**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра КСУ**

Индивидуальное домашнее задание

**по дисциплине «ММОиСУ»**

Тема: Математическое моделирование систем управления

Вариант 24-10-29

|  |  |  |
| --- | --- | --- |
| Студент гр. 9493 |  | Викторов А.Д. |
| Преподаватель |  | Шпекторов А.Г. |

Санкт-Петербург

2023

**ЗАДАНИЕ**

1. Разработать математическую модель движения корабля по индивидуальным исходным данным (п. 5).

2. Составить и отладить непрерывную и разностную программную модель движения корабля на языке высокого уровня MATLAB.

3. Разработать интерфейс программы моделирования кораблей разных типов с использованием средств MATLAB GUI, позволяющий осуществлять в рабочем окне ввод исходных данных и вывод результатов.

4. Исследовать динамические свойства моделируемого объекта на ПК, используя примеры или по указанию преподавателя. Результаты занести в таблицу.

5. По данным таблицы построить графики функций *P = f*(*t*); *x = f*(*t*); *V = f*(*t*).

6. По построенным графикам определить основные параметры движения корабля, указанные в соответствующих примерах.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Исходные данные представлены в таблице 1.  *Таблица 1 – Исходные данные*   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | № | Наименование | W, т | N, л.с. | v, уз | vk, уз | v1max, уз | v2max, уз | | 24 | «Активный» | 314 | 480 | 8.5 | - | - | - | | 10 | «Аист» | 4.6 | 235 | 19.5 | - | - | - | | 29 | КПК | 50 | 3500 | - | 16 | 19.2 | 46 | |

# **Формирование математических моделей объектов управления**

# **Математическая модель движения надводного водоизмещающего корабля**

В наиболее общем виде поступательное движение любого объекта, в данном случае надводного водоизмещающего корабля, можно описать с помощью второго закона Ньютона:



где, M, a – масса корабля и его ускорение.

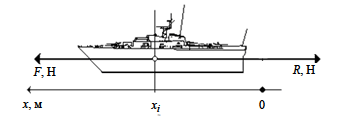
Правая часть представляет собой алгебраическую (с учетом направления, знака) сумму сил, действующих на корабль (Рис. 1.1.). Для НВК будем рассматривать следующие силы: F – движущая сила, или сила тяги винта (движителя), Н; R – сила сопротивления движению, Н.

Уравнение (1.1) называют также балансным уравнением движения корабля. При балансе или равенстве сил F и R ускорение корабля рано нулю, корабль движется равномерно (с постоянной скоростью) и прямолинейно. Если F> R – корабль разгоняется и наоборот.

Сила сопротивления направлена в сторону, противоположную силе тяги, и зависит от многих параметров движения, геометрии корпуса НВК и внешних возмущений. Однако в первом приближении ее можно считать пропорциональной квадрату скорости корабля: , или для возможности учета направления



где А – коэффициент пропорциональности.



1. Схема сил, действующих на надводный водоизмещающий корабль при прямолинейном движении

Такой подход является приближенным и годится лишь для построения упрощенных моделей. С учетом формулы (1.2) балансное уравнение движения (1.1) теперь может быть записано следующим образом:



Входящая в уравнение (1.3) скорость в некоторый момент времени ti (мгновенная скорость) в направлении координаты пройденного расстояния x (рис. 1) в общем случае определяется как 

Коэффициент пропорциональности силы сопротивления движению А может быть определен по граничным условиям, известным для каждого моделируемого объекта, максимальному значению скорости  при максимальной силе тяги . Поскольку скорость  ограничена силой сопротивления среды, т.е. , то можно записать , откуда:



Ограничения разработанной модели:

1. Предполагается, что корабль движется на постоянном курсе.
2. Не учитывается волнение моря, гидродинамические особенности корпуса, переменное воздействие ветра и т.п.

Сила тяги, определяемая оборотами винтов судна, не может изменяться мгновенно вследствии инерционности вращающих винг двигателей. Полагая зависимость силы тяги от времени линейной, ее изменение можно также описать при помощи дифференциального уравнения с нелинейностью типа «ограничение» («насыщение»):



где kF – коэффициент скорости изменения тяги, Н/с; F\*, F\*≤Fmax – заданное значение силы тяги. Коэффициент kF – постоянный по модулю, но может иметь разные знаки (для моделирования разгона и торможения НВК).

При известном значении силы тяги можно решить дифференциальное уравнение (1.5) и задавать в модели силу тяги как функцию времени.

Для придания модели универсальности целесообразно, по возможности, использовать не абсолютные величины, а их относительные значения. В данном случае предпочтительно принять относительные значения силы тяги в процентах от максимальной:



где Р\* - относительное значение силы тяги в процентах от максимальной.

Соответственно, имеем модель НВК:



Полученная математическая модель (1.7) позволяет для любого момента времени определить расстояние, пройденное кораблем (координата x(t)), и скорость движения v(t) по задаваемым значениям силы тяги P\*. Переход к относительным величинам позволяет вводить в качестве исходных данных значения силы тяги (мощности двигателя) в процентах от максимальной, а не в ньютонах, как и принято в современной практике судовождения.

Математическая модель (1.7) может быть использована для изучения динамических характеристик корабля при проектировании, эксплуатации, создания тренажеров и т.п.

# **Математическая модель движения корабля на подводных крыльях**

Скорость движения кораблей на подводных крыльях в два-три раза выше скорости водоизмещающих кораблей и может достигать 65 узлов (≈120 км/ч). Это предельная скорость для КПК из-за кавитации, возникающих на крыльях и лопастях винта.

Особенностью движения корабля на подводных крыльях, в отличии от водоизмещающего надводного корабля, является наличие трех режимов движения:

* Движение на корпусе (водоизмещающий режим). Осуществляется до определенной скорости vk, после чего переходит в режим глиссирования;
* Глиссирование – режим движения, когда корабль, набрав скорость vk, начинает выходить из воды на крылья, достигая при этом скорости v1max, после которой переходит в следующий режим;
* Движение на крыльях (крыльевой режим); при достижении скорости v1max на крыльях возникает и поддерживается соответствующая подъемная сила, сопротивление среды резко уменьшается, а максимально возможная скорость v2max увеличивается.

Указанная особенность режима движения на крыльях должна быть учтена при разработке математической модели. При моделировании движения водоизмещающего корабля было показано, что сила сопротивления движению пропорциональна квадрату скорости (1.2). Для корабля на подводных крыльях существует два коэффициента пропорциональности:

A1 – при  (водоизмещающий режим);

A2 – при  (режим глиссирования и движения на крыльях), что обеспечивает возможность достижения максимальных скоростей v1max и v2max.

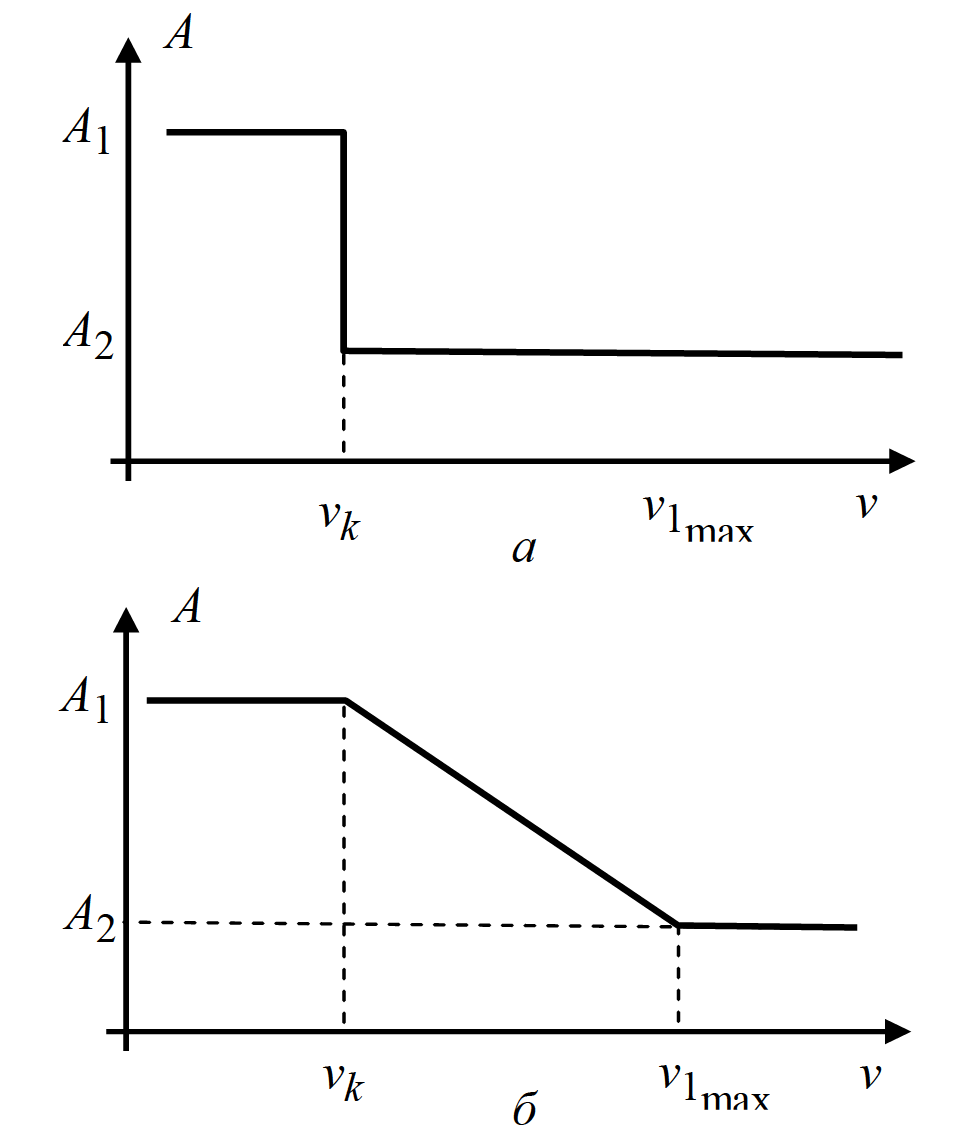
Таким образом, математическая модель КПК имеет вид



# **Уточнение математических моделей**

Одним из методов построения математической и реализующей ее на компьютере программной моделей является метод «быстрого прототипа». Основное содержание его заключается в том, что вначале строится максимально упрощенная математическая модель с целью получить первые результаты исследования и оценить их правдоподобность в максимально короткие сроки, т.е. проверяется перспективность выбранного направления исследований. В дальнейшем ММ постепенно усложняется и уточняется с целью все большего приближения выходных данных к реальным.

В математической модели (1.8), например, показано, что коэффициент силы сопротивления движению А принимает значения A2 при v≥vk, т.е. подчиняется релейному закону (рис. 2, а). В первом приближении это соответствует действительности. Однако в реальных условиях сила не может измениться скачком, вследствие чего корабль выходит в крыльевой режим не мгновенно, а в течении некоторого времени. Естественно, что сила сопротивления, выражаемая коэффициентом А, уменьшается постепенно, по мере выхода корпуса из воды на крылья. При скорости v1max значение коэффициента А действительно становится равным А2.



1. Законы изменения коэффициента А: a – релейный; б – линейный

Очевидно, что принятый релейный закон смены коэффициентов недостаточен. Для устранения недостатка можно принять, например линейный закон изменения коэффициента А в зависимости от скорости корабля (рис.2, б). Это приблизит модель к реальному объекту. Вероятно, этот закон не является строго линейным и для дальнейшего уточнения модели требуются дополнительные аналитические или натурные исследования. Процесс уточнения математической модель носит последовательный итерационный характер.

Для принятого линейного закона коэффициент А будет определяться из следующей системы уравнений:



# **ПРОГРАММНАЯ МОДЕЛЬ ДВИЖЕНИЯ КОРАБЛЯ**

Альтернативным способом численного моделирования нелинейных динамических систем является переход к разностным уравнениям. При этом программные модели могут быть сформированы без применения специализированных функций на любом высокоуровневом языке программирования. Уравнения решают последовательно для дискретных моделей времени ti, ti+1, …. При этом , где  - шаг приращения времени (интегрирования). Таким образом, при выборе достаточно малого шага ∆t мгновенная скорость может быть определена по конечной разности расстояний:



А ускорение в момент времени - по формуле



Подставив формулы (2.1) и (2.2) в уравнение движения (1.3), получим:



откуда легко получить расчетную рекуррентную формулу для вычисления очередного значения координаты пройденного расстояния:



Из уравнений (2.3) и (2.4) видно, что для вычисления скорости движения НВК необходимо иметь значения двух координат (xi-1 и xi), в то время как для определения ускорения – три, дополнительно xi+1.

В окончательном виде математическая модель движения надводного водоизмещающего корабля (1.7) может быть представлена системой уравнений:



а математическая модель движения КПК (1.8)



где Pi – относительное значение силы тяги в процентах от максимальной.

Математическая модель (2.6) также может быть уточнена в соответствии с (1.9).

Достоинством использования функции ode45 является встроенный выбор шага интегрирования. В расширенном формате записи функции ode45 можно задавать другие параметры моделирования, в том числе и диапазон изменения шага интегрирования.

# **Реализация разностной модели движения надводного водоизмещающего судна «Активный»**

*Таблица 2 – Исходные данные НВК «Активный»*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Наименование | W, т | N, л.с. | v, уз |
| 24 | «Активный» | 314 | 480 | 8.5 |

Судно «Активный» имеет водоизмещение 314 т, примем максимально допустимую скорость изменения силы тяги , а шаг интегрирования примем  Допустимая погрешность при моделировании принимается равной 0,005.

Пример расчета:

1. Масса судна: 
2. Максимальная мощность двигателя: 
3. Максимальная скорость судна: 
4. Максимальная сила тяги движителя: 
5. Максимально допустимое изменение силы тяги: 

Разработка разностной модели движения:

1. Шаг приращения: 
2. Относительное (в процентах от максимальной) допустимое изменение силы тяги за время
3. Коэффициент пропорциональности силы сопротивления движению:;
4. Координата пройденного расстояния: 
5. Текущая скорость: .

Код программы, реализующий разностную модель движения надводного водоизмещающего корабля представлен в листинге 1.

*Листинг 1 – Код программы для разностной модели НВК*

clear, clc, close all

W0 = 314;% тонн

N0 = 480;% лс

v0 = 8.5; %уз

W = W0\*1000;%кг

N = N0\*735.5;% Вт

Vmax = v0\*0.51;% м/с

Fmax = N/Vmax;% N = F\*v

A = Fmax/Vmax^2;

dFmax = 0.2\*Fmax;

dt = 1;

dPmax = dt\*dFmax/Fmax\*100;

dP = 20;

if (dP >= dPmax)

dP = dPmax;

end

% Н.У.

Pmax = 100; % максимальная тяга

i = 1;

t = [0];

v = [0];

P = [0];

x = [0];

dx = 0;

*Листинг 1 – Продолжение*

e = 0.005; % допустимая погрешность

% разгон

while ( (Vmax-v(i)) > e)

t = [t t(i)+dt];

x = [x x(i)+dx+(P(i)\*Fmax\*dt^2/100-A\*dx\*abs(dx))/W];

v = [v (x(i+1)-x(i))/dt];

if (P(i) < Pmax && (P(i)+dP) <= 100)

P = [P (P(i)+dP)];

else

P = [P Pmax];

end

i = i + 1;

dx = x(i)-x(i-1);

if ((v(i)-v(i-1))<e/10 && i > 3) %учитывает, что мощность может быть не 100%

break

end

end

% торможение

while ( v(i) > e)

tt = t(i)+dt;

t = [t tt];

tx = x(i)+dx+(P(i)\*Fmax\*dt^2/100-A\*dx\*abs(dx))/W;

x = [x tx];

tv = (x(i+1)-x(i))/dt;

v = [v tv];

if (P(i) > -Pmax)

P = [P (P(i)-dP)];

else

P = [P -Pmax];

end

i = i + 1;

dx = x(i)-x(i-1);

end

NVK\_active = [t' P' x' v']; % составление таблицы

figure(1)

subplot(3,1,1)

plot(t,x,'Linewidth',1.5); grid on;

xlabel('t, c','FontSize',12);

ylabel('x, м','FontSize',12);

subplot(3,1,2)

plot(t,v,'Linewidth',1.5);grid on;

xlabel('t, c','FontSize',12);

ylabel('v, м/с','FontSize',12);

subplot(3,1,3)

plot(t,P,'Linewidth',1.5);grid on;

xlabel('t, c','FontSize',12);

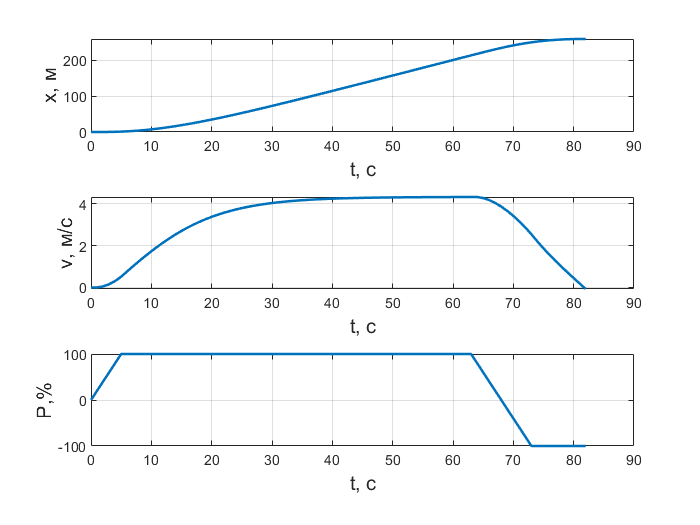
ylabel('P,%','FontSize', 12);

Результаты моделирования разностной модели НВК «Активный» сведены в таблицу 2.

Таблица 2. Результаты моделирования НВК «Активный»

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Режим | T, с | P, % | xi+1, м | vi+1, м/с | Режим | T, с | P, % | xi+1, м | vi+1, м/с |
| Разгон | 0 | 0 | 0,00 | 0,00 | Разгон | 41 | 100 | 118,77 | 4,26 |
| 1 | 20 | 0,00 | 0,00 | 42 | 100 | 123,04 | 4,27 |
| 2 | 40 | 0,05 | 0,05 | 43 | 100 | 127,32 | 4,28 |
| 3 | 60 | 0,21 | 0,16 | 44 | 100 | 131,61 | 4,28 |
| 4 | 80 | 0,52 | 0,31 | 45 | 100 | 135,90 | 4,29 |
| 5 | 100 | 1,04 | 0,52 | 46 | 100 | 140,19 | 4,30 |
| 6 | 100 | 1,81 | 0,77 | 47 | 100 | 144,49 | 4,30 |
| 7 | 100 | 2,83 | 1,02 | 48 | 100 | 148,80 | 4,30 |
| 8 | 100 | 4,10 | 1,27 | 49 | 100 | 153,10 | 4,31 |
| 9 | 100 | 5,61 | 1,51 | 50 | 100 | 157,42 | 4,31 |
| 10 | 100 | 7,34 | 1,73 | 51 | 100 | 161,73 | 4,31 |
| 11 | 100 | 9,29 | 1,95 | 52 | 100 | 166,05 | 4,32 |
| 12 | 100 | 11,45 | 2,16 | 53 | 100 | 170,37 | 4,32 |
| 13 | 100 | 13,80 | 2,35 | 54 | 100 | 174,69 | 4,32 |
| 14 | 100 | 16,34 | 2,54 | 55 | 100 | 179,01 | 4,32 |
| 15 | 100 | 19,05 | 2,71 | 56 | 100 | 183,33 | 4,32 |
| 16 | 100 | 21,91 | 2,87 | 57 | 100 | 187,66 | 4,33 |
| 17 | 100 | 24,92 | 3,01 | 58 | 100 | 191,98 | 4,33 |
| 18 | 100 | 28,07 | 3,15 | 59 | 100 | 196,31 | 4,33 |
| 19 | 100 | 31,34 | 3,27 | 60 | 100 | 200,64 | 4,33 |
| 20 | 100 | 34,72 | 3,38 | 61 | 100 | 204,97 | 4,33 |
| 21 | 100 | 38,20 | 3,48 | 62 | 100 | 209,30 | 4,33 |
| 22 | 100 | 41,77 | 3,57 | 63 | 100 | 213,63 | 4,33 |
| 23 | 100 | 45,43 | 3,66 | Торможение | 64 | 80 | 217,96 | 4,33 |
| 24 | 100 | 49,16 | 3,73 | 65 | 60 | 222,24 | 4,28 |
| 25 | 100 | 52,96 | 3,80 | 66 | 40 | 226,42 | 4,18 |
| 26 | 100 | 56,82 | 3,86 | 67 | 20 | 230,47 | 4,04 |
| 27 | 100 | 60,73 | 3,91 | 68 | 0 | 234,34 | 3,87 |
| 28 | 100 | 64,70 | 3,96 | 69 | -20 | 238,00 | 3,66 |
| 29 | 100 | 68,70 | 4,00 | 70 | -40 | 241,43 | 3,43 |
| 30 | 100 | 72,74 | 4,04 | 71 | -60 | 244,59 | 3,16 |
| 31 | 100 | 76,82 | 4,08 | 72 | -80 | 247,46 | 2,87 |
| 32 | 100 | 80,92 | 4,11 | 73 | -100 | 250,00 | 2,55 |
| 33 | 100 | 85,06 | 4,13 | 74 | -100 | 252,20 | 2,20 |
| 34 | 100 | 89,21 | 4,16 | 75 | -100 | 254,07 | 1,87 |
| 35 | 100 | 93,39 | 4,18 | 76 | -100 | 255,64 | 1,56 |
| 36 | 100 | 97,59 | 4,20 | 77 | -100 | 256,91 | 1,27 |
| 37 | 100 | 101,80 | 4,21 | 78 | -100 | 257,90 | 0,99 |
| 38 | 100 | 106,02 | 4,23 | 79 | -100 | 258,61 | 0,72 |
| 39 | 100 | 110,26 | 4,24 | 80 | -100 | 259,06 | 0,45 |
| 40 | 100 | 114,51 | 4,25 | 81 | -100 | 259,25 | 0,19 |
|  | | | | 82 | -100 | 259,18 | -0,07 |

Графики координаты, скорости и относительной мощности НВК «Активный» представлены на рисунке 3.



1. Графики координаты, скорости и мощности НВК «Активный»

По полученным данным можно сделать следующие выводы о динамических характеристиках НВК «Активный»:

* Время набора максимальной скорости: 4,3 м/с – 63 секунды, при этом судно проходит расстояние 214 м.
* Время торможения: 19 с на расстоянии 45 м;
* Общее время движения судна составило 82 с;
* Общее пройденное расстояние: 259 м.

# **Моделирование движения малого рабочего катера**

Исходные данные малого рабочего катера представлены в таблице 3.

Таблица 3. Исходные данные малого рабочего катера

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Наименование | W, т | N, л.с. | v, уз |
| 10 | «Аист» | 4.6 | 235 | 19.5 |

Малый рабочий катер имеет водоизмещение 4,6 т, следовательно, примем максимально допустимую скорость изменения силы тяги , а шаг интегрирования примем 

Математическая модель малого рабочего катера имеет такой же вид, что и математическая модель НВК «Активный». Основным отличием в реализации программного кода состоит в задании параметров судна.

Фрагмент листинга программы представлен ниже.

*Листинг 2 – Код программы для разностной модели НВК «Аист»*

clear, clc, close all

W0 = 4.6;% тонн

N0 = 235;% лс

v0 = 19.5; %уз

W = W0\*1000;%кг

N = N0\*735.5;% Вт

Vmax = v0\*0.51;% м/с

Fmax = N/Vmax;% N = F\*v

A = Fmax/Vmax^2;

dFmax = 0.2\*Fmax;

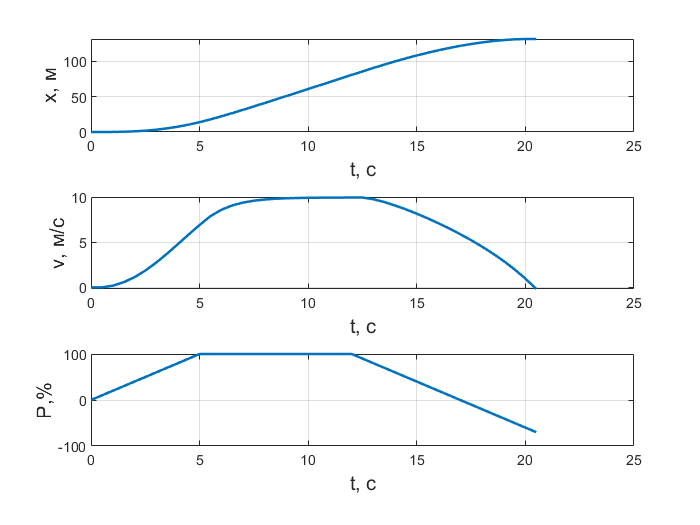
dt = 0.5;

Остальная часть кода идентична коду разностной модели НВК «Активный». Результаты моделирования разностной модели малого рабочего катера сведены в таблицу 4.

Таблица 4. Результаты моделирования малого рабочего катера

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Режим | T, с | P, % | xi+1, м | vi+1, м/с | Режим | T, с | P, % | xi+1, м | vi+1, м/с |
| Разгон | 0,00 | 0,00 | 0,00 | 0,00 | Торможение | 12,50 | 90,00 | 85,56 | 9,94 |
| 0,50 | 10,00 | 0,00 | 0,00 | 13,00 | 80,00 | 90,44 | 9,75 |
| 1,00 | 20,00 | 0,09 | 0,19 | 13,50 | 70,00 | 95,16 | 9,45 |
| 1,50 | 30,00 | 0,38 | 0,57 | 14,00 | 60,00 | 99,70 | 9,07 |
| 2,00 | 40,00 | 0,94 | 1,13 | 14,50 | 50,00 | 104,01 | 8,63 |
| 2,50 | 50,00 | 1,87 | 1,86 | 15,00 | 40,00 | 108,09 | 8,15 |
| 3,00 | 60,00 | 3,24 | 2,74 | 15,50 | 30,00 | 111,91 | 7,64 |
| 3,50 | 70,00 | 5,10 | 3,73 | 16,00 | 20,00 | 115,45 | 7,09 |
| 4,00 | 80,00 | 7,49 | 4,78 | 16,50 | 10,00 | 118,71 | 6,51 |
| 4,50 | 90,00 | 10,42 | 5,86 | 17,00 | 0,00 | 121,65 | 5,89 |
| 5,00 | 100,00 | 13,87 | 6,90 | 17,50 | -10,00 | 124,26 | 5,23 |
| 5,50 | 100,00 | 17,82 | 7,88 | 18,00 | -20,00 | 126,52 | 4,52 |
| 6,00 | 100,00 | 22,11 | 8,58 | 18,50 | -30,00 | 128,39 | 3,75 |
| 6,50 | 100,00 | 26,64 | 9,07 | 19,00 | -40,00 | 129,85 | 2,91 |
| 7,00 | 100,00 | 31,33 | 9,39 | 19,50 | -50,00 | 130,85 | 2,00 |
| 7,50 | 100,00 | 36,13 | 9,59 | 20,00 | -60,00 | 131,34 | 0,97 |
| 8,00 | 100,00 | 40,99 | 9,72 | 20,50 | -70,00 | 131,25 | -0,18 |
| 8,50 | 100,00 | 45,89 | 9,81 |  |  |  |  |
| 9,00 | 100,00 | 50,82 | 9,86 |  |  |  |  |
| 9,50 | 100,00 | 55,77 | 9,89 |  |  |  |  |
| 10,00 | 100,00 | 60,73 | 9,91 |  |  |  |  |
| 10,50 | 100,00 | 65,69 | 9,92 |  |  |  |  |
| 11,00 | 100,00 | 70,65 | 9,93 |  |  |  |  |
| 11,50 | 100,00 | 75,62 | 9,94 |  |  |  |  |
| 12,00 | 100,00 | 80,59 | 9,94 |  |  |  |  |

Основные параметры движения малого рабочего катера представлены на рисунке 4.



1. Основные параметры движения малого рабочего катера

Исходя из полученных результатов можно сделать следующие выводы о динамических характеристиках малого рабочего катера:

* Время набора максимальной скорости: 9.95 м/с – 12 с; при этом судно проходит 80.59 м;
* Общее время торможения: 8.5 с на расстоянии 50.66 м;
* Общее время движения судна составило 20.5 с;
* Общее пройденное расстояние: 131 м.

# **Моделирование движения корабля на подводных крыльях**

Исходные данные малого рабочего катера представлены в таблице 5.

Таблица 5. Исходные данные КПК

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Наименование | W, т | N, л.с. | vk, уз | v1max, уз | v2max, уз |
| 29 | КПК | 50 | 3500 | 16 | 19.2 | 46 |

Исходная математическая модель корабля на подводных крыльях представлена в формуле (2.6). Однако уточненную математическую модель можно получить, если учесть формулу (1.9). Тогда разностная модель корабля на подводных крыльях будет следующей:



Корабль на подводных крыльях имеет водоизмещение 50 т, следовательно, примем максимально допустимую скорость изменения силы тяги , а шаг интегрирования примем  Погрешность расчетов также примем е = 0,005.

Листинг программы, реализующий моделирование движения корабля на подводных крыльях представлен ниже.

*Листинг 3 – Код реализации разностной модели движения КПК*

clear, clc, close all

%КПК

W0 = 50; % тонн

N0 = 3500 ;% лс

vk0 = 16; %уз

v1max0 = 19.2; %уз

v2max0 = 46; %уз

%перевод в си

W = W0\*1000; %кг

N = N0\*735.5; % Вт

vk = vk0\*0.51 % м/с начало глиссирования

v1max = v1max0\*0.51 % м/с выход на крылья

v2max = v2max0\*0.51 % м/с

Fmax = N/v2max; % N = F\*v

dFmax = 0.15\*Fmax;

dt = 1;

dPmax = dt\*dFmax/Fmax\*100;

Pmax = 100; % максимальная тяга

dP = 10;

if (dP >= dPmax)

dP = dPmax;

end

i = 1;

t = [0];

v = [0];

P = [0];

x = [0];

dx = 0;

e = 0.005;% допустимая погрешность, т.к. идеально не попадем

%разгон

A1 = Fmax/v1max^2;

A2 = Fmax/v2max^2;

while ( (v2max-v(i)) > e)

if (v(i)<vk)

A = Fmax/v1max^2;

end

if ((v(i) >= vk)&& (v(i) <= v1max) )

A = A1 - (v(i)-vk)\*(A1-A2)/(v1max-vk);

end

if (v(i) > v1max)

A = A2;

end

t = [t t(i)+dt];

*Листинг 3 – Продолжение*

x = [x x(i)+dx+(P(i)\*Fmax\*dt^2/100-A\*dx\*abs(dx))/W];

v = [v (x(i+1)-x(i))/dt];

if (P(i) < Pmax)

P = [P (P(i)+dP)];

else

P = [P Pmax];

end

i = i + 1;

dx = x(i)-x(i-1);

if ((v(i)-v(i-1))<e/10 && i > 3) %учитывает, что мощность может быть не 100%

break

end

end

%торможение

while ( v(i) > e)

if (v(i)<vk)

A = Fmax/v1max^2;

end

if ((v(i) >= vk)&& (v(i) <= v1max) )

A = A1 - (v(i)-vk)\*(A1-A2)/(v1max-vk);

end

if (v(i) > v1max)

A = A2;

end

t = [t t(i)+dt];

x = [x x(i)+dx+(P(i)\*Fmax\*dt^2/100-A\*dx\*abs(dx))/W];

v = [v (x(i+1)-x(i))/dt];

if (P(i) > -Pmax)

P = [P (P(i)-dP)];

else

P = [P -Pmax];

end

i = i + 1;

dx = x(i)-x(i-1);

end

KPK\_table = [t' P' x' v']; % формирование таблицы

figure(3)

subplot(3,1,1)

plot(t,x,'Linewidth',1.5); grid on;

xlabel('t, c','FontSize',12);

ylabel('x, м','FontSize',12);

subplot(3,1,2)

plot(t,v,'Linewidth',1.5);grid on;

xlabel('t, c','FontSize',12);

ylabel('v, м/с','FontSize',12);

subplot(3,1,3)

plot(t,P,'Linewidth',1.5);grid on;

xlabel('t, c','FontSize',12);

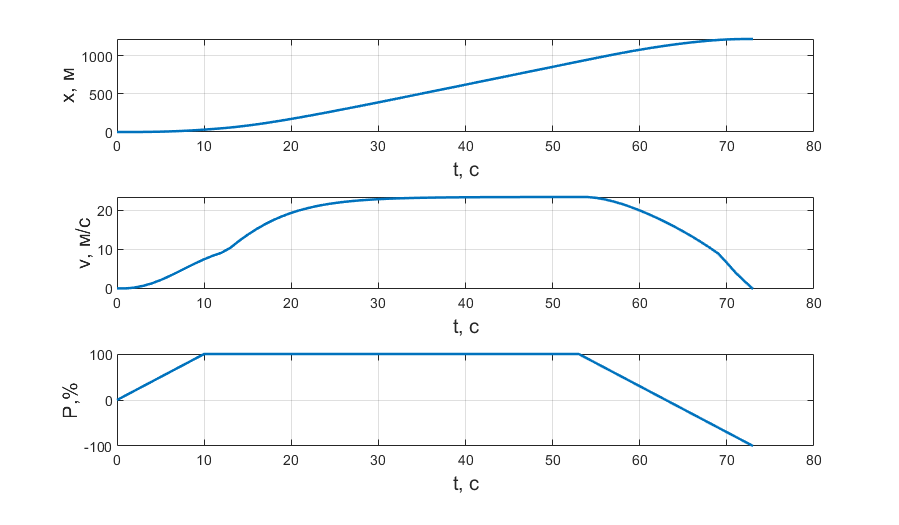
ylabel('P,%','FontSize',12);

Результаты моделирования разностной модели КПК сведены в таблицу 6.

Таблица 6. Результаты моделирования КПК

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Режим | T, с | P, % | xi+1, м | vi+1, м/с | Режим | T, с | P, % | xi+1, м | vi+1, м/с |
| Разгон | 0,00 | 0,00 | 0,00 | 0,00 | Разгон | 37,00 | 100,00 | 552,69 | 23,33 |
| 1,00 | 10,00 | 0,00 | 0,00 | 38,00 | 100,00 | 576,04 | 23,35 |
| 2,00 | 20,00 | 0,22 | 0,22 | 39,00 | 100,00 | 599,41 | 23,37 |
| 3,00 | 30,00 | 0,88 | 0,66 | 40,00 | 100,00 | 622,80 | 23,39 |
| 4,00 | 40,00 | 2,18 | 1,31 | 41,00 | 100,00 | 646,20 | 23,40 |
| 5,00 | 50,00 | 4,33 | 2,14 | 42,00 | 100,00 | 669,61 | 23,41 |
| 6,00 | 60,00 | 7,46 | 3,14 | 43,00 | 100,00 | 693,03 | 23,42 |
| 7,00 | 70,00 | 11,69 | 4,23 | 44,00 | 100,00 | 716,46 | 23,43 |
| 8,00 | 80,00 | 17,05 | 5,36 | 45,00 | 100,00 | 739,90 | 23,43 |
| 9,00 | 90,00 | 23,50 | 6,45 | 46,00 | 100,00 | 763,33 | 23,44 |
| 10,00 | 100,00 | 30,98 | 7,48 | 47,00 | 100,00 | 786,78 | 23,44 |
| 11,00 | 100,00 | 39,37 | 8,39 | 48,00 | 100,00 | 810,22 | 23,45 |
| 12,00 | 100,00 | 48,53 | 9,16 | 49,00 | 100,00 | 833,67 | 23,45 |
| 13,00 | 100,00 | 58,94 | 10,41 | 50,00 | 100,00 | 857,12 | 23,45 |
| 14,00 | 100,00 | 71,12 | 12,17 | 51,00 | 100,00 | 880,58 | 23,45 |
| 15,00 | 100,00 | 84,89 | 13,78 | 52,00 | 100,00 | 904,03 | 23,45 |
| 16,00 | 100,00 | 100,11 | 15,21 | 53,00 | 100,00 | 927,49 | 23,46 |
| 17,00 | 100,00 | 116,60 | 16,49 | Торможение | 54,00 | 90,00 | 950,94 | 23,46 |
| 18,00 | 100,00 | 134,19 | 17,60 | 55,00 | 80,00 | 974,18 | 23,24 |
| 19,00 | 100,00 | 152,75 | 18,56 | 56,00 | 70,00 | 997,02 | 22,84 |
| 20,00 | 100,00 | 172,13 | 19,38 | 57,00 | 60,00 | 1019,31 | 22,30 |
| 21,00 | 100,00 | 192,20 | 20,08 | 58,00 | 50,00 | 1040,94 | 21,63 |
| 22,00 | 100,00 | 212,87 | 20,66 | 59,00 | 40,00 | 1061,81 | 20,86 |
| 23,00 | 100,00 | 234,02 | 21,16 | 60,00 | 30,00 | 1081,81 | 20,00 |
| 24,00 | 100,00 | 255,59 | 21,57 | 61,00 | 20,00 | 1100,88 | 19,07 |
| 25,00 | 100,00 | 277,49 | 21,91 | 62,00 | 10,00 | 1118,93 | 18,06 |
| 26,00 | 100,00 | 299,68 | 22,19 | 63,00 | 0,00 | 1135,91 | 16,98 |
| 27,00 | 100,00 | 322,10 | 22,42 | 64,00 | -10,00 | 1151,74 | 15,83 |
| 28,00 | 100,00 | 344,71 | 22,61 | 65,00 | -20,00 | 1166,35 | 14,61 |
| 29,00 | 100,00 | 367,47 | 22,77 | 66,00 | -30,00 | 1179,66 | 13,32 |
| 30,00 | 100,00 | 390,37 | 22,89 | 67,00 | -40,00 | 1191,62 | 11,95 |
| 31,00 | 100,00 | 413,36 | 23,00 | 68,00 | -50,00 | 1202,12 | 10,51 |
| 32,00 | 100,00 | 436,45 | 23,08 | 69,00 | -60,00 | 1211,09 | 8,97 |
| 33,00 | 100,00 | 459,60 | 23,15 | 70,00 | -70,00 | 1217,65 | 6,56 |
| 34,00 | 100,00 | 482,81 | 23,21 | 71,00 | -80,00 | 1221,70 | 4,04 |
| 35,00 | 100,00 | 506,07 | 23,26 | 72,00 | -90,00 | 1223,61 | 1,91 |
| 36,00 | 100,00 | 529,36 | 23,29 | 73,00 | -100,00 | 1223,46 | -0,15 |

Основные параметры движения корабля на подводных крыльях представлены на рисунке 5.



1. Основные параметры движения корабля на подводных крыльях

Исходя из полученных результатов можно сделать следующие выводы о динамических характеристиках корабля на подводных крыльях:

* Время набора максимальной скорости: 23.46 м/с – 53 с; при этом судно проходит 927.5 м;
  + В водоизмещающем режиме – 8.8 с;
  + В режиме глиссирования – 1 с;
  + На крыльях – 47 с;
* Общее время торможения: 14.5 с на расстоянии 234 м;
* Общее время движения судна составило 61 с;
* Общее пройденное расстояние: 1014 м.

# **РЕЖИМ УПРАВЛЕНИЯ СКОРОСТЬЮ ХОДА**

Для реализации управления скоростью хода можно использовать регулятор. В данной работе в качестве регулятора был выбран ПД-регулятор. Это объясняется тем, что при использовании П-регулятора, при задании малых скоростей появляются колебания. Чтобы уменьшить колебания был использован ПД-регулятор. ПД-регулятор управляет мощностью двигателя. Управляющее воздействие формируется на основе значения ошибки:

,

где Vz – заданное значение скорости в м/с.

Расчет мощности двигателя для разностной модели будет выглядеть следующим образом:



где Kp – коэффициент пропорциональной составляющей, Kd – коэффициент дифференцирующей составляющей.

Листинг программы режима управления скорости хода НВК «Активный» представлен ниже. Заданная скорость – 2 м/с, коэффициенты регулятора kp = 10, kd = 80.

*Листинг 4 – Код реализации ПД регулятора*

clear, clc, close all

W0 = 314; % тонн

N0 = 480; % л.с.

v0 = 8.5; % уз

W = W0\*1000; %кг

N = N0\*735.5; % Вт

Vmax = v0\*0.51; % м/с

Fmax = N/Vmax; % N = F\*v

A = Fmax/Vmax^2;

dFmax = 0.2\*Fmax;

dt = 1;

dPmax = dt\*dFmax/Fmax\*100;

*Листинг 4 – Продолжение*

dP = 20;

if (dP >= dPmax)

dP = dPmax;

end

% Н.У.

Pmax = 100; % максимальная тяга

i = 1;

t = [0];

v = [0];

P = [0];

x = [0];

dx = 0;

eps = 0.005; % допустимая погрешность

Vz = 2; % задание м/с

kp = 10; % коэффициент ПД регулятора

kd = 80; % коэф ПД регулятора

prev\_e = Vz - v(1);

while(t(i) < 100)

t = [t t(i) + dt];

x = [x x(i) + dx + ((P(i)\*Fmax\*dt^2)/100 - A\*dx\*abs(dx))/W];

v = [v (x(i+1) - x(i))/dt];

e = Vz - v(i);

de = (e - prev\_e)/dt;

dP = kp\*e + kd\*de;

prev\_e = e;

if (abs(dP) >= dPmax)

dP = dPmax;

end

if (P(i) + dP > 100)

P = [P 100];

end

if (P(i) + dP <= 100)

P = [P P(i) + dP];

end

i = i + 1;

dx = x(i) - x(i-1);

end

figure(1)

subplot(3,1,1)

plot(t,x,'Linewidth',1.5); grid on;

xlabel('t, c','FontSize',12);

ylabel('x, м','FontSize',12);

subplot(3,1,2)

plot(t,v,'Linewidth',1.5);grid on;

xlabel('t, c','FontSize',12);

ylabel('v, м/с','FontSize',12);

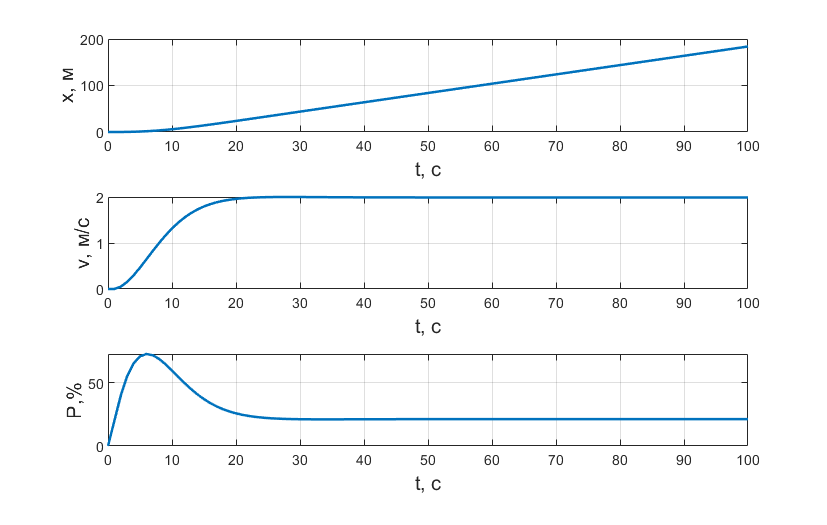
subplot(3,1,3)

plot(t,P,'Linewidth',1.5);grid on;

xlabel('t, c','FontSize',12);

ylabel('P,%','FontSize',12);

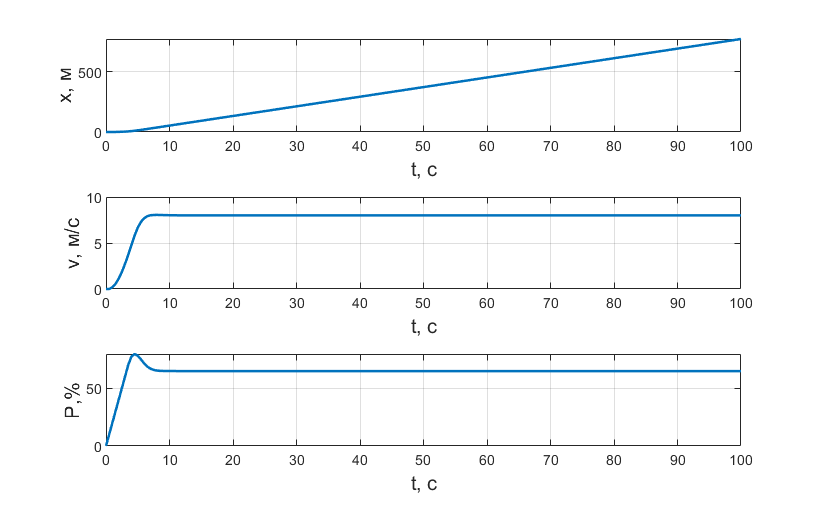
Результат работы программы представлен на рисунке 6.



1. Режим управления скоростью НВК «Активный»

Код реализации регулятора скорости для другого типа НВК отличается только параметрами судна и регулятора. Заданная скорость – 8 м/с, коэффициенты регулятора kp = 4, kd = 5. Листинг не приводится.

Результат работы программы представлен на рисунке 7.



1. Режим управления скоростью малого рабочего катера

Листинг программы режима управления скорости хода корабля на подводных крыльях представлен ниже. Заданная скорость – 8 м/с, коэффициенты регулятора kp = 3, kd = 5.

*Листинг 4 – Код реализации ПД регулятора КПК*

clear, clc, close all

%КПК

W0 = 50; % тонн

N0 = 3500 ;% лс

vk0 = 16; %уз

v1max0 = 19.2; %уз

v2max0 = 46; %уз

%перевод в си

W = W0\*1000; %кг

N = N0\*735.5; % Вт

vk = vk0\*0.51 % м/с начало глиссирования

v1max = v1max0\*0.51 % м/с выход на крылья

v2max = v2max0\*0.51 % м/с

Fmax = N/v2max; % N = F\*v

dFmax = 0.15\*Fmax;

dt = 0.5;

dPmax = dt\*dFmax/Fmax\*100;

Pmax = 100; % максимальная тяга

dP = 10;

if (dP >= dPmax)

dP = dPmax;

end

i = 1;

t = [0];

v = [0];

P = [0];

x = [0];

dx = 0;

Vz = 8; % задание м/с

kp = 3; % коэффициент ПД регулятора

kd = 5; % коэф ПД регулятора

prev\_e = Vz - v(1);

%разгон

A1 = Fmax/v1max^2;

A2 = Fmax/v2max^2;

while(t(i) < 100)

if (v(i)<vk)

A = Fmax/v1max^2;

end

if ((v(i) >= vk)&& (v(i) <= v1max) )

A = A1 - (v(i)-vk)\*(A1-A2)/(v1max-vk);

end

if (v(i) > v1max)

A = A2;

end

t = [t t(i) + dt];

x = [x x(i) + dx + (P(i)\*Fmax\*dt^2/100 - A\*dx\*abs(dx))/W];

v = [v (x(i+1) - x(i))/dt];

*Листинг 4 – Продолжение*

e = Vz - v(i);

de = (e - prev\_e)/dt;

dP = kp\*e + kd\*de;

prev\_e = e;

if (abs(dP) >= dPmax)

dP = dPmax;

end

if (P(i) + dP > 100)

P = [P 100];

end

if (P(i)+dP <= 100)

P = [P P(i)+dP];

end

i = i + 1;

dx = x(i)-x(i-1);

end

figure(3)

subplot(3,1,1)

plot(t,x,'Linewidth',1.5); grid on;

xlabel('t, c','FontSize',12);

ylabel('x, м','FontSize',12);

subplot(3,1,2)

plot(t,v,'Linewidth',1.5);grid on;

xlabel('t, c','FontSize',12);

ylabel('v, м/с','FontSize',12);

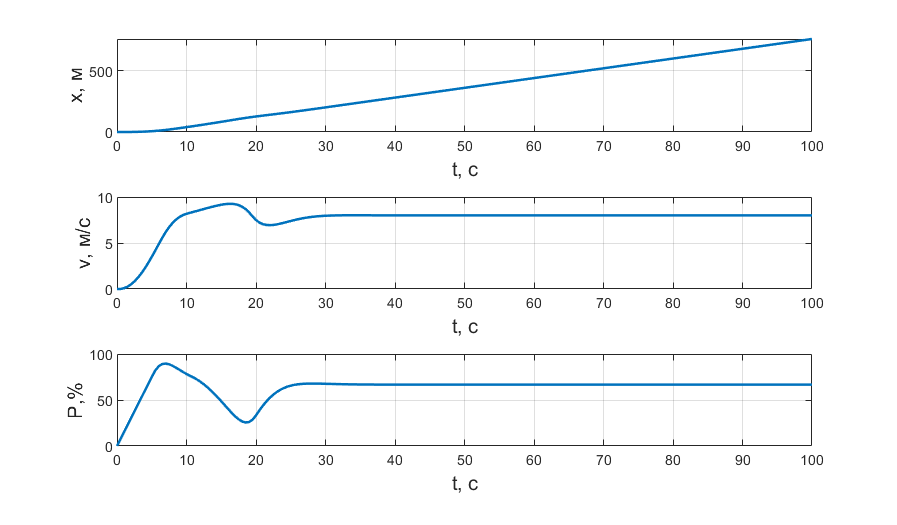
subplot(3,1,3)

plot(t,P,'Linewidth',1.5);grid on;

xlabel('t, c','FontSize',12);

ylabel('P,%','FontSize',12);

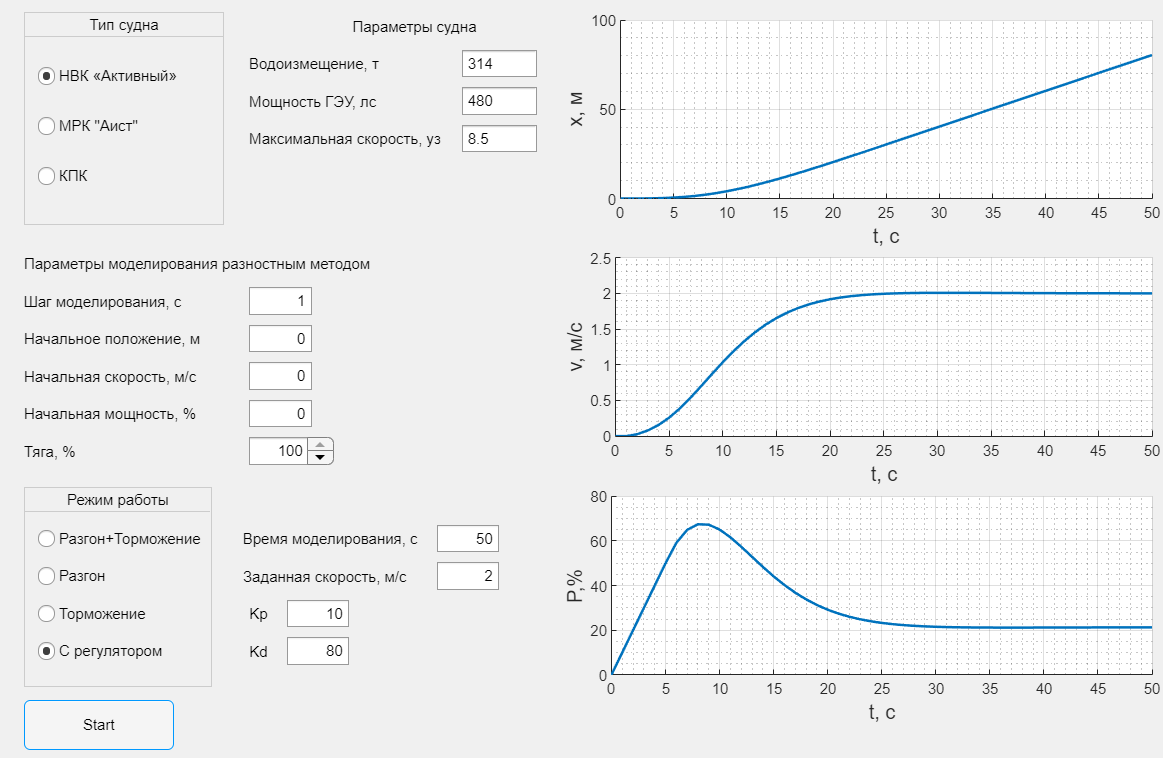
Результат работы программы представлен на рисунке 8.



1. Режим управления скоростью корабля на подводных крыльях

# **ПРИЛОЖЕНИЕ С ГРАФИЧЕСКИМ ИНТЕРФЕЙСОМ**

Для удобства взаимодействия с пользователем был разработан графический интерфейс. Окно приложения продемонстрированно на рисунке 9.



1. Начальное окно приложения

При запуске приложения открывается начальное окно. При выборе судна автоматически заполняются параметры судна.

После выбора судна необходимо заполнить параметры моделирования. Моделирование осуществляется разностным методом. Также имеется возможность задавать максимальную тягу двигателя с дискретностью 10%.

После выбора режима работы судна и нажатия кнопки «Start» в правой области приложения появятся графики, описывающие динамику движения судна.

Приложение осуществляет моделирование трех типов кораблей. При выборе типа корабля его параметры заполняются автоматически, при это их можно изменять, первые два типа моделируются по одной модели, а КПК по другой. При выборе КПК в окне интерфейса появляются дополнительные поля.

В приложении имеется режим работы с регулятором, который обеспечивает поддержание заданной скорости. При выборе режима работы с регулятором появляются дополнительные поля, с возможностью редактирования параметров ПД-регулятора.

Для работы с регулятором нужно задать время моделирования, а также заданную скорость. В параметрах коэффициентов регулятора автоматически появляются значения коэффициентов. При необходимости их можно менять. Для каждого судна свои параметры коэффициентов.

Сценарий приложения представлен в таблице 7.

Таблица 7. Сценарий приложения

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Событие | | Идентификатор элемента управления | | Действие | |
| Запуск приложения | | startupFcn | | Отключить видимость графиков, доп. параметров КПК;  Инициализировать переменные; | |
| Выбор типа судна | | Ship | | Установка параметров судна в соответствии с исходным заданием;  При выборе КПК появляются дополнительные параметры; | |
|  | Выбор параметров моделирования | | | |  |
| Шаг интегрирования | | dtValueChanged | | Присвоение заданного значения | |
| Начального значения координаты | | x0ValueChanged | | Присвоение заданного значения | |
| Начальной скорости | | V0ValueChanged | | Присвоение заданного значения | |
| Начальной мощности | | P0ValueChanged | | Присвоение заданного значения | |
| Максимальной тяги | | PmaxValueChanged | | Присвоение заданного значения | |
| Водоизмещения | | Field\_WValueChanged | | Присвоение заданного значения | |
| Мощности ГЭУ | | Field\_NValueChanged | | Присвоение заданного значения | |
| Максимальной скорости | | Field\_VValueChanged | | Присвоение заданного значения | |
| Скорости V1 КПК | | | Field\_V1ValueChanged | Присвоение заданного значения | |
| Скорости V2 КПК | | | Field\_V2ValueChanged | Присвоение заданного значения | |
| Коэффициента kp | | | Field\_kpValueChanged | Присвоение заданного значения | |
| Коэффициента kd | | | Field\_kdValueChanged | Присвоение заданного значения | |
| Выбор режима работы | | | mode | Установка соответствующего флага.  В режиме работы с регулятором появляются дополнительные параметры. Параметры регулятора автоматически заполняются. | |
| Задание требуемой скорости | | | VzValueChanged | Присвоение заданного значения | |
| Задание времени моделирования | | | tValueChanged | Присвоение заданного значения | |
| Запуск | | | Start | Включение отображения графиков;  Моделирование движения выбранного судна согласно режиму работы. | |

Листинг кода приложения представлен в приложении А.

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения индивидуального домашнего задания были выработаны умения и практические навыки составления и написания программных моделей для исследования в среде MATLAB, создания интерфейсов для вывода результатов моделирования с использованием элементов MATLAB GUI.

Были составлены и написаны программные модели движения для двух типов судов разностным методом. Разработанные модели были исследованы в режимах разгона и торможения. Для осуществления режимом управления скоростью судна был применен ПД-регулятор.

Разработанный графический интерфейс позволяет моделировать движение судов в различных режимах работы.

# **ПРИЛОЖЕНИЕ А**

**Листинг приложения**

% Callbacks that handle component events

methods (Access = private)

% Code that executes after component creation

function startupFcn(app, a)

app.Button\_NVK.Value = true;

app.Button\_Kater.Value = false;

app.Button\_KPK.Value = false;

app.Field\_V1max.Visible = 0;

app.Field\_V2max.Visible = 0;

app.LabelV1max.Visible=0;

app.LabelV2max.Visible=0;

app.Label\_Vz.Visible = false;

app.Label\_t.Visible = false;

app.EditField\_Vz.Visible = false;

app.EditField\_t.Visible = false;

app.UIAxes\_p.Visible = false;

app.UIAxes\_v.Visible = false;

app.UIAxes\_x.Visible = false;

app.Label\_kd.Visible = false;

app.Label\_kp.Visible = false;

app.EditField\_kp.Visible = false;

app.EditField\_kd.Visible = false;

%инициализация

app.W = 0;

app.N = 0;

app.Vmax = 0;

app.Vk = 0;

app.V1max = 0;

app.V2max = 0;

app.Flag\_mode = 4;

app.Flag\_ship = 1;

app.Spinner.Value = 100;

app.Pmax = 100;

end

% Selection changed function: ButtonGroup

function Ship(app, event)

app.Field\_V1max.Visible = 0;

app.Field\_V2max.Visible = 0;

app.LabelV1max.Visible=0;

app.LabelV2max.Visible=0;

if app.Button\_NVK.Value == true

app.Field\_W.Value = '314';

app.Field\_N.Value = '480';

app.Field\_V.Value = '8.5';

app.kp = 10;

app.kd = 80;

app.LabelV.Text='Максимальная скорость, уз';

%перевод в си

app.W = str2double(app.Field\_W.Value)\*1000;

app.N = str2double(app.Field\_N.Value)\*735.5;

app.Vmax = str2double(app.Field\_V.Value)\*0.51;

app.Flag\_ship = 1;

app.flag\_error = 1;

end

if app.Button\_Kater.Value == true

app.Field\_W.Value = '4.6';

app.Field\_N.Value = '235';

app.Field\_V.Value = '19.5';

app.LabelV.Text='Максимальная скорость, уз';

app.W = str2double(app.Field\_W.Value)\*1000;

app.N = str2double(app.Field\_N.Value)\*735.5;

app.Vmax = str2double(app.Field\_V.Value)\*0.51;

app.Flag\_ship = 2;

app.flag\_error = 1;

app.kp = 4;

app.kd = 5;

end

if app.Button\_KPK.Value == true

app.Field\_W.Value = '50';

app.Field\_N.Value = '3500';

app.Field\_V.Value = '16';

app.Field\_V1max.Visible = 1;

app.Field\_V2max.Visible = 1;

app.LabelV1max.Visible=1;

app.LabelV2max.Visible=1;

app.LabelV.Text='Vk, уз';

app.Field\_V1max.Value = '19.2';

app.Field\_V2max.Value = '46';

app.W = str2double(app.Field\_W.Value)\*1000;

app.N = str2double(app.Field\_N.Value)\*735.5;

app.Vk = str2double(app.Field\_V.Value)\*0.51;

app.V1max = str2double(app.Field\_V1max.Value)\*0.51;

app.V2max = str2double(app.Field\_V2max.Value)\*0.51;

app.Flag\_ship = 3;

app.flag\_error = 1;

app.kp = 3;

app.kd = 5;

end

end

% Value changed function: EditField\_dt

function dtValueChanged(app, event)

value = app.EditField\_dt.Value;

app.dt = value;

end

% Value changed function: EditField\_x0

function x0ValueChanged(app, event)

value = app.EditField\_x0.Value;

app.x0 = value;

end

% Value changed function: EditField\_V0

function V0ValueChanged(app, event)

value = app.EditField\_V0.Value;

app.V0 = value;

end

% Value changed function: EditField\_P0

function P0ValueChanged(app, event)

value = app.EditField\_P0.Value;

app.P0 = value;

end

% Value changed function: Spinner

function PmaxValueChanged(app, event)

value = app.Spinner.Value;

app.Pmax = value;

end

% Selection changed function: ButtonGroup\_2

function mode(app, event)

% Button pushed function: StartButton

function Start(app, event)

if app.Flag\_mode ~= 0

app.UIAxes\_p.Visible = true;

app.UIAxes\_v.Visible = true;

app.UIAxes\_x.Visible = true;

Fmax = app.N/app.Vmax;% N = F\*v

A = Fmax/app.Vmax^2;

i = 1;

t = [0];

v = [app.EditField\_V0.Value];

P = [app.EditField\_P0.Value];

x = [app.EditField\_x0.Value];

dx = 0;

eps = 0.005; % допустимая погрешность

if app.Flag\_ship == 1

dFmax = 0.1\*Fmax;

elseif app.Flag\_ship == 2

dFmax = 0.2\*Fmax;

elseif app.Flag\_ship == 3

dFmax = 0.2\*Fmax;

end

app.dt = app.EditField\_dt.Value;

dPmax = app.dt\*dFmax/Fmax\*100;

dP = 10;

if (dP >= dPmax)

dP = dPmax;

end

if app.Flag\_ship == 1 || app.Flag\_ship == 2

if app.Flag\_mode == 4 || app.Flag\_mode == 1 || app.Flag\_mode == 3

prev\_e = app.Vz-v(1);

while ( (app.Vmax-v(i)) > eps || app.Flag\_mode == 3)

t = [t t(i) + app.dt];

x = [x x(i) + dx + (P(i)\*Fmax\*app.dt^2/100 - A\*dx\*abs(dx))/app.W];

v = [v (x(i+1) - x(i))/app.dt];

if app.Flag\_mode == 3 %reg

e = app.Vz - v(i);

de = (e-prev\_e)/app.dt;

dP = app.kp\*e+app.kd\*de;

prev\_e = e;

if (abs(dP) >= dPmax)

dP = dPmax;

end

end

if (P(i) < app.Pmax)

P = [P P(i) + dP];

else

P = [P app.Pmax];

end

i = i + 1;

dx = x(i) - x(i-1);

if t(i) >= app.EditField\_t.Value && app.Flag\_mode == 3 %timeout

break

end

if ((v(i) - v(i-1)) < eps/10 && i > 3 && app.Flag\_mode ~= 3) %учитывает, что мощность может быть не 100%

break

end

end

end % разгон или разгон + тормоз

if app.Flag\_mode == 4 || app.Flag\_mode == 2

while ( v(i) > eps)

t = [t t(i)+app.dt];

x = [x x(i)+dx+(P(i)\*Fmax\*app.dt^2/100-A\*dx\*abs(dx))/app.W];

v = [v (x(i+1)-x(i))/app.dt];

if (P(i) > -app.Pmax)

P = [P (P(i)-dP)];

else

P = [P -app.Pmax];

end

i = i + 1;

dx = x(i)-x(i-1);

end

end % тормоз или разгон плюс тормоз

elseif app.Flag\_ship == 3 % KPK

Fmax = app.N/app.V2max;% N = F\*v

A1 = Fmax/app.V1max^2;

A2 = Fmax/app.V2max^2;

dFmax = 0.2\*Fmax;

dPmax = app.dt\*dFmax/Fmax\*100;

if app.Flag\_mode == 4 || app.Flag\_mode == 1 || app.Flag\_mode == 3

prev\_e = app.Vz-v(1);

while ( (app.V2max-v(i)) > eps || app.Flag\_mode == 3)

if (v(i)<app.Vk)

A = Fmax/app.V1max^2;

end

if ((v(i) >= app.Vk)&& (v(i) <= app.V1max) )

A = A1 - (v(i)-app.Vk)\*(A1-A2)/(app.V1max-app.Vk);

end

if (v(i) > app.V1max)

A = A2;

end

t = [t t(i)+app.dt];

x = [x x(i)+dx+(P(i)\*Fmax\*app.dt^2/100-A\*dx\*abs(dx))/app.W];

v = [v (x(i+1)-x(i))/app.dt];

if app.Flag\_mode == 3 %reg

e = app.Vz - v(i);

de = (e-prev\_e)/app.dt;

dP = app.kp\*e+app.kd\*de;

prev\_e = e;

if (dP >= dPmax)

dP = dPmax;

end

if (dP <= -dPmax)

dP = - dPmax;

end

end

if (P(i)+dP <= app.Pmax)

P = [P (P(i)+dP)];

else

P = [P app.Pmax];

end

i = i + 1;

dx = x(i)-x(i-1);

if t(i) >= app.EditField\_t.Value && app.Flag\_mode == 3 %timeout

break

end

if ((v(i)-v(i-1))<eps/10 && i > 3 && app.Flag\_mode ~= 3) %учитывает, что мощность может быть не 100%

break

end

end

end %

if app.Flag\_mode == 4 || app.Flag\_mode == 2

%торможение

while ( v(i) > eps)

if (v(i)<app.Vk)

A = Fmax/app.V1max^2;

end

if ((v(i) >= app.Vk)&& (v(i) <= app.V1max) )

A = A1 - (v(i)-app.Vk)\*(A1-A2)/(app.V1max-app.Vk);

end

if (v(i) > app.V1max)

A = A2;

end

t = [t t(i)+app.dt];

x = [x x(i)+dx+(P(i)\*Fmax\*app.dt^2/100-A\*dx\*abs(dx))/app.W];

v = [v (x(i+1)-x(i))/app.dt];

if (P(i) > -app.Pmax)

P = [P (P(i)-dP)];

else

P = [P -app.Pmax];

end

i = i + 1;

dx = x(i) - x(i-1);

end

end

end % type of ship

plot(app.UIAxes\_x,t,x,'Linewidth',1.5);

xlabel(app.UIAxes\_x,'t, c','FontSize',12);

ylabel(app.UIAxes\_x,'x, м','FontSize',12);

plot(app.UIAxes\_v,t,v,'Linewidth',1.5);

xlabel(app.UIAxes\_v,'t, c','FontSize',12);

ylabel(app.UIAxes\_v,'v, м/с','FontSize',12);

plot(app.UIAxes\_p,t,P,'Linewidth',1.5);

xlabel(app.UIAxes\_p,'t, c','FontSize',12);

ylabel(app.UIAxes\_p,'P,%','FontSize',12);

else

helpdlg('Ошибка','Help');

end

end

% Value changed function: EditField\_t

function tValueChanged(app, event)

value = app.EditField\_t.Value;

app.t\_mod = value;

end

% Value changed function: EditField\_Vz

function VzValueChanged(app, event)

value = app.EditField\_Vz.Value;

app.Vz = value;

end

% Value changed function: Field\_W

function Field\_WValueChanged(app, event)

value = app.Field\_W.Value;

app.W = str2double(value)\*1000;

end

% Value changed function: Field\_N

function Field\_NValueChanged(app, event)

value = app.Field\_N.Value;

app.N = str2double(value)\*735.5;

end

% Value changed function: Field\_V

function Field\_VValueChanged(app, event)

value = app.Field\_V.Value;

app.Vmax = str2double(value)\*0.51;

end

% Value changed function: Field\_V1max

function Field\_V1maxValueChanged(app, event)

value = app.Field\_V1max.Value;

app.V1max = str2double(value)\*0.51;

end

% Value changed function: Field\_V2max

function Field\_V2maxValueChanged(app, event)

value = app.Field\_V2max.Value;

app.V2max = str2double(value)\*0.51;

end

% Value changed function: EditField\_kp

function EditField\_kpValueChanged(app, event)

value = app.EditField\_kp.Value;

app.kp = value;

end

% Value changed function: EditField\_kd

function EditField\_kdValueChanged(app, event)

value = app.EditField\_kd.Value;

app.kd = value;

end

end

% Component initialization

methods (Access = private)

% Create UIFigure and components

function createComponents(app)

% Create UIFigure and hide until all components are created

app.UIFigure = uifigure('Visible', 'off');

app.UIFigure.Color = [0.9412 0.9412 0.9412];

app.UIFigure.Position = [100 100 933 610];

app.UIFigure.Name = 'MATLAB App';

% Create UIAxes\_x

app.UIAxes\_x = uiaxes(app.UIFigure);

xlabel(app.UIAxes\_x, 'X')

ylabel(app.UIAxes\_x, 'Y')

zlabel(app.UIAxes\_x, 'Z')

app.UIAxes\_x.XGrid = 'on';

app.UIAxes\_x.XMinorGrid = 'on';

app.UIAxes\_x.YGrid = 'on';

app.UIAxes\_x.YMinorGrid = 'on';

app.UIAxes\_x.Tag = 'UIAxes\_x';

app.UIAxes\_x.Position = [451 411 480 190];

% Create UIAxes\_v

app.UIAxes\_v = uiaxes(app.UIFigure);

xlabel(app.UIAxes\_v, 'X')

ylabel(app.UIAxes\_v, 'Y')

zlabel(app.UIAxes\_v, 'Z')

app.UIAxes\_v.XGrid = 'on';

app.UIAxes\_v.XMinorGrid = 'on';

app.UIAxes\_v.YGrid = 'on';

app.UIAxes\_v.YMinorGrid = 'on';

app.UIAxes\_v.Tag = 'UIAxes\_v';

app.UIAxes\_v.Position = [451 221 480 190];

% Create UIAxes\_p

app.UIAxes\_p = uiaxes(app.UIFigure);

xlabel(app.UIAxes\_p, 'X')

ylabel(app.UIAxes\_p, 'Y')

zlabel(app.UIAxes\_p, 'Z')

app.UIAxes\_p.XGrid = 'on';

app.UIAxes\_p.XMinorGrid = 'on';

app.UIAxes\_p.YGrid = 'on';

app.UIAxes\_p.YMinorGrid = 'on';

app.UIAxes\_p.Tag = 'UIAxes\_p';

app.UIAxes\_p.Position = [451 31 480 190];

% Create ButtonGroup

app.ButtonGroup = uibuttongroup(app.UIFigure);

app.ButtonGroup.SelectionChangedFcn = createCallbackFcn(app, @Ship, true);

app.ButtonGroup.TitlePosition = 'centertop';

app.ButtonGroup.Title = 'Тип судна';

app.ButtonGroup.Tag = 'ButtonGroup';

app.ButtonGroup.Position = [21 431 160 170];

% Create Button\_NVK

app.Button\_NVK = uiradiobutton(app.ButtonGroup);

app.Button\_NVK.Tag = 'Button\_NVK';

app.Button\_NVK.Text = 'НВК «Активный»';

app.Button\_NVK.Position = [11 108 116 22];

app.Button\_NVK.Value = true;

% Create Button\_Kater

app.Button\_Kater = uiradiobutton(app.ButtonGroup);

app.Button\_Kater.Tag = 'Button\_Kater';

app.Button\_Kater.Text = 'МРК "Аист"';

app.Button\_Kater.Position = [11 68 85 22];

% Create Button\_KPK

app.Button\_KPK = uiradiobutton(app.ButtonGroup);

app.Button\_KPK.Tag = 'Button\_KPK';

app.Button\_KPK.Text = 'КПК';

app.Button\_KPK.Position = [11 28 65 22];

% Create Label\_2

app.Label\_2 = uilabel(app.UIFigure);

app.Label\_2.Position = [201 549 160 22];

app.Label\_2.Text = 'Водоизмещение, т';

% Create Field\_W

app.Field\_W = uieditfield(app.UIFigure, 'text');

app.Field\_W.ValueChangedFcn = createCallbackFcn(app, @Field\_WValueChanged, true);

app.Field\_W.Tag = 'EditFieldW';

app.Field\_W.Position = [371 549 60 22];

% Create Label\_3

app.Label\_3 = uilabel(app.UIFigure);

app.Label\_3.Position = [201 519 160 22];

app.Label\_3.Text = 'Мощность ГЭУ, лс';

% Create Field\_N

app.Field\_N = uieditfield(app.UIFigure, 'text');

app.Field\_N.ValueChangedFcn = createCallbackFcn(app, @Field\_NValueChanged, true);

app.Field\_N.Tag = 'EditFieldN';

app.Field\_N.Position = [371 519 60 22];

% Create Field\_V

app.Field\_V = uieditfield(app.UIFigure, 'text');

app.Field\_V.ValueChangedFcn = createCallbackFcn(app, @Field\_VValueChanged, true);

app.Field\_V.Tag = 'EditFieldV';

app.Field\_V.Position = [371 489 60 22];

% Create Label

app.Label = uilabel(app.UIFigure);

app.Label.Tag = 'Label';

app.Label.HorizontalAlignment = 'center';

app.Label.Position = [281 579 105 22];

app.Label.Text = 'Параметры судна';

% Create Field\_V2max

app.Field\_V2max = uieditfield(app.UIFigure, 'text');

app.Field\_V2max.ValueChangedFcn = createCallbackFcn(app, @Field\_V2maxValueChanged, true);

app.Field\_V2max.Tag = 'EditFieldV2max';

app.Field\_V2max.Position = [371 429 60 22];

% Create Field\_V1max

app.Field\_V1max = uieditfield(app.UIFigure, 'text');

app.Field\_V1max.ValueChangedFcn = createCallbackFcn(app, @Field\_V1maxValueChanged, true);

app.Field\_V1max.Tag = 'EditFieldV1max';

app.Field\_V1max.Position = [371 459 60 22];

% Create LabelV1max

app.LabelV1max = uilabel(app.UIFigure);

app.LabelV1max.Tag = 'LabelV1max';

app.LabelV1max.Position = [201 459 58 22];

app.LabelV1max.Text = 'V1max,уз';

% Create LabelV2max

app.LabelV2max = uilabel(app.UIFigure);

app.LabelV2max.Tag = 'LabelV2max';

app.LabelV2max.Position = [201 429 61 22];

app.LabelV2max.Text = 'V2max, уз';

% Create LabelV

app.LabelV = uilabel(app.UIFigure);

app.LabelV.Tag = 'LabelV';

app.LabelV.Position = [201 489 159 22];

app.LabelV.Text = 'Максимальная скорость, уз';

% Create Label\_4

app.Label\_4 = uilabel(app.UIFigure);

app.Label\_4.Tag = 'Label\_4';

app.Label\_4.Position = [21 389 285 22];

app.Label\_4.Text = 'Параметры моделирования разностным методом';

% Create Label\_dt

app.Label\_dt = uilabel(app.UIFigure);

app.Label\_dt.Tag = 'Label\_dt';

app.Label\_dt.Position = [21 359 131 22];

app.Label\_dt.Text = 'Шаг моделирования, с';

% Create Label\_x0

app.Label\_x0 = uilabel(app.UIFigure);

app.Label\_x0.Tag = 'Label\_x0';

app.Label\_x0.Position = [21 329 146 22];

app.Label\_x0.Text = 'Начальное положение, м';

% Create Label\_V0

app.Label\_V0 = uilabel(app.UIFigure);

app.Label\_V0.Tag = 'Label\_V0';

app.Label\_V0.Position = [21 299 143 22];

app.Label\_V0.Text = 'Начальная скорость, м/с';

% Create Label\_P0

app.Label\_P0 = uilabel(app.UIFigure);

app.Label\_P0.Tag = 'Label\_P0';

app.Label\_P0.Position = [21 269 143 22];

app.Label\_P0.Text = 'Начальная мощность, %';

% Create EditField\_dt

app.EditField\_dt = uieditfield(app.UIFigure, 'numeric');

app.EditField\_dt.ValueChangedFcn = createCallbackFcn(app, @dtValueChanged, true);

app.EditField\_dt.Tag = 'EditField\_dt';

app.EditField\_dt.Position = [201 359 50 22];

% Create EditField\_x0

app.EditField\_x0 = uieditfield(app.UIFigure, 'numeric');

app.EditField\_x0.ValueChangedFcn = createCallbackFcn(app, @x0ValueChanged, true);

app.EditField\_x0.Tag = 'EditField\_x0';

app.EditField\_x0.Position = [201 329 50 22];

% Create EditField\_V0

app.EditField\_V0 = uieditfield(app.UIFigure, 'numeric');

app.EditField\_V0.ValueChangedFcn = createCallbackFcn(app, @V0ValueChanged, true);

app.EditField\_V0.Tag = 'EditField\_V0';

app.EditField\_V0.Position = [201 299 50 22];

% Create EditField\_P0

app.EditField\_P0 = uieditfield(app.UIFigure, 'numeric');

app.EditField\_P0.ValueChangedFcn = createCallbackFcn(app, @P0ValueChanged, true);

app.EditField\_P0.Tag = 'EditField\_P0';

app.EditField\_P0.Position = [201 269 50 22];

% Create Label\_5

app.Label\_5 = uilabel(app.UIFigure);

app.Label\_5.Position = [21 239 55 22];

app.Label\_5.Text = 'Тяга, %';

% Create Spinner

app.Spinner = uispinner(app.UIFigure);

app.Spinner.Step = 10;

app.Spinner.Limits = [0 100];

app.Spinner.ValueChangedFcn = createCallbackFcn(app, @PmaxValueChanged, true);

app.Spinner.Tag = 'Spinner';

app.Spinner.Position = [201 239 70 22];

% Create ButtonGroup\_2

app.ButtonGroup\_2 = uibuttongroup(app.UIFigure);

app.ButtonGroup\_2.SelectionChangedFcn = createCallbackFcn(app, @mode, true);

app.ButtonGroup\_2.TitlePosition = 'centertop';

app.ButtonGroup\_2.Title = 'Режим работы';

app.ButtonGroup\_2.Tag = 'ButtonGroup\_2';

app.ButtonGroup\_2.Position = [21 61 150 160];

% Create Button\_Racing

app.Button\_Racing = uiradiobutton(app.ButtonGroup\_2);

app.Button\_Racing.Tag = 'Button\_Racing';

app.Button\_Racing.Text = 'Разгон';

app.Button\_Racing.Position = [11 78 59 22];

% Create Button\_Braking

app.Button\_Braking = uiradiobutton(app.ButtonGroup\_2);

app.Button\_Braking.Tag = 'Button\_Braking';

app.Button\_Braking.Text = 'Торможение';

app.Button\_Braking.Position = [11 48 91 22];

% Create Button\_Reg

app.Button\_Reg = uiradiobutton(app.ButtonGroup\_2);

app.Button\_Reg.Tag = 'Button\_Reg';

app.Button\_Reg.Text = 'С регулятором';

app.Button\_Reg.Position = [11 18 104 22];

% Create Button\_Full

app.Button\_Full = uiradiobutton(app.ButtonGroup\_2);

app.Button\_Full.Tag = 'Button\_Full';

app.Button\_Full.Text = 'Разгон+Торможение';

app.Button\_Full.Position = [11 108 135 22];

app.Button\_Full.Value = true;

% Create Label\_t

app.Label\_t = uilabel(app.UIFigure);

app.Label\_t.Tag = 'Label\_t';

app.Label\_t.HorizontalAlignment = 'right';

app.Label\_t.Position = [191 169 145 22];

app.Label\_t.Text = 'Время моделирования, с';

% Create Label\_Vz

app.Label\_Vz = uilabel(app.UIFigure);

app.Label\_Vz.Tag = 'Label\_Vz';

app.Label\_Vz.HorizontalAlignment = 'right';

app.Label\_Vz.Position = [191 139 136 22];

app.Label\_Vz.Text = 'Заданная скорость, м/с';

% Create EditField\_t

app.EditField\_t = uieditfield(app.UIFigure, 'numeric');

app.EditField\_t.ValueChangedFcn = createCallbackFcn(app, @tValueChanged, true);

app.EditField\_t.Tag = 'EditField\_t';

app.EditField\_t.Position = [351 169 50 22];

% Create EditField\_Vz

app.EditField\_Vz = uieditfield(app.UIFigure, 'numeric');

app.EditField\_Vz.ValueChangedFcn = createCallbackFcn(app, @VzValueChanged, true);

app.EditField\_Vz.Tag = 'EditField\_Vz';

app.EditField\_Vz.Position = [351 139 50 22];

% Create StartButton

app.StartButton = uibutton(app.UIFigure, 'push');

app.StartButton.ButtonPushedFcn = createCallbackFcn(app, @Start, true);

app.StartButton.Tag = 'StartButton';

app.StartButton.Position = [21 11 120 40];

app.StartButton.Text = 'Start';

% Create Label\_kp

app.Label\_kp = uilabel(app.UIFigure);

app.Label\_kp.HorizontalAlignment = 'right';

app.Label\_kp.Position = [191 109 25 22];

app.Label\_kp.Text = 'Kp';

% Create Label\_kd

app.Label\_kd = uilabel(app.UIFigure);

app.Label\_kd.HorizontalAlignment = 'right';

app.Label\_kd.Position = [191 79 25 22];

app.Label\_kd.Text = 'Kd';

% Create EditField\_kp

app.EditField\_kp = uieditfield(app.UIFigure, 'numeric');

app.EditField\_kp.ValueChangedFcn = createCallbackFcn(app, @EditField\_kpValueChanged, true);

app.EditField\_kp.Position = [231 109 50 22];

% Create EditField\_kd

app.EditField\_kd = uieditfield(app.UIFigure, 'numeric');

app.EditField\_kd.ValueChangedFcn = createCallbackFcn(app, @EditField\_kdValueChanged, true);

app.EditField\_kd.Position = [231 79 50 22];

% Show the figure after all components are created

app.UIFigure.Visible = 'on';

end

end

% App creation and deletion

methods (Access = public)

% Construct app

function app = app(varargin)

% Create UIFigure and components

createComponents(app)

% Register the app with App Designer

registerApp(app, app.UIFigure)

% Execute the startup function

runStartupFcn(app, @(app)startupFcn(app, varargin{:}))

if nargout == 0

clear app

end

end

% Code that executes before app deletion

function delete(app)

% Delete UIFigure when app is deleted

delete(app.UIFigure)

end

end

end