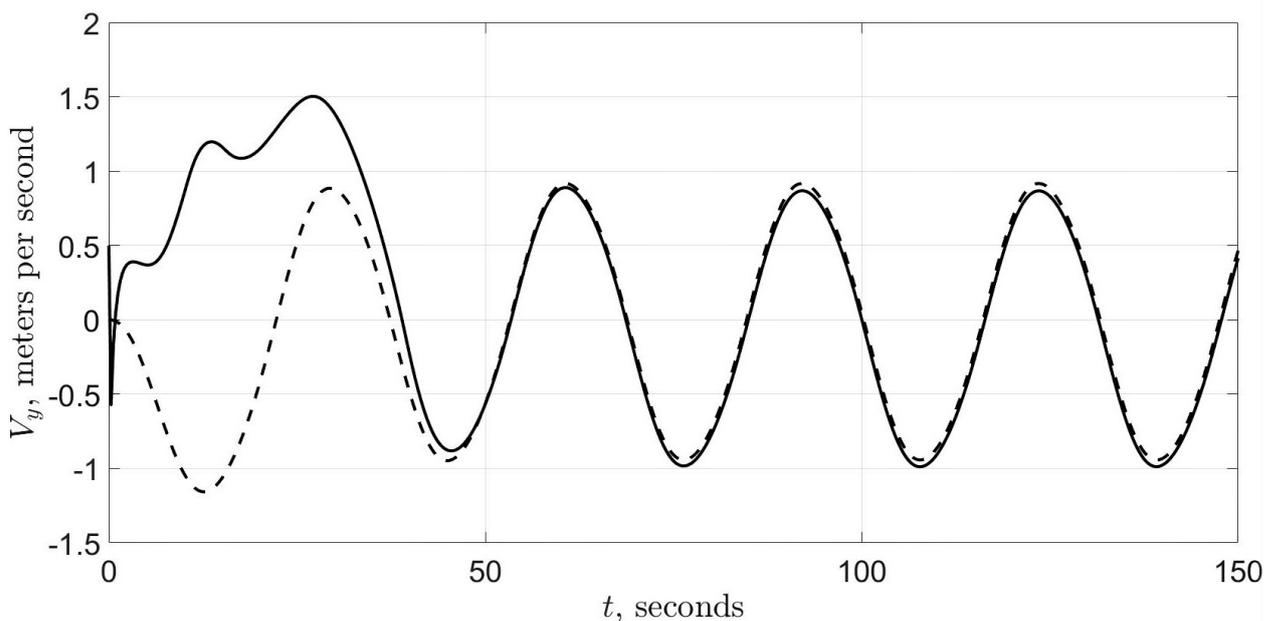


Рис. 16. Переходный процесс по $V_y(t)$

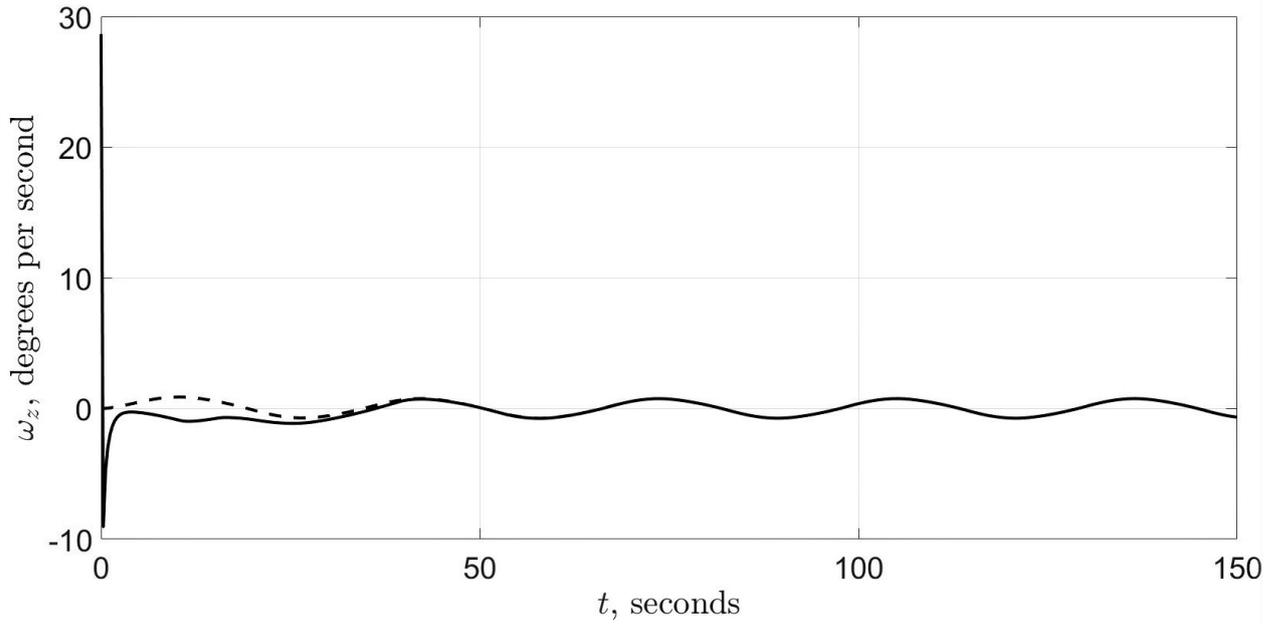


Рис. 17. Переходный процесс по $\omega_z(t)$

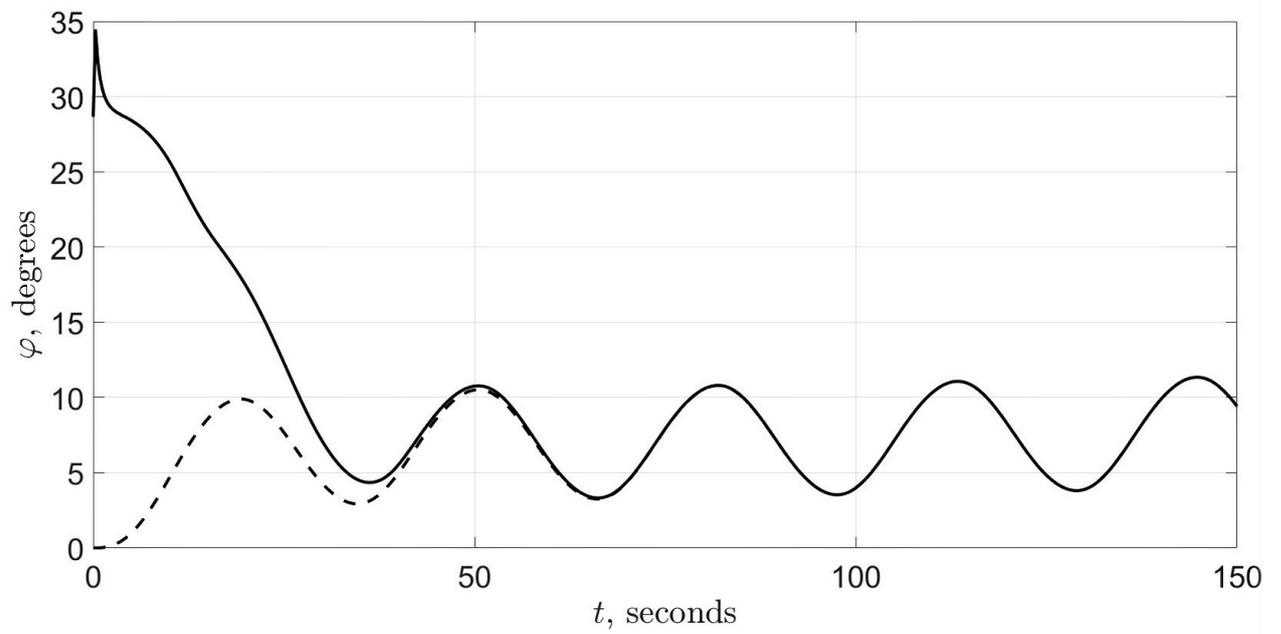


Рис. 18. Переходный процесс по $\varphi(t)$

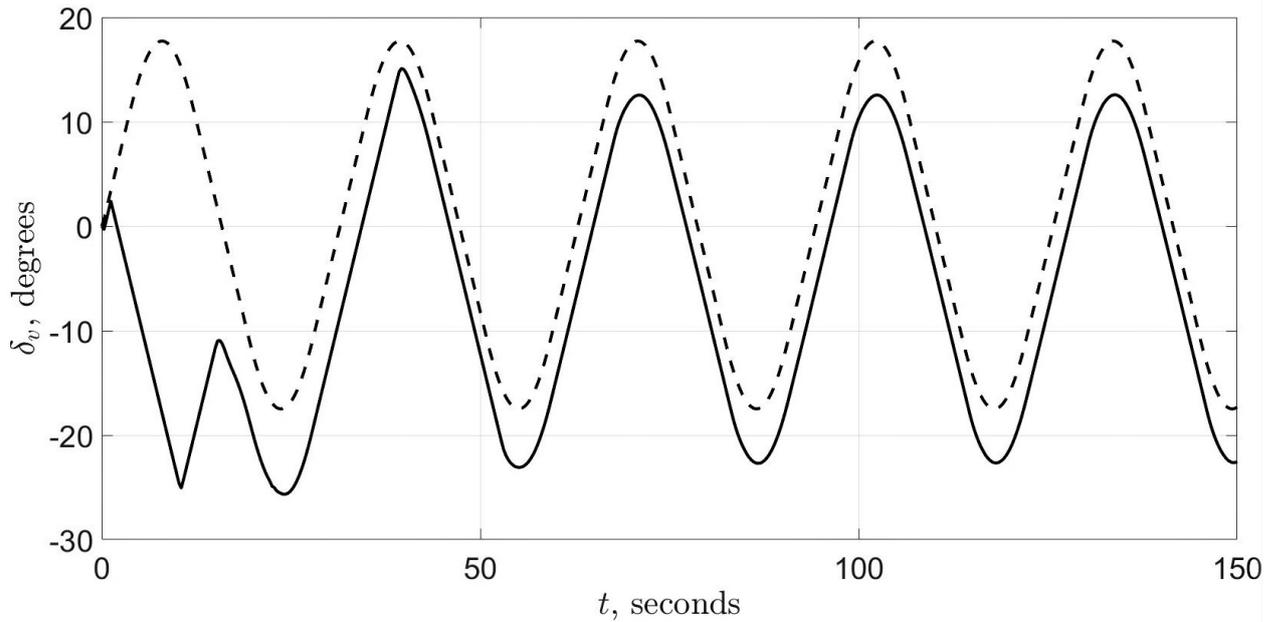


Рис. 19. Переходный процесс по $\delta_v(t)$

Исходя из полученных результатов можно сделать вывод, что многоцелевой закон управления (22) выполняет свою задачу по стабилизации программного движения морского судна в горизонтальной плоскости в случае отклонения начального состояния x_0 от \tilde{x}_0 , а также обеспечивает астатизм угла курса $\varphi(t)$ и угловой скорости $\omega_z(t)$ относительно внешнего возмущения — бокового ветра (как можно заметить на рис. 16, при стабилизации боковой скорости $V_y(t)$ имеет место статическая ошибка).

Однако, в связи с ограничениями (2), накладываемыми на управление $\delta_v(t)$, разработанная система управления не способна компенсировать воздействие бокового ветра при слишком сильной его интенсивности (начиная, примерно, со скорости ветра $V_w = 30$ м/с при угле $A_w = 45^\circ$).

Выводы

Целью работы являлось построение системы автоматического управления, стабилизирующей программное движение морского судна в горизонтальной плоскости при наличии ограничений на управление и отсутствии полной информации о состоянии объекта управления. Морское судно при этом подвержено внешнему возмущающему воздействию аэродинамической природы.

В процессе решения поставленной задачи после анализа соответствующей литературы были сформированы математические модели бокового ветра и движения морского судна. Затем от нелинейной модели динамики морского судна был совершён переход к линейной системе в отклонениях от программного движения, являющейся нестационарной при рассмотренном режиме функционирования объекта.

Для работы с полученной линейной моделью был предложен следующий подход: в каждый момент времени она рассматривалась в качестве стационарной и для неё синтезировался сообщающий требуемые свойства закон управления.

Для возможности использования принципа обратной связи при построении закона управления был синтезирован асимптотический наблюдатель, позволивший получить сходящуюся к реальным значениям оценку вектора состояния объекта управления.

Построение многоцелевого закона управления, решающего поставленную задачу, было произведено в два этапа: сначала с помощью LQR-синтеза был найден базовый закон управления, стабилизирующий программное движение морского судна в случае отклонения фактического начального состояния от программного, а затем была произведена его модификация для компенсации воздействия бокового ветра на требуемые характеристики движения судна по курсу.

Затем в среде MATLAB–Simulink был разработан программный комплекс, позволивший осуществить имитационное компьютерное моделирование процессов управления. Результаты моделирования свидетельствуют об эффективности реализованного подхода.

В то же время данная работа требует дальнейшего серьёзного развития по двум направлениям.

1. Очевидно, что в реальных условиях судно подвержено не только воздействиям аэродинамической природы. Морское волнение также порождает силы, оказывающие существенное влияние на движение морского подвижного объекта. В связи с этим дальнейшая модификация системы управления для компенсации возмущений такого рода является первоочередной задачей. Кроме того канал, по которому передаётся доступная информация о состоянии объекта управления, зачастую оказывается зашумлён, поэтому необходимо построение некоторого фильтра для его очистки от шума.
2. Несмотря на продемонстрированную эффективность предложенного подхода, без теоретического обоснования нет, строго говоря, никакой гарантии его универсальности. Поэтому в дальнейшем планируется, с использованием результатов работы [7], разработать теоретическое обоснование описанного метода.

Заключение

В ходе выполнения данной работы получены следующие основные результаты, выносимые на защиту:

1. на основе анализа соответствующей литературы сформированы математические модели движения морского судна и воздействия бокового ветра;
2. предложен и реализован подход, позволяющий решить задачу стабилизации программного движения морского судна в горизонтальной плоскости при наличии детерминированного внешнего возмущения, ограничений на управление и неполной информации о состоянии объекта управления;
3. разработан программный комплекс, с помощью которого реализовано имитационное компьютерное моделирование процессов управления.

Список литературы

- [1] Зубов В. И. Лекции по теории управления. М.: Наука, 1975. 494 с.
- [2] Дмитриев С. П., Пелевин А. Е. Задачи навигации и управления при стабилизации судна на траектории. СПб.: ГНЦ РФ-ЦНИИ «Электроприбор», 2002. 160 с.
- [3] Компьютерное моделирование систем управления движением морских подвижных объектов / Веремей Е. И., Корчанов В. М., Коровкин М. В., Погожев С. В. СПб.: НИИ Химии СПбГУ, 2002. 370 с.
- [4] Лукомский Ю. А., Корчанов В. М. Управление морскими подвижными объектами: Учебник. СПб.: Элмор, 1996. 320 с.
- [5] Фихтенгольц Г. М. Основы математического анализа, том 1. 6-е изд. М.: Наука, 1968. 440 с.
- [6] Никифоров В. О., Слита О. В., Ушаков А. В. Интеллектуальное управление в условиях неопределённости: учебное пособие. СПб.: СПбГУ ИТМО, 2011. 226 с.
- [7] Зубов Н. И. Математические методы анализа и синтеза линейных нестационарных систем управления. Дис. ... канд. физ.-мат. наук. Санкт-Петербургский гос. университет, Санкт-Петербург, 2003.
- [8] Slotine J.-J. E., Li W. Applied nonlinear control. New Jersey: Prentice-Hall, 1991. 476 p.
- [9] Веремей Е. И. Линейные системы с обратной связью. СПб.: Лань, 2013. 448 с.
- [10] Основы автоматического регулирования и управления / Каргу Л. И., Литвинов А. П., Майборода Л. А., Юсупов Р. М. и др.; под ред. В. М. Пономарёва, А. П. Литвинова. М.: Высшая школа, 1974. 439 с.
- [11] Веремей Е. И., Сотникова М. В. Многоцелевая структура законов управления морскими подвижными объектами // XII Всероссийское совещание по проблемам управления. 2014. С. 3289–3300.

- [12] Сотникова М. В. Алгоритм автоматического удержания колёсного робота на визуально заданной линии // Вестник Санкт-Петербургского университета. Серия 10: Прикладная математика. Информатика. Процессы управления. 2016. № 1. С. 101–110.
- [13] Диткин В. А., Кузнецов П. И. Справочник по операционному исчислению: Основы теории и таблицы формул. М.: Государственное издательство технико-теоретической литературы, 1951. 256 с.
- [14] Гантмахер Ф. Р. Теория матриц. 2-е изд., доп. М.: Наука, 1966. 576 с.
- [15] MATLAB function «diff». <https://www.mathworks.com/help/symbolic/diff.html>
- [16] MATLAB function «lqr». <https://www.mathworks.com/help/control/ref/lqr.html>
- [17] Ногин В. Д. Теория устойчивости движения. СПб.: СПбГУ, ф-т ПМ–ПУ, 2008. 153 с.
- [18] Fossen T. I. Handbook of Marine Craft Hydrodynamics and Motion Control. John Wiley & Sons, Ltd., 2011. 596 p.
- [19] Fossen T. I., Strand J. P. Passive Nonlinear Observer Design for Ships Using Lyapunov Methods: Experimental Results with a Supply Vessel // Automatica. 1999. Vol. 35, No. 1. P. 3–16.
- [20] Fredriksen E., Pettersen K. Y. Global K-exponential way-point maneuvering of ships: Theory and Experiments // Automatica. 2006. Vol. 42, No. 4. P. 677–687.
- [21] Loria A., Fossen T. I., Panteley E. A Separation Principle for Dynamic Positioning of Ships: Theoretical and Experimental Results // IEEE Transactions of Control Systems Technology. 2000. Vol. 8, No. 2. P. 332–343.
- [22] Perez T. Ship Motion Control: Course Keeping and Roll Stabilisation using Rudder and Fins. London: Springer-Verlag, 2005. 300 p.

Приложения

Приложение 1. Инициализация параметров Simulink-модели.

```
% программный угол перекладки вертикального руля:
% nominal delta_v = Amp*sin(freq*time).
global Amp freq;
Amp = 20; % амплитуда
freq = 0.2; % частота
% выбор желаемого регулятора:
global regulator;
% 1 – LQR (R, Q); % линейно-квадратичный регулятор
% 2 – pole placement (desiredPoles); % модальный синтез
regulator = 1;
global V_w A_w; % характеристики бокового ветра
V_w = 20; % величина скорости ветра
A_w = 45; % угол ветра по отношению к инерциальной СК
% не порождаемые боковым ветром проекции
% внешней возмущающей силы и её момента:
global F_x M_x;
F_x = 0;
M_x = 0;
% начальные условия для фактического движения:
global initialConditions;
V_p = 15.2;
V_y0 = 0.5;
omega_x0 = 0.5;
omega_z0 = 0.5;
ksi_0 = 0.5;
dzeta_0 = 0.5;
theta_0 = 0.5;
phi_0 = 0.5;
initialConditions = [V_p, V_y0, omega_x0, omega_z0, ksi_0, dzeta_0,...
    theta_0, phi_0];
```

```

% начальные условия для программного движения:
global nominalInitialConditions;
V_pN = 15.2;
V_y0N = 0;
omega_x0N = 0;
omega_z0N = 0;
ksi_0N = 0;
dzeta_0N = 0;
theta_0N = 0;
phi_0N = 0;
nominalInitialConditions = [V_pN, V_y0N, omega_x0N, omega_z0N,...
    ksi_0N, dzeta_0N, theta_0N, phi_0N];
% желаемое расположение полюсов при модальном синтезе:
global desiredPoles;
% desiredPoles = [-1 + 0.8i; -1 - 0.8i; -0.2; -0.1];
desiredPoles = [-1; -1; -0.2; -0.1];
% параметры линейно-квадратичного функционала в LQR-синтезе:
global R Q;
R = 0.01;
Q = [0.0001, 0, 0, 0, 0, 100, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0];
% C – матрица наблюдения – отражает возможность измерения
% компонент вектора состояния:
global C;
% output = [C*[V_y; w_z; phi]; delta_v];
C = [0, 0, 1];

```

Приложение 2. Реализация модели динамики морского судна.

```

function output = shipNonlinearDynamics(input)
% физические параметры:
m = 402.68; J_zz = 266270; L = 115; L_R = 52.75;
k_11 = 0.02; k_22 = 0.616; k_66 = 0.527; k_44 = 0.503;
mu = 0.34; h_0 = 1.22; z_k = 3.3; g = 9.81;
J_xx = 6573.5; k_24 = 0.31;

```

```

dynamics = zeros(8, 1);
% компоненты вектора состояния:
V_x = input(1); % проекция вектора линейной скорости на ось Oх
V_y = input(2); % проекция вектора линейной скорости на ось Oу
omega_x = input(3); % проекция вектора угловой скорости на ось Oх
omega_z = input(4); % проекция вектора угловой скорости на ось Oz
% ksi = input(5); % продольное смещение центра масс
% dzeta = input(6); % боковое смещение центра масс
teta = input(7); % угол крена
phi = input(8); % угол курса
delta_v = input(9); % угол перекладки вертикального руля
% скорость хода в установившемся режиме:
V_p = input(10);
% проекции силы и момента внешнего возмущающего воздействия
% на оси связанной системы координат:
F_x = input(11);
F_y = input(12);
M_x = input(13);
M_z = input(14);
V = sqrt(V_x^2 + V_y^2); % величина скорости морского судна
V_LR = sqrt(V^2 + (L_R^2)*(omega_z^2));
omega = omega_z*L/V_LR;
V_L = sqrt(V^2 + (L^2)*(omega_z^2));
Omega = omega_z*L/V_L; % угловая скорость рыскания
beta = -atan(V_y/V_x); % угол дрейфа
nu = V_p/3.039; % число оборотов гребного винта
T_v = 9.74*(nu^2) - 2.23*V*nu; % тяга гребного винта
alpha = delta_v - (beta + L_R*omega_z/V);
% проекции гидродинамических сил и момента вязкостной природы,
% действующих на корпус судна:
X_H = 0.5775*(1 - Omega^2)*V_L^2;
Y_H = 4.89*(V_L^2)*beta - 4.39*(V_L^2)*Omega*sqrt(1 - Omega^2) +...
23.7*(V_L^2)*beta*abs(beta);

```

```

M_xH = -11.39*(V_L^2)*beta + 10.23*(V_L^2)*Omega*sqrt(1 -...
    Omega^2) - 55.23*(V_L^2)*beta*abs(beta);
M_zH = 260.0*(V^2)*beta - 223.6*(V_L^2)*Omega +...
    69.1*(V_L^2)*Omega*abs(beta) - 160.8*(V_L^2)*Omega*abs(Omega);
% проекции сил и моментов, обусловленных перекладкой
% вертикального руля:
X_R = 0.462*(alpha^2)*(V_LR^2);
Y_R = 1.236*(V_LR^2)*beta + 0.567*(V_LR^2)*omega -...
    1.236*(V_LR^2)*delta_v;
M_xR = -3.91*(V_LR^2)*beta - 1.79*(V_LR^2)*omega +...
    3.91*(V_LR^2)*delta_v;
M_zR = -65.2*(V_LR^2)*beta - 29.9*(V_LR^2)*omega +...
    65.2*(V_LR^2)*delta_v;
% проекции совокупности сил и моментов, действующих на корпус судна:
Q_x = 1.8*T_v - X_H - X_R + m*(1 + k_22)*V_y*omega_z + F_x;
Q_y = Y_H + Y_R + F_y;
N_x = -mu*J_xx*omega_x - m*g*h_0*teta - z_k*m*V*omega_z +...
    M_xH + M_xR + M_x;
N_z = M_zH + M_zR + M_z;
% динамика морского судна:
dynamics(1) = Q_x/(m*(1 + k_11));
dynamics(2) = (Q_y*J_xx*(1 + k_44) - N_x*m*k_24)/(m*(1 +...
    k_22)*J_xx*(1 + k_44) - (m*k_24)^2);
dynamics(3) = (N_x*m*(1 + k_22) - Q_y*m*k_24)/(m*(1 +...
    k_22)*J_xx*(1 + k_44) - (m*k_24)^2);
dynamics(4) = N_z/(J_zz*(1 + k_66));
dynamics(5) = V_x*cos(phi) + V_y*cos(teta)*sin(phi);
dynamics(6) = -V_x*sin(phi) + V_y*cos(teta)*cos(phi);
dynamics(7) = omega_x;
dynamics(8) = omega_z;
% выходные данные:
output = [dynamics; beta];
end

```

Приложение 3. Реализация модели бокового ветра.

```
function output = sideWindForceAndMomentum(input)
V_x = input(1); % проекция вектора линейной скорости на ось Oх
V_y = input(2); % проекция вектора линейной скорости на ось Oу
phi = input(3); % угол курса
A_w = input(4); % угол между ортом оси Oх_0 и вектором  $-V_w$ 
V_w = input(5); % величина скорости ветра
phi = rem(phi, 2*pi);
A_c = A_w - phi;
V = sqrt(V_x^2 + V_y^2); % величина скорости морского судна
beta = -asin(V_y/V); % угол дрейфа
Omega = V_w^2 + V^2 + 2*V*V_w*cos(A_c + beta);
% - квадрат величины скорости потока воздуха относительно корпуса судна
Gamma = atan((V*sin(beta) - V_w*sin(A_c))/(V_w*cos(A_c) + ...
    V*cos(beta))); % угол встречи с набегающим потоком
% проекции векторов силы, порождаемой боковым ветром, и её момента
% на оси связанной СК:
F_y = 0.076*sin(Gamma)*Omega;
M_z = -8.72*sin(Gamma)*(0.273 - Gamma/(2*pi))*Omega;
output = [F_y; M_z];
end
```

Приложение 4. Реализация многоцелевого закона управления.

```
function u = LQR(input)
% программный вектор состояния:
x_nom = [input(1); input(2); input(3); input(4)];
A = zeros(4, 4);
A(1, 1) = input(5);
A(1, 2) = input(6);
A(1, 4) = input(7);
A(2, 1) = input(8);
A(2, 2) = input(9);
```

```

A(2, 4) = input(10);
A(3, 2) = 1;
a11 = A(1, 1);
a12 = A(1, 2);
a14 = A(1, 4);
a21 = A(2, 1);
a22 = A(2, 2);
a24 = A(2, 4);
y_p = input(11); % измеряемая информация о состоянии морского судна
B = [0; 0; 0; 1];
% оценка вектора отклонения x_hat:
z_hat = [input(13); input(14); input(15); input(16)];
% матрица при невязке в динамике наблюдателя:
G = [input(17); input(18); input(19)];
g1 = G(1);
g2 = G(2);
g3 = G(3);
% параметры LQR-синтеза:
R = input(20);
Q = [input(21), input(22), input(23), input(24);
     input(25), input(26), input(27), input(28);
     input(29), input(30), input(31), input(32);
     input(33), input(34), input(35), input(36)];
C = [input(37), input(38), input(39)]; % матрица наблюдения
c1 = C(1);
c2 = C(2);
c3 = C(3);
K = lqr(A, B, Q, R); % LQR-синтез
k1 = K(1);
k2 = K(2);
k3 = K(3);
k4 = K(4);
% множитель при невязке в многоцелевом законе управления:

```

```

K_Delta = (a11*a24*k3 - a14*a21*k3 + a11*c3*g2*k4 + a14*c1*g2*k3 - ...
a14*c3*g2*k1 - a21*c3*g1*k4 - a24*c1*g1*k3 + a24*c3*g1*k1 - ...
a11*a22*c3*g3*k4 - a11*a24*c2*g3*k3 + a11*a24*c3*g3*k2 + ...
a12*a21*c3*g3*k4 + a12*a24*c1*g3*k3 - a12*a24*c3*g3*k1 + ...
a14*a21*c2*g3*k3 - a14*a21*c3*g3*k2 - a14*a22*c1*g3*k3 + ...
a14*a22*c3*g3*k1)/(-a11*a24*c3 + a14*a21*c3);
delta_v = input(40); % прибавляется к стабилизирующему управлению
% для компенсации инерционности привода
% стабилизирующее управление:
u = -K*z_hat + delta_v + K_Delta*(y_p - [C, 0]*x_nom - [C, 0]*z_hat);
end

```

Приложение 5. Реализация уравнений динамики асимптотического наблюдателя.

```

function output = observer(input)
% программный вектор состояния:
x_nom = [input(1); input(2); input(3); input(4)];
A_p = zeros(3, 3);
A_p(1, 1) = input(5);
A_p(1, 2) = input(6);
A_p(2, 1) = input(8);
A_p(2, 2) = input(9);
A_p(3, 2) = 1;
a11 = A_p(1, 1);
a12 = A_p(1, 2);
a21 = A_p(2, 1);
a22 = A_p(2, 2);
B_p = zeros(3, 1);
B_p(1, 1) = input(7);
B_p(2, 1) = input(10);
% измеряемая информация о состоянии морского судна:
y = [input(11); input(12)];
% матрица наблюдения:

```

```

C = [input(13), input(14), input(15)];
c1 = C(1);
c2 = C(2);
c3 = C(3);
% желаемое распределение полюсов:
observerDesiredPoles = [input(16); input(17); input(18)];
p1 = observerDesiredPoles(1);
p2 = observerDesiredPoles(2);
p3 = observerDesiredPoles(3);
z_p = [input(19); input(20); input(21)]; % часть оценки вектора отклонения
% найденная с помощью модального синтеза
% матрица при невязке в динамике наблюдателя:
G = [-(a12^2*a21^2*c2 - a11^3*c3 - a11^3*a22*c2 + a11^2*c3*p1 +...
      a11^2*c3*p2 + a11^2*c3*p3 - a11*c3*p1*p2 - a11*c3*p1*p3 -...
      a11*c3*p2*p3 + c3*p1*p2*p3 -...
      a11*a12^2*a21*c1 + a11*a12*a22^2*c1 +...
      a11^2*a12*a21*c2 + a11^2*a12*a22*c1 - a12^2*a21*a22*c1 +...
      a12^2*a21*c1*p1 + a11^2*a22*c2*p1 + a12^2*a21*c1*p2 +...
      a11^2*a22*c2*p2 + a12^2*a21*c1*p3 + a11^2*a22*c2*p3 -...
      2*a11*a12*a21*c3 - a12*a21*a22*c3 +...
      a12*a21*c3*p1 + a12*a21*c3*p2 +...
      a12*a21*c3*p3 - a11*a12*a21*a22*c2 - a11*a12*a21*c2*p1 -...
      a11*a12*a22*c1*p1 - a11*a12*a21*c2*p2 -...
      a11*a12*a22*c1*p2 - a11*a12*a21*c2*p3 -...
      a11*a12*a22*c1*p3 - a11*a22*c2*p1*p2 +...
      a12*a21*c2*p1*p2 - a11*a22*c2*p1*p3 +...
      a12*a21*c2*p1*p3 - a11*a22*c2*p2*p3 +...
      a12*a21*c2*p2*p3 + a12*c1*p1*p2*p3 +...
      a22*c2*p1*p2*p3)/(a11^2*a22*c1*c2 +...
      a11^2*c1*c3 - a11*a12*a21*c1*c2 -...
      a11*a12*a22*c1^2 + a11*a21*a22*c2^2 +...
      a11*a21*c2*c3 - a11*a22^2*c1*c2 -...
      a11*a22*c1*c3 + a12^2*a21*c1^2 -...

```

$$\begin{aligned}
& a_{12}^2 a_{21}^2 c^2 + a_{12} a_{21} a_{22} c_1 c_2 + \dots \\
& 2 a_{12} a_{21} c_1 c_3 + a_{21} a_{22} c_2 c_3 + a_{21} c_3^2); \\
& (a_{12}^2 a_{21}^2 c_1 - a_{11} a_{22}^3 c_1 + a_{11}^2 a_{21} c_3 + a_{12} a_{21}^2 c_3 + \dots \\
& a_{21} a_{22}^2 c_3 + a_{21} c_3^2 p_1 p_2 + a_{21} c_3^2 p_1 p_3 + a_{21} c_3^2 p_2 p_3 - \dots \\
& a_{11} a_{12} a_{21}^2 c_2 + a_{11} a_{21} a_{22}^2 c_2 + \dots \\
& a_{12} a_{21} a_{22}^2 c_1 + a_{11}^2 a_{21} a_{22} c_2 - \dots \\
& a_{12} a_{21}^2 a_{22} c_2 + a_{11} a_{22}^2 c_1 p_1 + \dots \\
& a_{11} a_{22}^2 c_1 p_2 + a_{12} a_{21}^2 c_2 p_1 + \dots \\
& a_{11} a_{22}^2 c_1 p_3 + a_{12} a_{21}^2 c_2 p_2 + \dots \\
& a_{12} a_{21}^2 c_2 p_3 + a_{11} a_{21} a_{22} c_3 - \dots \\
& a_{11} a_{21} c_3^2 p_1 - a_{11} a_{21} c_3^2 p_2 - a_{11} a_{21} c_3^2 p_3 - a_{21} a_{22} c_3^2 p_1 - \dots \\
& a_{21} a_{22} c_3^2 p_2 - a_{21} a_{22} c_3^2 p_3 - \dots \\
& a_{11} a_{12} a_{21} a_{22} c_1 - a_{11} a_{21} a_{22} c_2 p_1 - \dots \\
& a_{12} a_{21} a_{22} c_1 p_1 - a_{11} a_{21} a_{22} c_2 p_2 - a_{12} a_{21} a_{22} c_1 p_2 - \dots \\
& a_{11} a_{21} a_{22} c_2 p_3 - a_{12} a_{21} a_{22} c_1 p_3 - a_{11} a_{22} c_1 p_1 p_2 + \dots \\
& a_{12} a_{21} c_1 p_1 p_2 - a_{11} a_{22} c_1 p_1 p_3 + \dots \\
& a_{12} a_{21} c_1 p_1 p_3 - a_{11} a_{22} c_1 p_2 p_3 + \dots \\
& a_{12} a_{21} c_1 p_2 p_3 + a_{11} c_1 p_1 p_2 p_3 + \dots \\
& a_{21} c_2 p_1 p_2 p_3) / (a_{11}^2 a_{22} c_1 c_2 + \dots \\
& a_{11}^2 c_1 c_3 - a_{11} a_{12} a_{21} c_1 c_2 - \dots \\
& a_{11} a_{12} a_{22} c_1^2 + a_{11} a_{21} a_{22} c_2^2 + \dots \\
& a_{11} a_{21} c_2 c_3 - a_{11} a_{22}^2 c_1 c_2 - \dots \\
& a_{11} a_{22} c_1 c_3 + a_{12}^2 a_{21} c_1^2 - \dots \\
& a_{12} a_{21}^2 c_2^2 + a_{12} a_{21} a_{22} c_1 c_2 + 2 a_{12} a_{21} c_1 c_3 + \dots \\
& a_{21} a_{22} c_2 c_3 + a_{21} c_3^2); \\
& -(a_{21} c_3^2 p_1 - a_{21} a_{22} c_3^2 - a_{11} a_{21} c_3^2 + a_{21} c_3^2 p_2 + \dots \\
& a_{21} c_3^2 p_3 + a_{11} a_{22}^2 c_1 c_3 + \dots \\
& a_{12} a_{21}^2 c_2 c_3 - a_{11} a_{21} a_{22} c_2 c_3 - \dots \\
& a_{12} a_{21} a_{22} c_1 c_3 - a_{11} a_{22} c_1 c_3 p_1 + a_{12} a_{21} c_1 c_3 p_1 - \dots \\
& a_{11} a_{22} c_1 c_3 p_2 + a_{12} a_{21} c_1 c_3 p_2 - \dots \\
& a_{11} a_{22} c_1 c_3 p_3 + a_{12} a_{21} c_1 c_3 p_3 + \dots \\
& a_{11} c_1 c_3 p_1 p_2 + a_{11} c_1 c_3 p_1 p_3 + \dots \\
& a_{11} c_1 c_3 p_2 p_3 + a_{21} c_2 c_3 p_1 p_2 + \dots
\end{aligned}$$

```

a21*c2*c3*p1*p3 + a21*c2*c3*p2*p3 -...
c1*c3*p1*p2*p3 - a12*c1^2*p1*p2*p3 +...
a21*c2^2*p1*p2*p3 + a11*c1*c2*p1*p2*p3 -...
a22*c1*c2*p1*p2*p3)/(a11^2*a22*c1*c2*c3 + a11^2*c1*c3^2 -...
a11*a12*a21*c1*c2*c3 - a11*a12*a22*c1^2*c3 +...
a11*a21*a22*c2^2*c3 +...
a11*a21*c2*c3^2 - a11*a22^2*c1*c2*c3 -...
a11*a22*c1*c3^2 + a12^2*a21*c1^2*c3 -...
a12*a21^2*c2^2*c3 + a12*a21*a22*c1*c2*c3 + 2*a12*a21*c1*c3^2 +...
a21*a22*c2*c3^2 + a21*c3^3)];
% G = (place(A_p', C', observerDesiredPoles));
% уравнения динамики асимптотического наблюдателя:
dynamicsEstimation = A_p*z_p +...
    B_p*(y(2) - x_nom(4)) +...
    G*(y(1) - [C, 0]*x_nom - C*z_p);
output = [dynamicsEstimation; y(2) - x_nom(4); G];
end

```