**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра КСУ**

Курсовая РАБОТА

**по дисциплине «ММОиСУ»**

Тема: Математическое моделирование систем управления

Вариант 3-5-24

|  |  |  |
| --- | --- | --- |
| Студент гр. 7408 |  | Ким А.А. |
| Преподаватель |  | Шпекторов А.Г. |

Санкт-Петербург

2021

**ЗАДАНИЕ**

**на курсовую работу**

|  |  |  |
| --- | --- | --- |
| Студент Ким А.А. | | |
| Группа 7408 | | |
| Тема работы: Математическое моделирование систем управления | | |
| Исходные данные:   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | № | Наименование | W, т | N, л.с. | v, уз | vk, уз | v1max, уз | v2max, уз | | 3 | «Парижская Коммуна» | 30395 | 61000 | 21.5 | - | - | - | | 5 | Малый рабочий катер | 4.6 | 50 | 9.0 | - | - | - | | 24 | КПК | 23.0 | 800 | - | 9.0 | 10.8 | 26 | | | |
| Содержание пояснительной записки:  Исходные данные, математическая модель движения корабля, программная модель, сценарий программы с графическим интерфейсом, рабочие экраны интерфейса программы моделирования, результаты исследования программной модели (таблица, графики), основные параметры движения корабля при разгоне и торможении. | | |
| Предполагаемый объем пояснительной записки:  Не менее 30 страниц. | | |
| Дата выдачи задания: 28.10.2021 | | |
| Дата сдачи реферата: 7.12.2021 | | |
| Дата защиты реферата: \_\_.\_\_.2021 | | |
| Студент |  | Ким А.А. |
| Преподаватель |  | Шпекторов А.Г. |

**Аннотация**

В данной курсовой работе были рассмотрены модели надводного водоизмещающего судна большого и малого водоизмещения, а также корабля на подводных крыльях. Моделирование движения судов осуществлялось разностным методом. В результате выполнения курсовой работы были разработаны математические модели судов, получены динамические показатели судов, осуществлен режим управления скоростью судна, а также разработан графический интерфейс.

**Summary**

In this course work, models of surface displacement vessels of large and small displacement, as well as hydrofoils were considered. Modeling of the movement of ships was carried out by the difference method. As a result of the course work, mathematical models of ships were developed, dynamic indicators of ships were obtained, a ship speed control mode was implemented, and a graphical interface was developed.

**содержание**

[**ВВЕДЕНИЕ** 5](#_Toc89738824)

[**1.** **Формирование математических моделей объектов управления** 6](#_Toc89738825)

[**1.1.** **Математическая модель движения надводного водоизмещающего корабля** 6](#_Toc89738826)

[**1.2.** **Математическая модель движения корабля на подводных крыльях** 9](#_Toc89738827)

[**1.3.** **Уточнение математических моделей** 10](#_Toc89738828)

[**2.** **ПРОГРАММНАЯ МОДЕЛЬ ДВИЖЕНИЯ КОРАБЛЯ** 12](#_Toc89738829)

[**2.1.** **Моделирование движения надводного водоизмещающего судна «Парижская Коммуна»** 14](#_Toc89738830)

[**2.2.** **Моделирование движения малого рабочего катера** 18](#_Toc89738831)

[**2.3.** **Моделирование движения корабля на подводных крыльях** 20](#_Toc89738832)

[**3.** **РЕЖИМ УПРАВЛЕНИЯ СКОРОСТЬЮ ХОДА** 25](#_Toc89738833)

[**4.** **ПРИЛОЖЕНИЕ С ГРАФИЧЕСКИМ ИНТЕРФЕЙСОМ** 31](#_Toc89738834)

[**ЗАКЛЮЧЕНИЕ** 39](#_Toc89738835)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 40](#_Toc89738836)

[**ПРИЛОЖЕНИЕ А** 41](#_Toc89738837)

# **ВВЕДЕНИЕ**

Целью курсовой работы является выработка студентов умения и практических навыков составления и написания программных моделей для исследования в среде MATLAB, создания интерфейсов для вывода результатов моделирования с использованием элементов MATLAB GUI.

Курсовая работа включает в себя следующие основные этапы:

1. Постановка задачи и формирование ее математической модели.
2. Разработка блок-схемы алгоритма решения задачи.
3. Написание и отладка программной модели движения.
4. Представление и анализ результатов работы.

# **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.

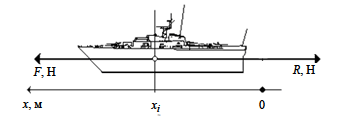
Правая часть представляет собой алгебраическую (с учетом направления, знака) сумму сил, действующих на корабль (Рис. 1.1.). Для НВК будем рассматривать следующие силы: F – движущая сила, или сила тяги винта (движителя), Н; R – сила сопротивления движению, Н.

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.

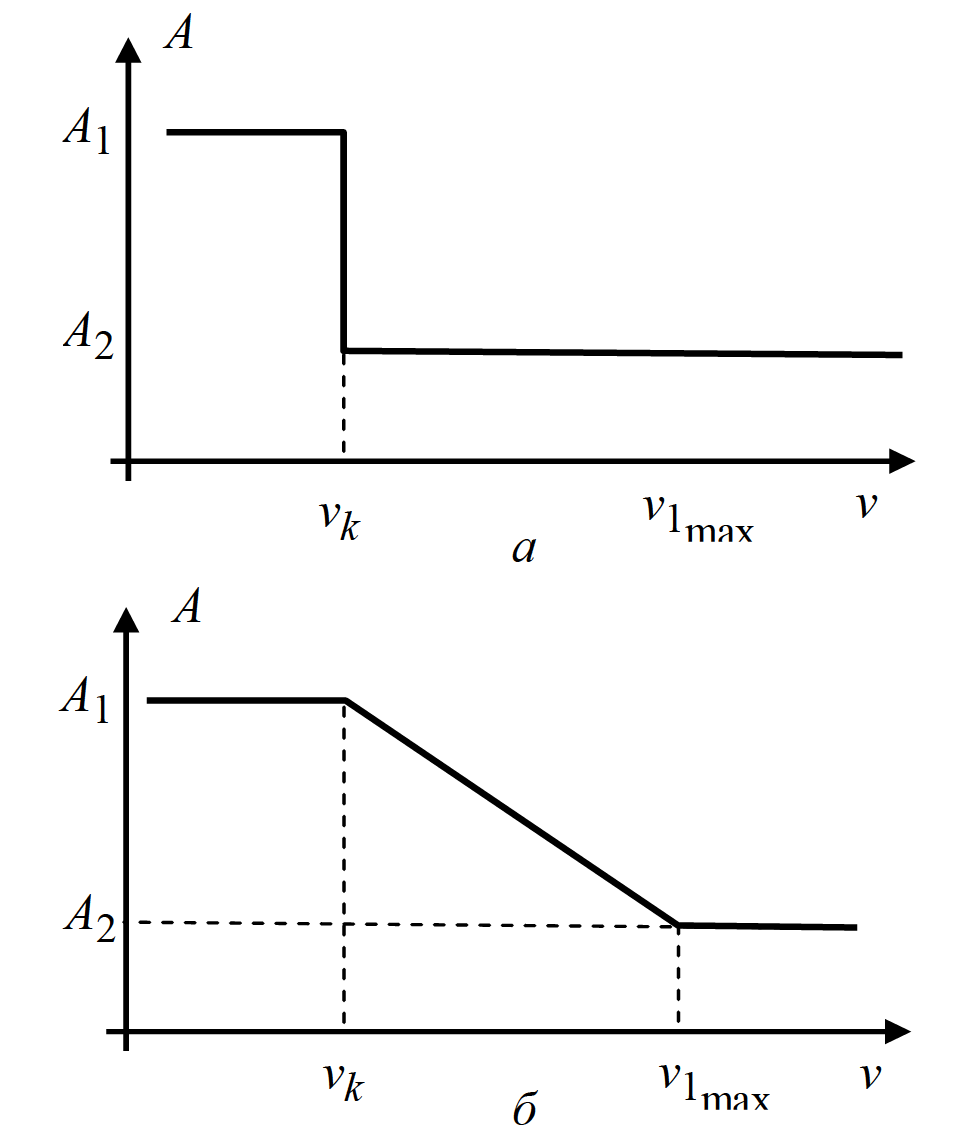
Таким образом, математическая модель КПК имеет вид



# **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**

Компьютерная программа, реализующая полученную математическую модель, обычно носит название программной модели. Создание программной модели является следующим этапом моделирования на ПК.

Современные программные средства позволяют моделировать динамические системы в непрерывном времени. Программная среда MATLAB имеет достаточно мощный аппарат для численного интегрирования и аналитического решения дифференциальных уравнений. Одной из многих функций численного интегрирования является функция ode45, имеющая следующий базовый формат записи:

[TOUT, YOUT]=ode45(ODEFUN,TSPAN,Y0).

Здесь ODEFUN – имя функции, вычисляющей правую часть дифференциального уравнения; TSPAN – вектор, определяющий начальное и конечное значения временного интервала, на котором осуществляется интегрирование; Y0 – вектор начальных условий; TOUT – вектор моментов времени, для которого получены матрицы частного решения YOUT.

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.

# **Моделирование движения надводного водоизмещающего судна «Парижская Коммуна»**

В ходе выполнения курсовой работы будет использоваться разностный метод моделирования. Исходные данные судна «Парижская Коммуна» приведены в таблице 1.

Таблица 1. Исходные данные НВК «Парижская Коммуна»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Наименование | W, т | N, л.с. | v, уз |
| 3 | «Парижская Коммуна» | 30395 | 61000 | 21.5 |

Так как в таблице 1 исходные данные даны размерности, отличной от СИ, то целесообразно осуществить переход к системе СИ:



Судно «Парижская Коммуна» имеет водоизмещение свыше 10000т, следовательно, примем максимально допустимую скорость изменения силы тяги , а шаг интегрирования примем  Допустимая погрешность при моделировании принимается равное 0,007.

Пример расчета:

1. Масса судна: 
2. Максимальная сила тяги движителя: 
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: .

Листинг программы, реализующий моделирование движения надводного водоизмещающего корабля представлен ниже.

%НВК Парижская Коммуна

W0 = 30395;% тонн

N0 = 61000;% лс

v0 = 21.5; %уз

%перевод в си

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 = 0.1\*Fmax;

dt = 10;

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

dP = 10;

if (dP >= dPmax)

dP = dPmax;

end

Pmax = 100; % максимальная тяга

i = 1;

t = [0];

v = [0];

P = [0];

x = [0];

dx = 0;

eps = 0.007;% допустимая погрешность, т.к. идеально не попадем

%разгон

while ( (vmax-v(i)) > eps)

tt = t(i)+dt;

t = [t tt];

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

x = [x xx];

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

v = [v vv];

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))<eps/10 && i > 3) %учитывает, что мощность может быть не 100%

break

end

end

% торможение

while ( v(i) > eps)

tt = t(i)+dt;

t = [t tt];

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

x = [x xx];

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

v = [v vv];

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\_Paris = [t' P' x' v'];% для таблицы

figure(1)

subplot(3,1,1)

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

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

ylabel('x, м','FontSize',12,'FontWeight','bold');

subplot(3,1,2)

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

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

ylabel('v, м/с','FontSize',12,'FontWeight','bold');

subplot(3,1,3)

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

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

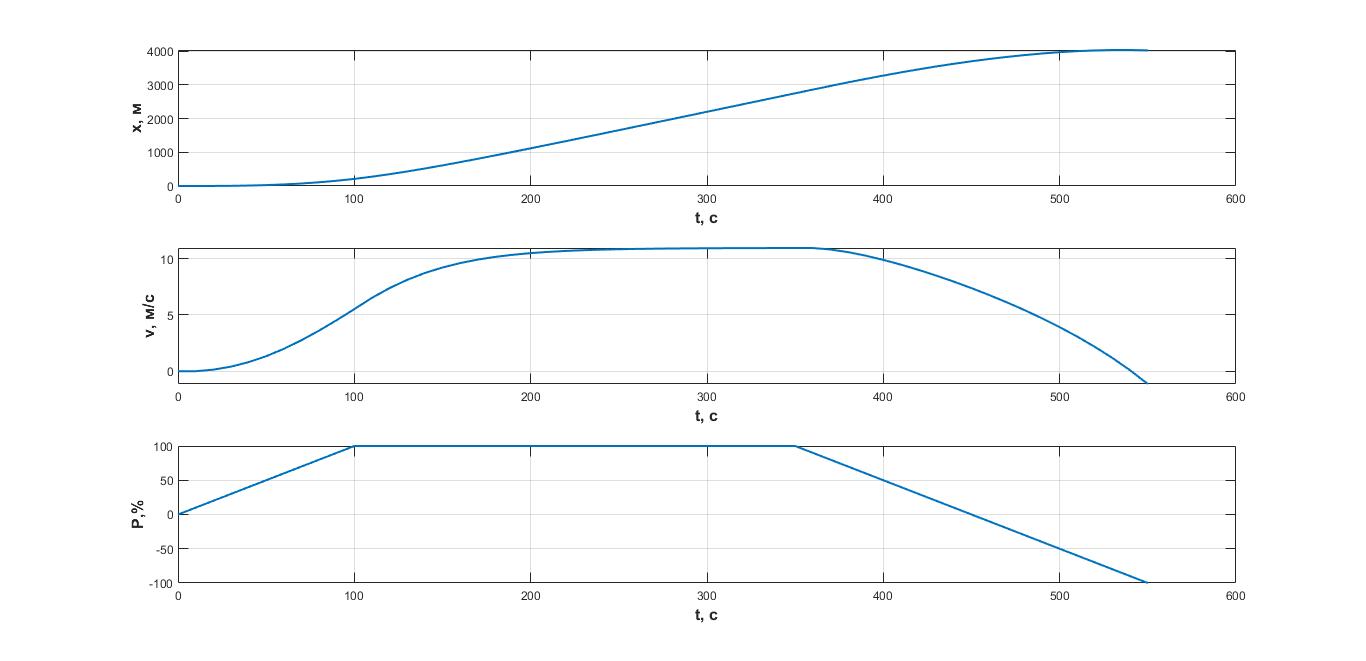
ylabel('P,%','FontSize',12,'FontWeight','bold');

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

Таблица 2. Результаты моделирования НВК «Парижская Коммуна»

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 | 0 | Торможение | 360 | 90 | 2857 | 10.959 |
| 10 | 10 | 0 | 0 | 370 | 80 | 2965 | 10.826 |
| 20 | 20 | 1 | 0.134 | 380 | 70 | 3071 | 10.591 |
| 30 | 30 | 5 | 0.403 | 390 | 60 | 31745 | 10.277 |
| 40 | 40 | 13 | 0.805 | 400 | 50 | 3273 | 9.902 |
| 50 | 50 | 26 | 1.336 | 410 | 40 | 3367 | 9.477 |
| 60 | 60 | 46 | 1.989 | 420 | 30 | 3458 | 9.010 |
| 70 | 70 | 74 | 2.753 | 430 | 20 | 3543 | 8.505 |
| 80 | 80 | 110 | 3.610 | 440 | 10 | 3622 | 7.964 |
| 90 | 90 | 155 | 4.541 | 450 | 0 | 3696 | 7.388 |
| 100 | 100 | 210 | 5.522 | 460 | -10 | 3764 | 6.777 |
| 110 | 100 | 276 | 6.527 | 470 | -20 | 3825 | 6.128 |
| 120 | 100 | 350 | 7.396 | 480 | -30 | 3880 | 5.438 |
| 130 | 100 | 431 | 8.129 | 490 | -40 | 3927 | 4.703 |
| 140 | 100 | 518 | 8.736 | 500 | -50 | 3966 | 3.917 |
| 150 | 100 | 611 | 9.227 | 510 | -60 | 3997 | 3.072 |
| 160 | 100 | 707 | 9.620 | 520 | -70 | 4018 | 2.159 |
| 170 | 100 | 806 | 9.930 | 530 | -80 | 4030 | 1.164 |
| 180 | 100 | 908 | 10.172 | 540 | -90 | 4031 | 0.072 |
| 190 | 100 | 1011 | 10.360 | 550 | -100 | 4019 | -1.139 |
| 200 | 100 | 1117 | 10.504 |  |  |  |  |
| 210 | 100 | 1223 | 10.6152 |  |  |  |  |
| 220 | 100 | 1330 | 10.699 |  |  |  |  |
| 230 | 100 | 1437 | 10.7643 |  |  |  |  |
| 240 | 100 | 1545 | 10.812 |  |  |  |  |
| 250 | 100 | 1654 | 10.850 |  |  |  |  |
| 260 | 100 | 1763 | 10.878 |  |  |  |  |
| 270 | 100 | 1872 | 10.899 |  |  |  |  |
| 280 | 100 | 1981 | 10.915 |  |  |  |  |
| 290 | 100 | 2090 | 10.927 |  |  |  |  |
| 300 | 100 | 2200 | 10.936 |  |  |  |  |
| 310 | 100 | 2309 | 10.943 |  |  |  |  |
| 320 | 100 | 2418 | 10.948 |  |  |  |  |
| 330 | 100 | 2528 | 10.952 |  |  |  |  |
| 340 | 100 | 2638 | 10.955 |  |  |  |  |
| 350 | 100 | 2747 | 10.958 |  |  |  |  |

Основные зависимости движения НВК «Парижская Коммуна» представлены на рисунке 3.



1. Основные зависимости движение НВК «Парижская Коммуна»

Исходя из полученных результатов можно сделать следующие выводы о динамических характеристиках НВК «Парижская коммуна»:

* Время набора максимальной скорости: 10.958 м/с – 350 с; при этом судно проходит 2747 м;
* Общее время торможения: 2000 с на расстоянии 1272 м;
* Общее время движения судна составило 550 с;
* Общее пройденное расстояние: 4019 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 |
| 5 | Малый рабочий катер | 4.6 | 50 | 9.0 |

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 «Парижская Коммуна». Основным отличием в реализации программного кода состоит в задании параметров судна.

Фрагмент листинга программы представлен ниже.

%НВК Малый рабочий катер

W0 = 4.6;% тонн

N0 = 50;% лс

v0 = 9.0; %уз

%перевод в си

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;

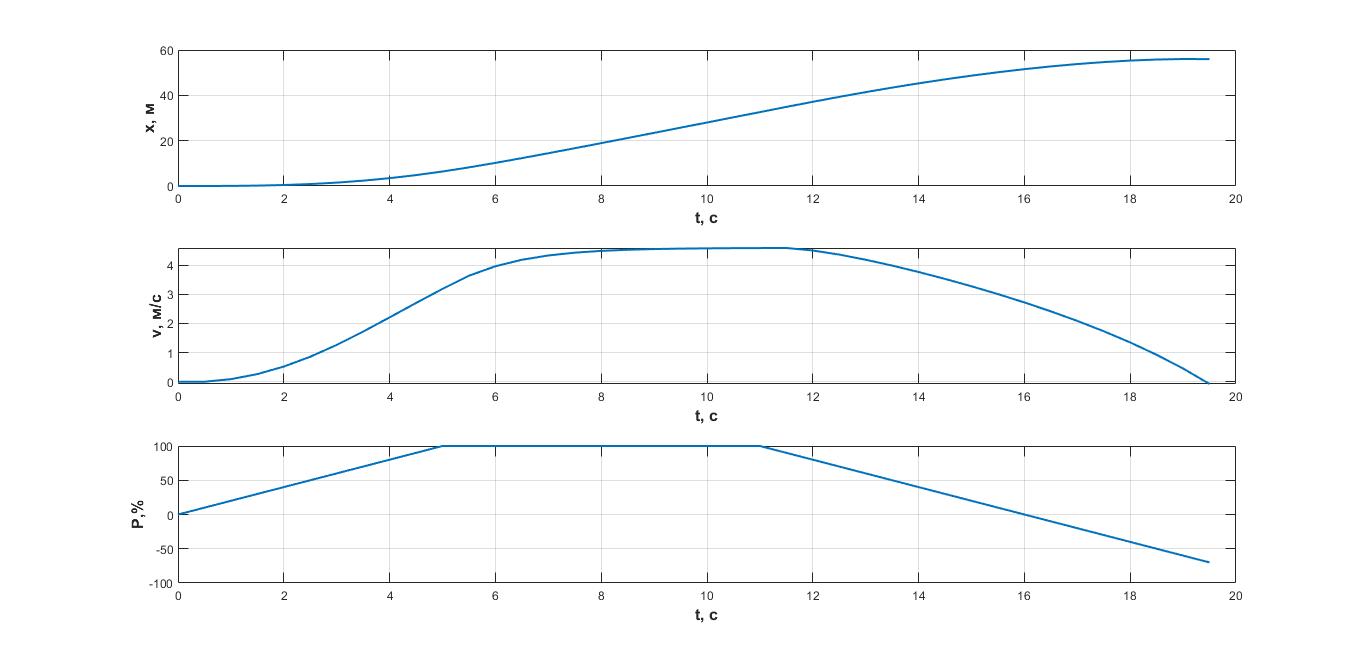
dt = 0.5;

Остальная часть кода идентична, что у НВК «Парижская Коммуна». 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 | 0 | 0 | 0 | Торможение | 11.5 | 90 | 34.89 | 4.586 |
| 0.5 | 10 | 0 | 0 | 12 | 80 | 37.14 | 4.500 |
| 1 | 20 | 0.04 | 0.087 | 12.5 | 70 | 39.32 | 4.360 |
| 1.5 | 30 | 0.17 | 0.260 | 13 | 60 | 41.41 | 4.183 |
| 2 | 40 | 0.43 | 0.519 | 13.5 | 50 | 43.40 | 3.982 |
| 2.5 | 50 | 0.8 | 0.856 | 14 | 40 | 45.28 | 3.762 |
| 3 | 60 | 1.49 | 1.261 | 14.5 | 30 | 47.04 | 3.525 |
| 3.5 | 70 | 2.35 | 1.718 | 15 | 20 | 48.68 | 3.273 |
| 4 | 80 | 3.45 | 2.205 | 15.5 | 10 | 50.18 | 3.004 |
| 4.5 | 90 | 4.80 | 2.701 | 16 | 0 | 51.54 | 2.718 |
| 5 | 100 | 6.39 | 3.183 | 16.5 | -10 | 52.75 | 2.412 |
| 5.5 | 100 | 8.21 | 3.635 | 17 | -20 | 53.79 | 2.085 |
| 6 | 100 | 10.19 | 3.960 | 17.5 | -30 | 54.66 | 1.731 |
| 6.5 | 100 | 12.28 | 4.182 | 18 | -40 | 55.33 | 1.346 |
| 7 | 100 | 14.45 | 4.330 | 18.5 | -50 | 55.79 | 0.9220 |
| 7.5 | 100 | 16.66 | 4.426 | 19 | -60 | 56.02 | 0.452 |
| 8 | 100 | 18.90 | 4.487 | 19.5 | -70 | 55.98 | -0.078 |
| 8.5 | 100 | 21.17 | 4.525 |  |  |  |  |
| 9 | 100 | 23.44 | 4.549 |  |  |  |  |
| 9.5 | 100 | 25.72 | 4.565 |  |  |  |  |
| 10 | 100 | 28.01 | 4.574 |  |  |  |  |
| 10.5 | 100 | 30.30 | 4.580 |  |  |  |  |
| 11 | 100 | 32.59 | 4.584 |  |  |  |  |

Основные зависимости движения малого рабочего катера представлены на рисунке 4.



1. Основные зависимости движение малого рабочего катера

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

* Время набора максимальной скорости: 4.584 м/с – 11 с; при этом судно проходит 32.59 м;
* Общее время торможения: 8.5 с на расстоянии 23.39м;
* Общее время движения судна составило 19.5 с;
* Общее пройденное расстояние: 55.98 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. | v, уз | vk, уз | v1max, уз | v2max, уз |
| 24 | КПК | 23.0 | 800 | - | 9.0 | 10.8 | 26 |

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 23 т, следовательно, примем максимально допустимую скорость изменения силы тяги , а шаг интегрирования примем  Погрешность расчетов также примем за 0,007.

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

clear; clc;

%КПК

W0 = 23.0;% тонн

N0 = 800;% лс

vk0 = 9.0; %уз

v1max0 = 10.8; %уз

v2max0 = 26; %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 = 0.2\*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;

eps = 0.007;% допустимая погрешность, т.к. идеально не попадем

%разгон

A1 = Fmax/v1max^2;

A2 = Fmax/v2max^2;

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

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

tt = t(i)+dt;

t = [t tt];

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

x = [x xx];

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

v = [v vv];

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))<eps/10 && i > 3) %учитывает, что мощность может быть не 100%

break

end

end

%торможение

while ( v(i) > eps)

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

tt = t(i)+dt;

t = [t tt];

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

x = [x xx];

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

v = [v vv];

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 = [t' P' x' v'];% для таблицы

figure(3)

subplot(3,1,1)

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

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

ylabel('x, м','FontSize',12,'FontWeight','bold');

subplot(3,1,2)

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

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

ylabel('v, м/с','FontSize',12,'FontWeight','bold');

subplot(3,1,3)

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

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

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

Результаты моделирования разностной модели КПК сведены в таблицу 6.

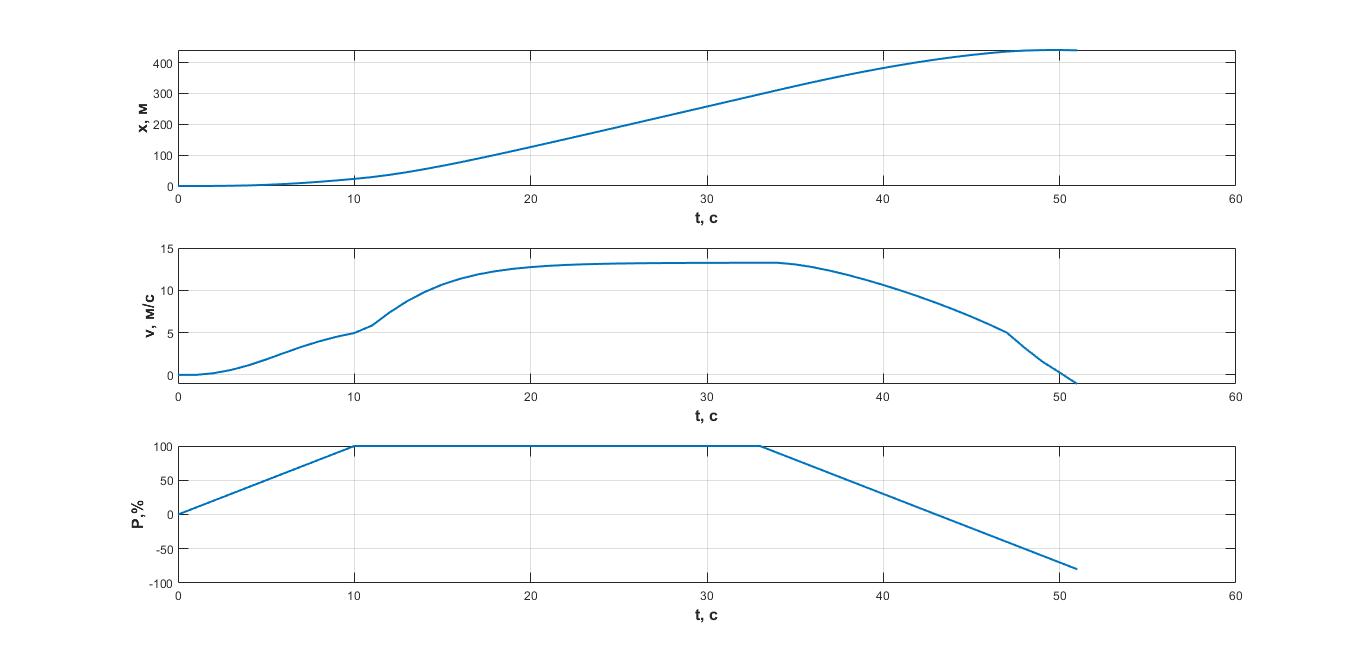
Таблица 6. Результаты моделирования КПК

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Режим | T, с | P, % | xi+1, м | vi+1, м/с | Режим | T, с | P, % | xi+1, м | vi+1, м/с |
| Разгон | 0 | 0 | 0 | 0 | Торможение | 34 | 90 | 310.6 | 13.25 |
| 1 | 10 | 0 | 0 | 35 | 80 | 323.6 | 13.06 |
| 2 | 20 | 0.1 | 0.19 | 36 | 70 | 336.4 | 12.73 |
| 3 | 30 | 0.7 | 0.57 | 37 | 60 | 348.7 | 12.30 |
| 4 | 40 | 1.9 | 1.13 | 38 | 50 | 360.5 | 11.80 |
| 5 | 50 | 3.7 | 1.82 | 39 | 40 | 371.7 | 11.23 |
| 6 | 60 | 6.3 | 2.57 | 40 | 30 | 382.3 | 10.62 |
| 7 | 70 | 9.6 | 3.31 | 41 | 20 | 392.3 | 9.96 |
| 8 | 80 | 13.5 | 3.96 | 42 | 10 | 401.6 | 9.26 |
| 9 | 90 | 18.0 | 4.50 | 43 | 0 | 410.1 | 8.51 |
| 10 | 100 | 23.0 | 4.95 | 44 | -10 | 417.8 | 7.71 |
| 11 | 100 | 28.8 | 5.83 | 45 | -20 | 424.7 | 6.87 |
| 12 | 100 | 36.2 | 7.38 | 46 | -30 | 430.6 | 5.96 |
| 13 | 100 | 44.9 | 8.71 | 47 | -40 | 435.6 | 4.99 |
| 14 | 100 | 54.7 | 9.81 | 48 | -50 | 438.9 | 3.22 |
| 15 | 100 | 65.4 | 10.68 | 49 | -60 | 440.4 | 1.59 |
| 16 | 100 | 76.8 | 11.36 | 50 | -70 | 440.7 | 0.27 |
| 17 | 100 | 88.7 | 11.87 | 51 | -80 | 439.6 | -1.07 |
| 18 | 100 | 100.9 | 12.25 |  |  |  |  |
| 19 | 100 | 113.5 | 12.53 |  |  |  |  |
| 20 | 100 | 126.2 | 12.74 |  |  |  |  |
| 21 | 100 | 139.1 | 12.88 |  |  |  |  |
| 22 | 100 | 152.1 | 12.99 |  |  |  |  |
| 23 | 100 | 165.2 | 13.07 |  |  |  |  |
| 24 | 100 | 178.3 | 13.12 |  |  |  |  |
| 25 | 100 | 191.5 | 13.16 |  |  |  |  |
| 26 | 100 | 204.6 | 13.19 |  |  |  |  |
| 27 | 100 | 217.9 | 13.21 |  |  |  |  |

Таблица 6. Продолжение

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Режим | T, с | P, % | xi+1, м | vi+1, м/с | Режим | T, с | P, % | xi+1, м | vi+1, м/с |
| Разгон | 28 | 100 | 231.1 | 13.22 | Торможение |  |  |  |  |
| 29 | 100 | 244.3 | 13.23 |  |  |  |  |
| 30 | 100 | 257.6 | 13.24 |  |  |  |  |
| 31 | 100 | 270.8 | 13.24 |  |  |  |  |
| 32 | 100 | 284.1 | 13.25 |  |  |  |  |
| 33 | 100 | 297.3 | 13.254 |  |  |  |  |

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



1. Основные зависимости движения корабля на подводных крыльях

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

* Время набора максимальной скорости: 13.254 м/с – 33 с; при этом судно проходит 297.3 м;
  + В водоизмещающем режиме – 10 с;
  + В режиме глиссирования – 1 с;
  + На крыльях – 22 с;
* Общее время торможения: 18 с на расстоянии 142,3м;
* Общее время движения судна составило 51 с;
* Общее пройденное расстояние: 439,6 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.

Формирование мощности двигателя будет выглядеть следующим образом:



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

Листинг программы режима управления скорости хода НВК «Парижская Коммуна» представлен ниже. Заданная скорость – 8 м/с, коэффициенты регулятора kp = 5, kd = 150.

clear; clc;

%регулятор НВК Париж

W0 = 30395;% тонн

N0 = 61000;% лс

v0 = 21.5; %уз

%перевод в си

W = W0\*1000;%кг

N = N0\*735.5;% Вт

vmax = v0\*0.51;% м/с

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

dFmax = 0.1\*Fmax;

dt = 10;

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

dP = 0;

if (dP >= dPmax)

dP = dPmax;

end

Vz = 8;% задание м/с

kp = 5;% коэффициент ПД регулятора

kd = 150;% коэф ПД регулятора

A = Fmax/vmax^2;

i = 1;

t = [0];

v = [0];

P = [0];

x = [0];

dx = 0;

prev\_e = Vz-v(1);

while(t(i) < 600)

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

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

x = [x xx];

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

v = [v vv];

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,'FontWeight','bold');

ylabel('x, м','FontSize',12,'FontWeight','bold');

subplot(3,1,2)

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

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

ylabel('v, м/с','FontSize',12,'FontWeight','bold');

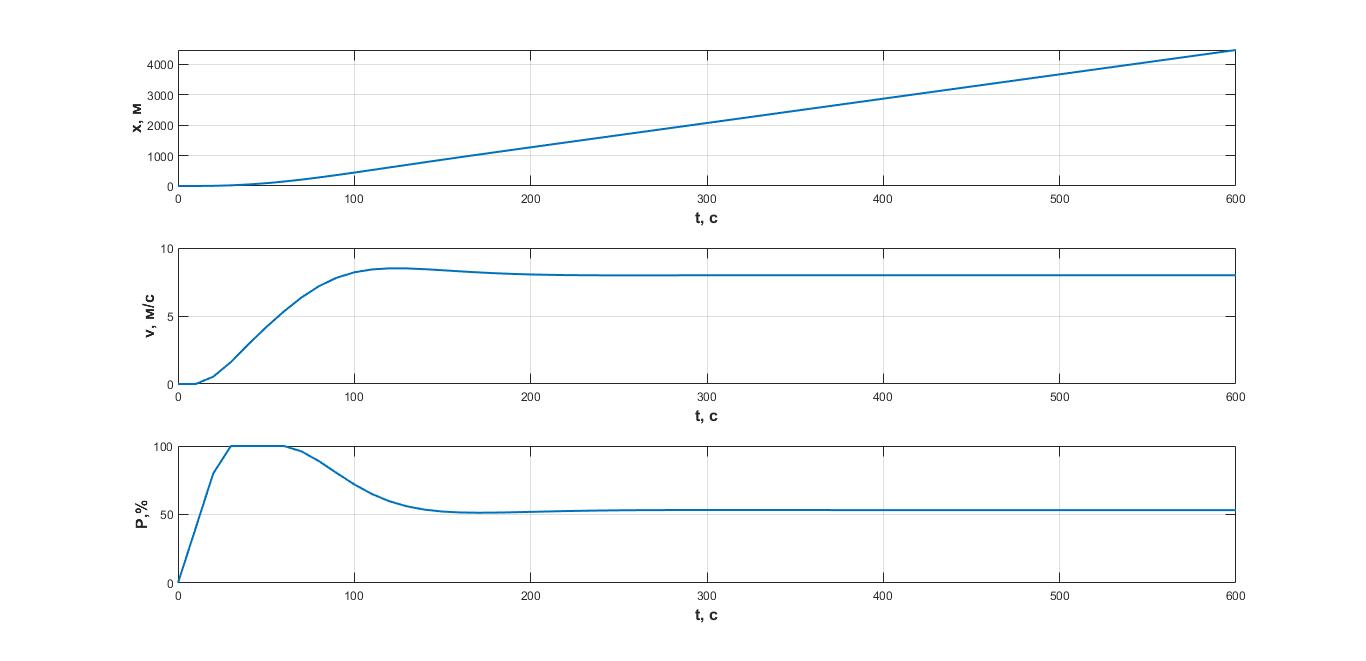
subplot(3,1,3)

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

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

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

Результат работы программы представлен на рисунке 6.



1. Режим управления скоростью НВК «Парижская Коммуна»

Листинг программы режима управления скорости хода малого рабочего катера мало отличается от НВК «Парижская Коммуна». Отличия только в параметрах судна. Заданная скорость – 3 м/с, коэффициенты регулятора kp = 7, kd = 25.

clear; clc;

%Регулятор НВК Малый рабочий катер

W0 = 4.6;% тонн

N0 = 50;% лс

v0 = 9.0; %уз

%перевод в си

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;

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

kp = 7;

kd = 25;

i = 1;

t = [0];

v = [0];

P = [0];

x = [0];

dx = 0;

Vz = 3;% м/с

prev\_e = Vz-v(1);

while(t(i) < 50)

tt = t(i)+dt;

t = [t tt];

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

x = [x xx];

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

v = [v vv];

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(2)

subplot(3,1,1)

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

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

ylabel('x, м','FontSize',12,'FontWeight','bold');

subplot(3,1,2)

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

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

ylabel('v, м/с','FontSize',12,'FontWeight','bold');

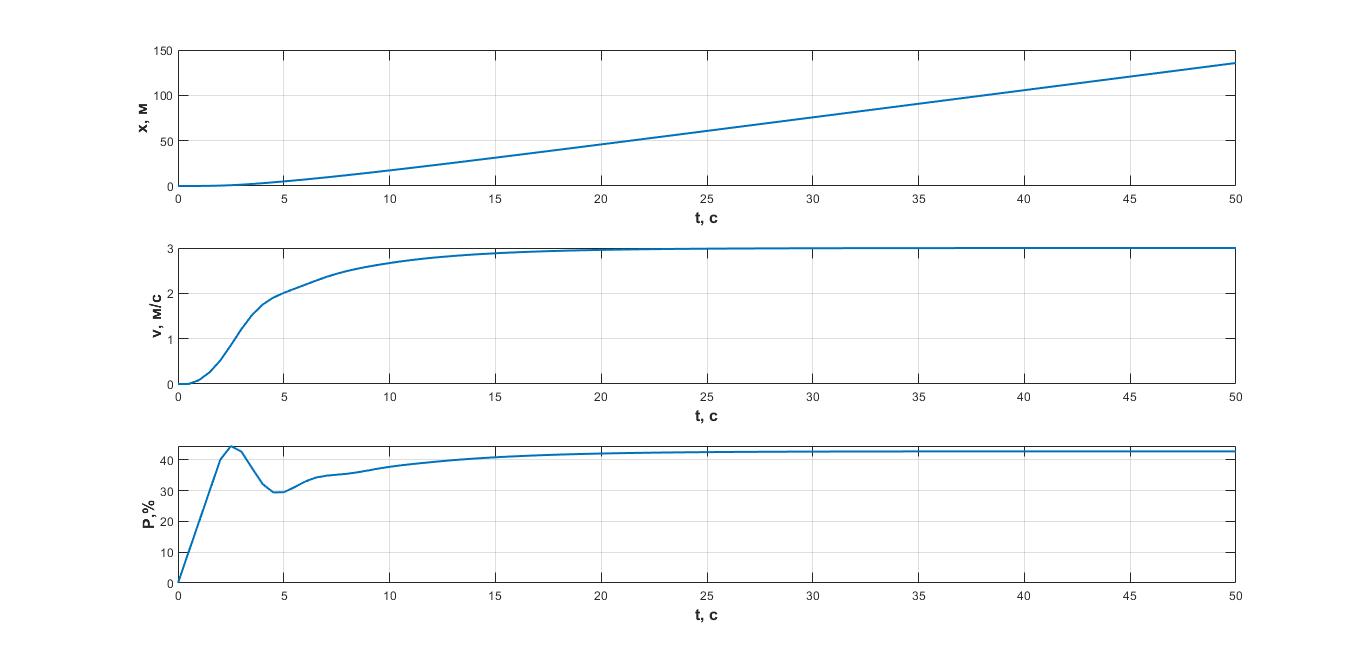
subplot(3,1,3)

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

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

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

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 – 7 м/с, коэффициенты регулятора kp = 15, kd = 7.

clear; clc;

W0 = 23.0;% тонн

N0 = 800;% лс

vk0 = 9.0; %уз

v1max0 = 10.8; %уз

v2max0 = 26; %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 = 0.2\*Fmax;

dt = 1;

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

dP = 0;

kp = 3;

kd = 15;

Vz = 7;% м/с

A1 = Fmax/v1max^2;

A2 = Fmax/v2max^2;

i = 1;

t = [0];

v = [0];

P = [0];

x = [0];

dx = 0;

prev\_e = Vz-v(1);

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

tt = t(i)+dt;

t = [t tt];

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

x = [x xx];

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

v = [v vv];

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,'FontWeight','bold');

ylabel('x, м','FontSize',12,'FontWeight','bold');

subplot(3,1,2)

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

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

ylabel('v, м/с','FontSize',12,'FontWeight','bold');

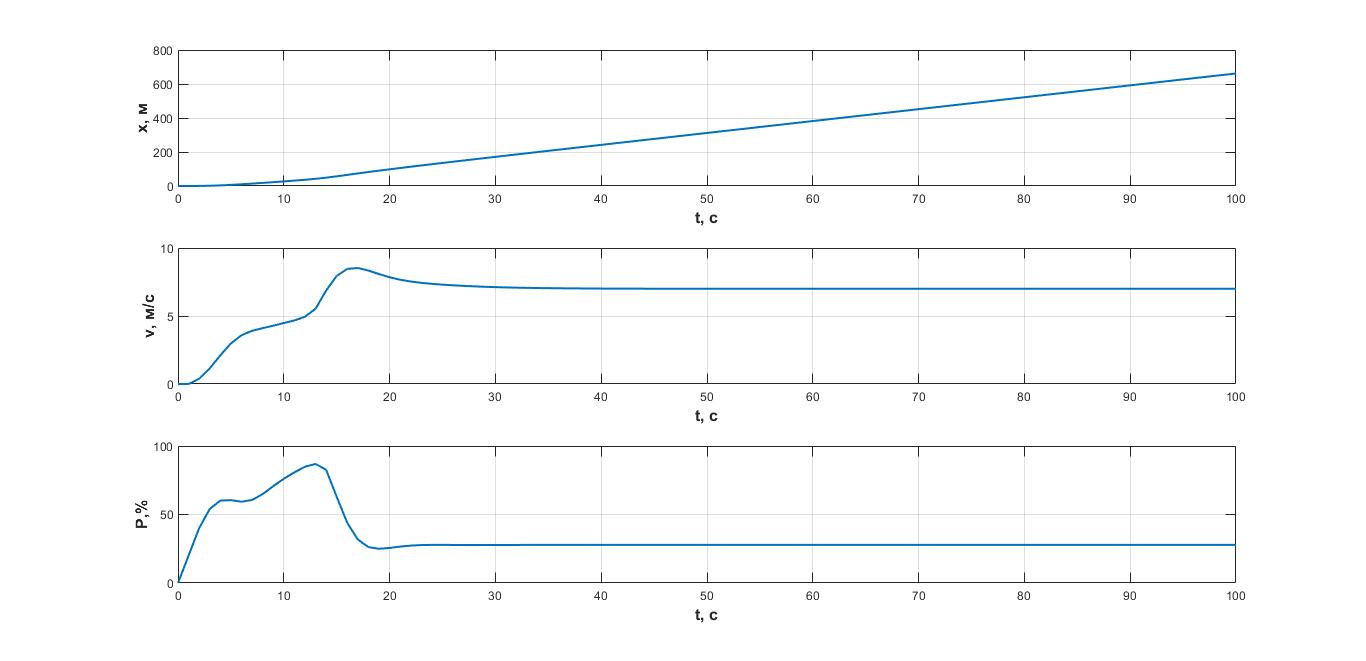
subplot(3,1,3)

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

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

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

Результат работы программы представлен на рисунке 8.



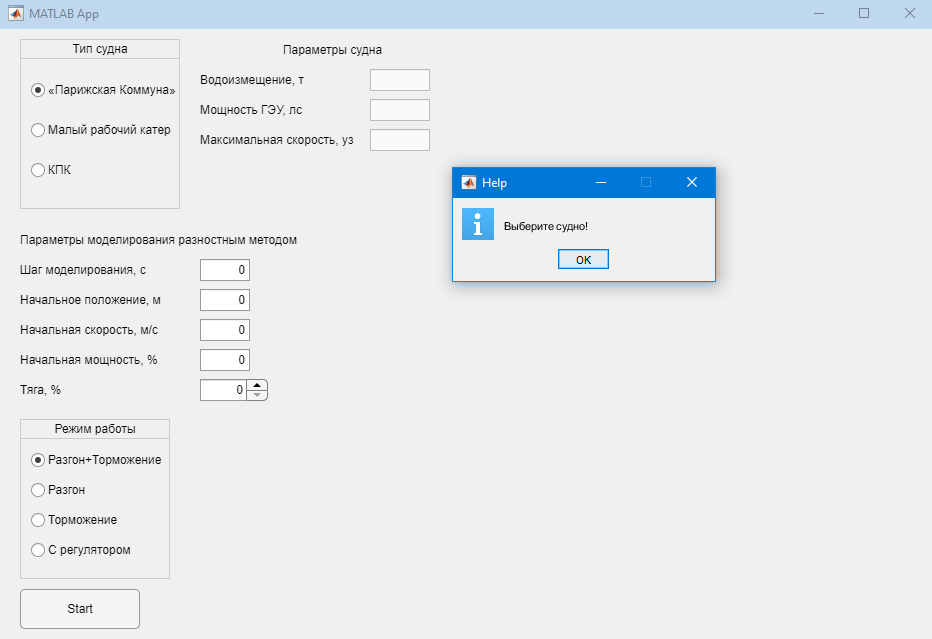
1. Режим управления скоростью корабля на подводных крыльях

# **ПРИЛОЖЕНИЕ С ГРАФИЧЕСКИМ ИНТЕРФЕЙСОМ**

Средства GUI представляют собой набор графических объектов, которые носят общее название элементов управления. К ним относятся различные кнопки, списки, поля для ввода текста и др. Разработка приложений с графическим интерфейсом состоит в том, чтобы создать набор таких элементов и запрограммировать события, которые происходят при работе с ними.

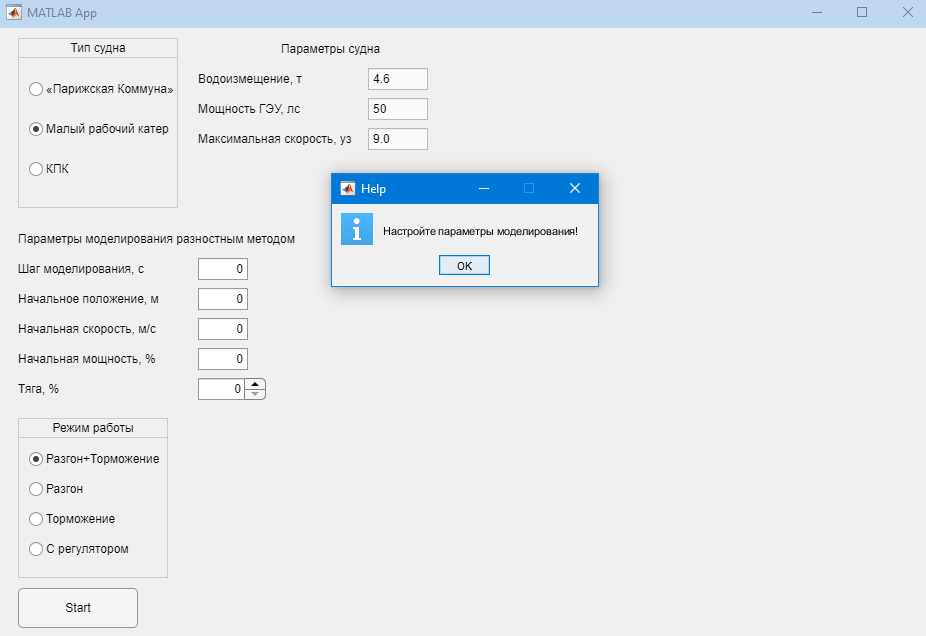
В состав среды MATLAB интегрирован пакет GUIDE, позволяющий удобно и быстро создавать окна с элементами управления. Есть возможности для создания приложений GUI вручную.

В данной работе было разработано приложение с графическим интерфейсом, задачей которой являлась моделирование динамики различных кораблей. Окно приложения продемонстрированно на рисунке 9.



1. Начальное окно приложения

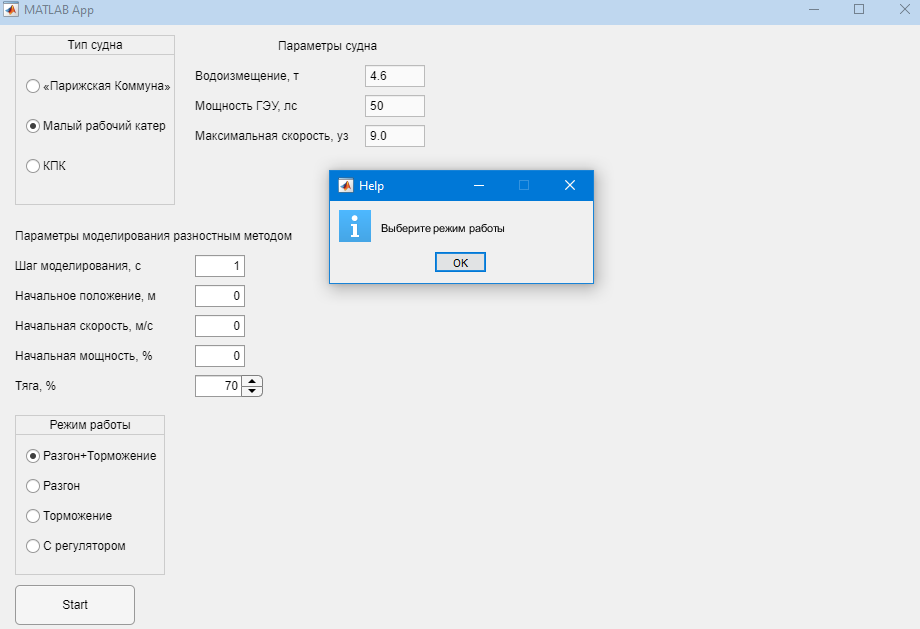
При запуске приложения открывается начальное окно и вспомогательное окно с инструкцией. При выборе судна автоматически заполняются параметры судна (рисунок 10), изменить их нельзя. Также всплывает окно с инструкцией к следующему шагу.



1. Выбор судна

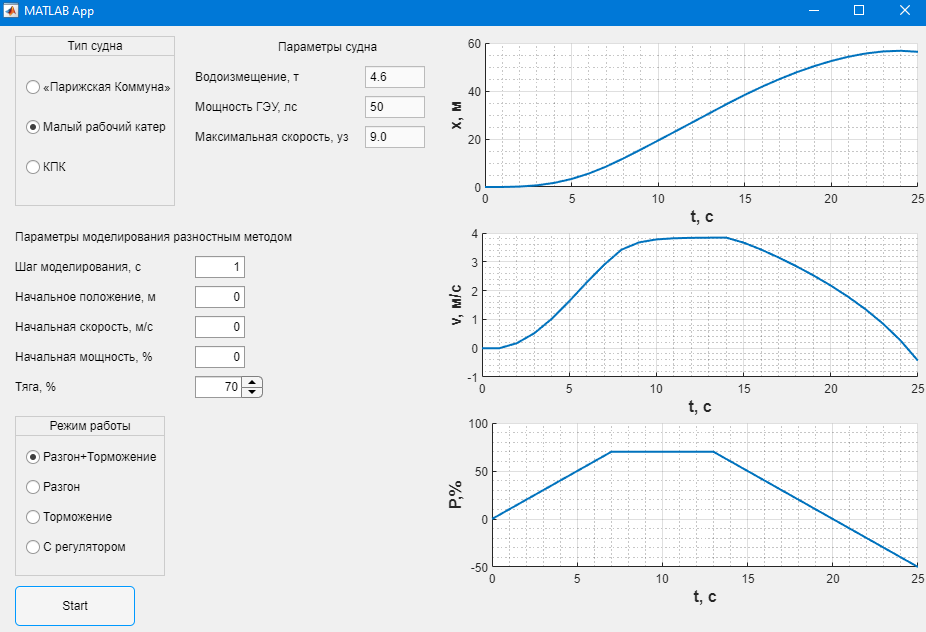
После выбора судна необходимо заполнить параметры моделирования. Моделирование осуществляется разностным методом. Также имеется возможность задавать максимальную тягу двигателя.

После заполнения параметров моделирования оператору нужно выбрать режим работы судна. В случае если оператор не выберет режим работы судна и нажмет кнопку «Start» приложение выдаст соответствующую ошибку (рисунок 11).



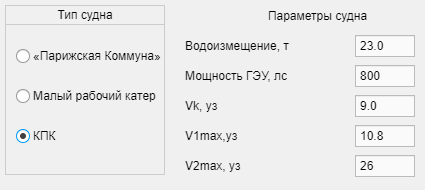
1. Запуск программы без выбора режима работы судна

После выбора режима работы судна и нажатия кнопки «Start» в правой области приложения появятся графики, описывающие динамику движения судна (рисунок 12).



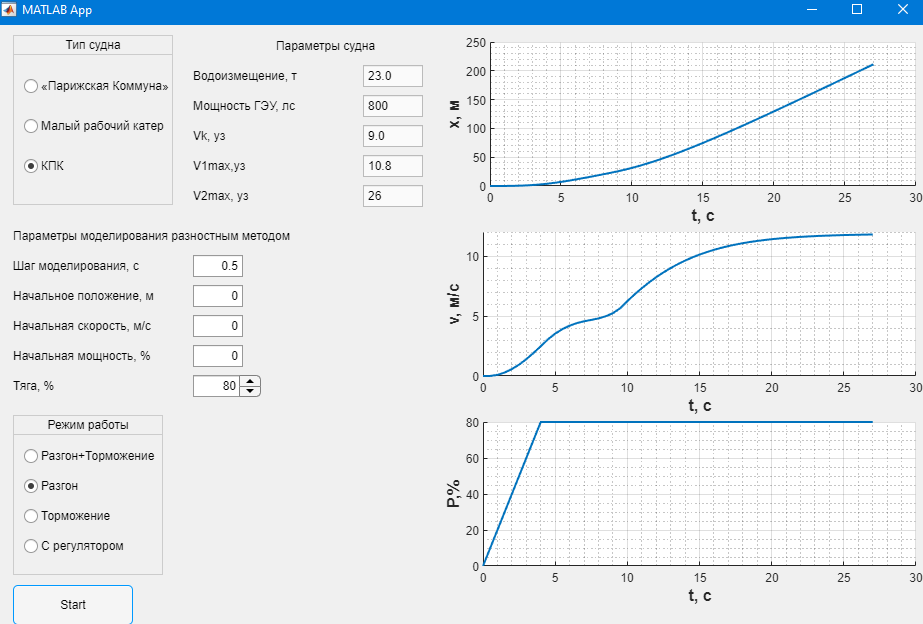
1. Результат работы приложения

Приложение осуществляет моделирование трех типов кораблей. При выборе кораблей «Парижская Коммуна» и малый рабочий катер параметры судна изменяются в соответствии с исходными данными. А при выборе судна КПК в параметрах судна появляются новые строки, которые также соответствуют исходным данным (рисунок 13).



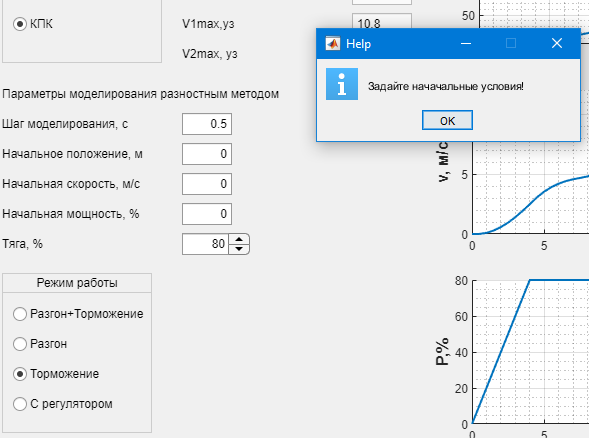
1. Выбор КПК

Работа программы в режиме разгона продемонстрирована на рисунке 14.

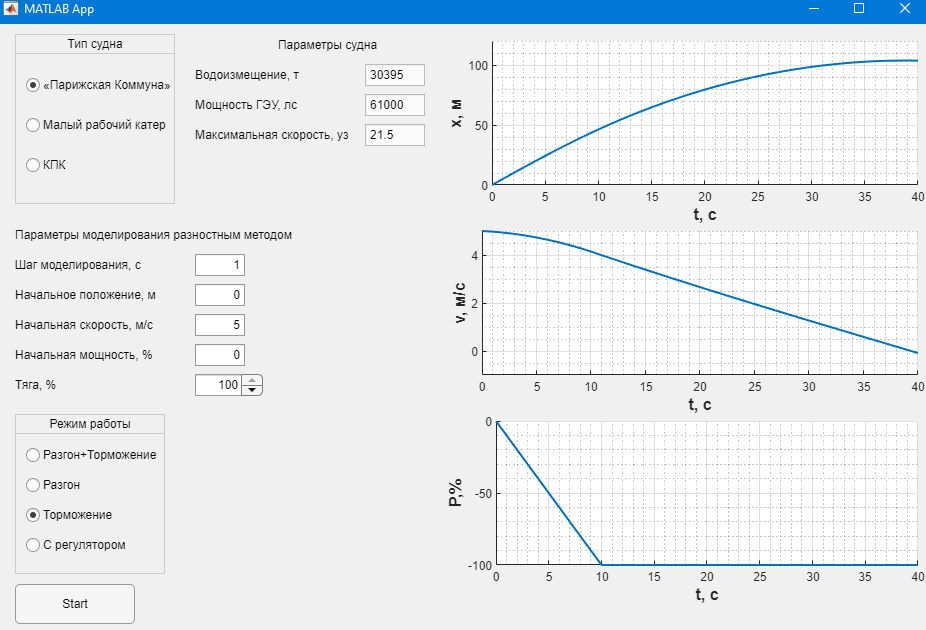


1. Режим разгона

При выборе режима торможения программа проинформирует, что нужно задать начальные условия (рисунок 15). Демонстрация работы приложения в режиме торможения представлена на рисунке 16.

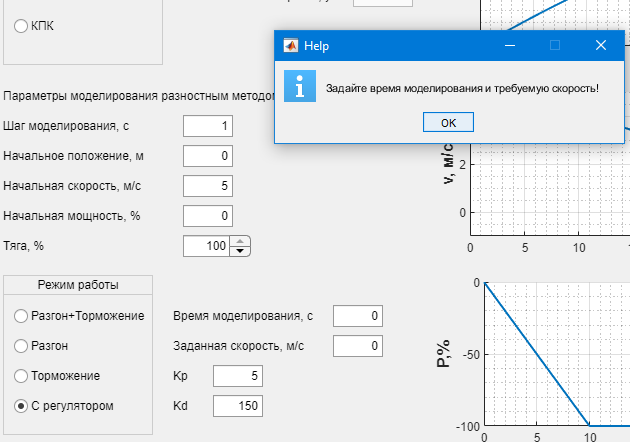


1. Предупреждение при работе в режиме торможения



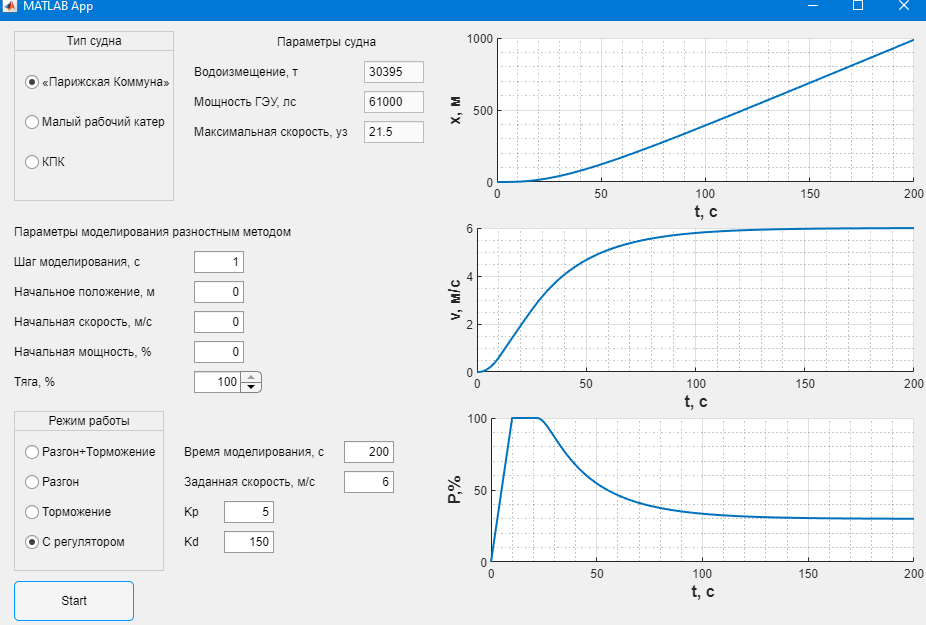
1. Режим работы торможение

В приложение имеется режим работы с регулятором, который обеспечивает поддержание заданной скорости. При выборе режима работы с регулятором появляются дополнительные параметры моделирования, представленные на рисунке 17.



1. Режим работы с регулятором

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. Результат работы приложения в режиме работы с регулятором продемонстрирован на рисунке 18.



1. Режим работы с регулятором

Сценарий приложения представлен в таблице 7.

Таблица 7. Сценарий приложения

|  |  |  |  |
| --- | --- | --- | --- |
| Событие | Идентификатор элемента управления | Действие | Примечание |
| Запуск приложения | startupFcn | Отключить видимость графиков, доп. параметров КПК;  Инициализировать переменные;  Открытие вспомогательного окна | Свойство Visible;  Функция helpdlg; |
| Выбор типа судна | Ship | Установка параметров судна в соответствии с исходным заданием;  При выборе КПК появляются дополнительные параметры;  Открытие вспомогательного окна | Свойство Visible;  Свойство Value;  Функция helpdlg; |
| Выбор параметров моделирования |  |  |  |
| Шаг интегрирования | dtValueChanged | Присвоение заданного значения | Свойство Value; |
| Начального значения координаты | x0ValueChanged | Присвоение заданного значения | Свойство Value; |
| Начальной скорости | V0ValueChanged | Присвоение заданного значения | Свойство Value; |
| Начальной мощности | P0ValueChanged | Присвоение заданного значения | Свойство Value; |

Таблица 7 - Продолжение

|  |  |  |  |
| --- | --- | --- | --- |
| Максимальной тяги | PmaxValueChanged | Присвоение заданного значения | Свойство Value; |
| Выбор режима работы | mode | Установка соответствующего флага.  В режиме работы с регулятором появляются дополнительные параметры. Параметры регулятора автоматически заполняются.  Вспомогательное окно. | Свойство Visible;  Свойство Value;  Функция helpdlg; |
| Задание требуемой скорости | VzValueChanged | Присвоение заданного значения | Свойство Value; |
| Задание времени моделирования | tValueChanged | Присвоение заданного значения | Свойство Value; |
| Запуск | Start | Включение отображения графиков;  Моделирование движения выбранного судна согласно режиму работы. | Свойство Visible;  Свойство Value; |

Листинг приложения представлен в приложении А.

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения курсовой работы 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.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Математическое моделирование системы управления: методич. указания к курсовому проектированию / сост.: О. Ю. Лукомская, А. Г. Шпекторов. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2014. 40 с.

# **ПРИЛОЖЕНИЕ А**

**Листинг приложения**

classdef app < matlab.apps.AppBase

% Properties that correspond to app components

properties (Access = public)

UIFigure matlab.ui.Figure

ButtonGroup matlab.ui.container.ButtonGroup

Button\_Paris matlab.ui.control.RadioButton

Button\_Kater matlab.ui.control.RadioButton

Button\_KPK matlab.ui.control.RadioButton

Label\_2 matlab.ui.control.Label

EditFieldW matlab.ui.control.EditField

Label\_3 matlab.ui.control.Label

EditFieldN matlab.ui.control.EditField

EditFieldV matlab.ui.control.EditField

Label matlab.ui.control.Label

EditFieldV2max matlab.ui.control.EditField

EditFieldV1max matlab.ui.control.EditField

LabelV1max matlab.ui.control.Label

LabelV2max matlab.ui.control.Label

LabelV matlab.ui.control.Label

Label\_4 matlab.ui.control.Label

Label\_dt matlab.ui.control.Label

Label\_x0 matlab.ui.control.Label

Label\_V0 matlab.ui.control.Label

Label\_P0 matlab.ui.control.Label

EditField\_dt matlab.ui.control.NumericEditField

EditField\_x0 matlab.ui.control.NumericEditField

EditField\_V0 matlab.ui.control.NumericEditField

EditField\_P0 matlab.ui.control.NumericEditField

Label\_5 matlab.ui.control.Label

Spinner matlab.ui.control.Spinner

ButtonGroup\_2 matlab.ui.container.ButtonGroup

Button\_Racing matlab.ui.control.RadioButton

Button\_Braking matlab.ui.control.RadioButton

Button\_Reg matlab.ui.control.RadioButton

Button\_Full matlab.ui.control.RadioButton

Label\_t matlab.ui.control.Label

Label\_Vz matlab.ui.control.Label

EditField\_t matlab.ui.control.NumericEditField

EditField\_Vz matlab.ui.control.NumericEditField

StartButton matlab.ui.control.Button

Label\_kp matlab.ui.control.Label

Label\_kd matlab.ui.control.Label

EditField\_kp matlab.ui.control.NumericEditField

EditField\_kd matlab.ui.control.NumericEditField

UIAxes\_x matlab.ui.control.UIAxes

UIAxes\_v matlab.ui.control.UIAxes

UIAxes\_p matlab.ui.control.UIAxes

end

properties (Access = private)

W % Водоизмещение, кг

N % Мощность, Вт

Vmax % Скорость, м/с

Vk % м/с

V1max

V2max

dt % шаг моделирования

x0 % нач. положение

V0 % нач. скорость

P0 % нач. мощность

Pmax % макс. тяга

Flag\_mode % Режим работы

Flag\_ship % Тип судна

t\_mod % время моделирования

Vz %заданная скорость

kp

kd

flag\_error

end

% Callbacks that handle component events

methods (Access = private)

% Code that executes after component creation

function startupFcn(app, a)

app.Button\_Paris.Value = false;

app.Button\_Kater.Value = false;

app.Button\_KPK.Value = false;

app.EditFieldV1max.Visible = 0;

app.EditFieldV2max.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 = 0;

helpdlg('Выберите судно!','Help');

end

% Selection changed function: ButtonGroup

function Ship(app, event)

if app.Button\_Paris.Value == true

app.EditFieldW.Value = '30395';

app.EditFieldN.Value = '61000';

app.EditFieldV.Value = '21.5';

app.EditFieldV1max.Visible = 0;

app.EditFieldV2max.Visible = 0;

app.LabelV1max.Visible=0;

app.LabelV2max.Visible=0;

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

%перевод в си

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

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

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

app.Flag\_ship = 1;

app.flag\_error = 1;

end

if app.Button\_Kater.Value == true

app.EditFieldW.Value = '4.6';

app.EditFieldN.Value = '50';

app.EditFieldV.Value = '9.0';

app.EditFieldV1max.Visible = 0;

app.EditFieldV2max.Visible = 0;

app.LabelV1max.Visible=0;

app.LabelV2max.Visible=0;

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

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

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

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

app.Flag\_ship = 2;

app.flag\_error = 1;

end

if app.Button\_KPK.Value == true

app.EditFieldW.Value = '23.0';

app.EditFieldN.Value = '800';

app.EditFieldV.Value = '9.0';

app.EditFieldV1max.Visible = 1;

app.EditFieldV2max.Visible = 1;

app.LabelV1max.Visible=1;

app.LabelV2max.Visible=1;

app.LabelV.Text='Vk, уз';

app.EditFieldV1max.Value = '10.8';

app.EditFieldV2max.Value = '26';

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

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

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

app.V1max = str2double(app.EditFieldV1max.Value)\*0.51;

app.V2max = str2double(app.EditFieldV2max.Value)\*0.51;

app.Flag\_ship = 3;

app.flag\_error = 1;

end

helpdlg('Настройте параметры моделирования!','Help');

end

% Value changed function: EditField\_dt

function dtValueChanged(app, event)

if app.flag\_error == 1

value = app.EditField\_dt.Value;

app.dt = value;

else

helpdlg('Сначала выберите судно!','Help');

end

end

% Value changed function: EditField\_x0

function x0ValueChanged(app, event)

if app.flag\_error == 1

value = app.EditField\_x0.Value;

app.x0 = value;

else

helpdlg('Сначала выберите судно!','Help');

end

end

% Value changed function: EditField\_V0

function V0ValueChanged(app, event)

if app.flag\_error == 1

value = app.EditField\_V0.Value;

app.V0 = value;

else

helpdlg('Сначала выберите судно!','Help');

end

end

% Value changed function: EditField\_P0

function P0ValueChanged(app, event)

if app.flag\_error == 1

value = app.EditField\_P0.Value;

app.P0 = value;

else

helpdlg('Сначала выберите судно!','Help');

end

end

% Value changed function: Spinner

function PmaxValueChanged(app, event)

if app.flag\_error == 1

value = app.Spinner.Value;

app.Pmax = value;

else

helpdlg('Сначала выберите судно!','Help');

end

end

% Selection changed function: ButtonGroup\_2

function mode(app, event)

if app.Button\_Racing.Value == 1

app.Flag\_mode = 1;

app.Label\_Vz.Visible = false;

app.Label\_t.Visible = false;

app.EditField\_Vz.Visible = false;

app.EditField\_t.Visible = false;

app.Label\_kd.Visible = false;

app.Label\_kp.Visible = false;

app.EditField\_kp.Visible = false;

app.EditField\_kd.Visible = false;

end

if app.Button\_Braking.Value == 1

app.Flag\_mode = 2;

app.Label\_Vz.Visible = false;

app.Label\_t.Visible = false;

app.EditField\_Vz.Visible = false;

app.EditField\_t.Visible = false;

app.Label\_kd.Visible = false;

app.Label\_kp.Visible = false;

app.EditField\_kp.Visible = false;

app.EditField\_kd.Visible = false;

helpdlg('Задайте начачальные условия!','Help');

end

if app.Button\_Reg.Value == 1

app.Flag\_mode = 3;

app.Label\_Vz.Visible = true;

app.Label\_t.Visible = true;

app.EditField\_Vz.Visible = true;

app.EditField\_t.Visible = true;

app.Label\_kd.Visible = true;

app.Label\_kp.Visible = true;

app.EditField\_kp.Visible = true;

app.EditField\_kd.Visible = true;

if (app.Flag\_ship == 1)

app.EditField\_kp.Value = 5;

app.EditField\_kd.Value = 150;

end

if (app.Flag\_ship == 2)

app.EditField\_kp.Value = 7;

app.EditField\_kd.Value = 25;

end

if (app.Flag\_ship == 3)

app.EditField\_kp.Value = 3;

app.EditField\_kd.Value = 15;

end

helpdlg('Задайте время моделирования и требуемую скорость!','Help');

end

if app.Button\_Full.Value == 1

app.Flag\_mode = 4;

app.Label\_Vz.Visible = false;

app.Label\_t.Visible = false;

app.EditField\_Vz.Visible = false;

app.EditField\_t.Visible = false;

app.Label\_kd.Visible = false;

app.Label\_kp.Visible = false;

app.EditField\_kp.Visible = false;

app.EditField\_kd.Visible = false;

end

end

% 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 = 0.007;% допустимая погрешность, т.к. идеально не попадем

if app.Flag\_mode == 4%full

if app.Flag\_ship == 1%париж

dFmax = 0.1\*Fmax;

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

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

dP = 10;

if (dP >= dPmax)

dP = dPmax;

end

%разгон

while ( (app.Vmax-v(i)) > eps)

tt = t(i)+app.dt;

t = [t tt];

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

x = [x xx];

vv = (x(i+1)-x(i))/app.dt;

v = [v vv];

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

break

end

end

while ( v(i) > eps)

tt = t(i)+app.dt;

t = [t tt];

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

x = [x xx];

vv = (x(i+1)-x(i))/app.dt;

v = [v vv];

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

if app.Flag\_ship == 2%катер

dFmax = 0.2\*Fmax;

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

dP = 10;

if (dP >= dPmax)

dP = dPmax;

end

while ( (app.Vmax-v(i)) > eps)

tt = t(i)+app.dt;

t = [t tt];

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

x = [x xx];

vv = (x(i+1)-x(i))/app.dt;

v = [v vv];

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

break

end

end

%торможение

while ( v(i) > eps)

tt = t(i)+app.dt;

t = [t tt];

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

x = [x xx];

vv = (x(i+1)-x(i))/app.dt;

v = [v vv];

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

if app.Flag\_ship == 3%КПК

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

dFmax = 0.2\*Fmax;

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

dP = 10;

if (dP >= dPmax)

dP = dPmax;

end

A1 = Fmax/app.V1max^2;

A2 = Fmax/app.V2max^2;

while ( (app.V2max-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

tt = t(i)+app.dt;

t = [t tt];

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

x = [x xx];

vv = (x(i+1)-x(i))/app.dt;

v = [v vv];

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

break

end

end

%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

tt = t(i)+app.dt;

t = [t tt];

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

x = [x xx];

vv = (x(i+1)-x(i))/app.dt;

v = [v vv];

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

if app.Flag\_mode == 1%разгон

if app.Flag\_ship == 1%париж

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

A = Fmax/app.Vmax^2;

dFmax = 0.1\*Fmax;

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

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

dP = 10;

if (dP >= dPmax)

dP = dPmax;

end

%разгон

while ( (app.Vmax-v(i)) > eps)

tt = t(i)+app.dt;

t = [t tt];

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

x = [x xx];

vv = (x(i+1)-x(i))/app.dt;

v = [v vv];

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

break

end

end

end

if app.Flag\_ship == 2%катер

dFmax = 0.2\*Fmax;

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

dP = 10;

if (dP >= dPmax)

dP = dPmax;

end

while ( (app.Vmax-v(i)) > eps)

tt = t(i)+app.dt;

t = [t tt];

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

x = [x xx];

vv = (x(i+1)-x(i))/app.dt;

v = [v vv];

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

break

end

end

end

if app.Flag\_ship == 3%КПК

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

dFmax = 0.2\*Fmax;

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

dP = 10;

if (dP >= dPmax)

dP = dPmax;

end

A1 = Fmax/app.V1max^2;

A2 = Fmax/app.V2max^2;

while ( (app.V2max-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

tt = t(i)+app.dt;

t = [t tt];

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

x = [x xx];

vv = (x(i+1)-x(i))/app.dt;

v = [v vv];

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

break

end

end

end

end

if app.Flag\_mode == 2%торможение

if app.Flag\_ship == 1%париж

dFmax = 0.1\*Fmax;

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

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

dP = 10;

if (dP >= dPmax)

dP = dPmax;

end

dx = v(i)\*app.dt;

while ( v(i) > eps)

tt = t(i)+app.dt;

t = [t tt];

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

x = [x xx];

vv = (x(i+1)-x(i))/app.dt;

v = [v vv];

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

if app.Flag\_ship == 2%катер

dFmax = 0.2\*Fmax;

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

dP = 10;

if (dP >= dPmax)

dP = dPmax;

end

dx = v(i)\*app.dt;

while ( v(i) > eps)

tt = t(i)+app.dt;

t = [t tt];

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

x = [x xx];

vv = (x(i+1)-x(i))/app.dt;

v = [v vv];

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

if app.Flag\_ship == 3%КПК

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

dFmax = 0.2\*Fmax;

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

dP = 10;

if (dP >= dPmax)

dP = dPmax;

end

A1 = Fmax/app.V1max^2;

A2 = Fmax/app.V2max^2;

dx = v(i)\*app.dt;

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

tt = t(i)+app.dt;

t = [t tt];

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

x = [x xx];

vv = (x(i+1)-x(i))/app.dt;

v = [v vv];

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

if app.Flag\_mode == 3%регулятор

app.kp = app.EditField\_kp.Value;% коэф ПД регулятора

app.kd = app.EditField\_kd.Value;% коэф ПД регулятора

if app.Flag\_ship == 1%париж

dFmax = 0.1\*Fmax;

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

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

while(t(i) < app.t\_mod)

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

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

x = [x xx];

vv = (x(i+1)-x(i))/app.dt;

v = [v vv];

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

if (P(i)+dP > app.Pmax)

P = [P app.Pmax];

end

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

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

end

i = i+1;

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

end

end

if app.Flag\_ship == 2%катер

dFmax = 0.2\*Fmax;

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

app.kp = 7;

app.kd = 25;

dx = 0;

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

while(t(i) < app.t\_mod)

tt = t(i)+app.dt;

t = [t tt];

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

x = [x xx];

vv = (x(i+1)-x(i))/app.dt;

v = [v vv];

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

if (P(i)+dP > app.Pmax)

P = [P app.Pmax];

end

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

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

end

i = i+1;

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

end

end

if app.Flag\_ship == 3%КПК

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

dFmax = 0.2\*Fmax;

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

app.kp = 3;

app.kd = 15;

A1 = Fmax/app.V1max^2;

A2 = Fmax/app.V2max^2;

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

while(t(i) < app.t\_mod)

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

tt = t(i)+app.dt;

t = [t tt];

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

x = [x xx];

vv = (x(i+1)-x(i))/app.dt;

v = [v vv];

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

if (P(i)+dP > app.Pmax)

P = [P app.Pmax];

end

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

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

end

i = i + 1;

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

end

end

end

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

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

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

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

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

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

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

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

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

else

helpdlg('Выберите режим работы','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

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 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\_Paris

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

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

app.Button\_Paris.Text = '«Парижская Коммуна»';

app.Button\_Paris.Position = [11 108 150 22];

app.Button\_Paris.Value = true;

% Create Button\_Kater

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

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

app.Button\_Kater.Text = 'Малый рабочий катер';

app.Button\_Kater.Position = [11 68 145 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 EditFieldW

app.EditFieldW = uieditfield(app.UIFigure, 'text');

app.EditFieldW.Tag = 'EditFieldW';

app.EditFieldW.Editable = 'off';

app.EditFieldW.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 EditFieldN

app.EditFieldN = uieditfield(app.UIFigure, 'text');

app.EditFieldN.Tag = 'EditFieldN';

app.EditFieldN.Editable = 'off';

app.EditFieldN.Position = [371 519 60 22];

% Create EditFieldV

app.EditFieldV = uieditfield(app.UIFigure, 'text');

app.EditFieldV.Tag = 'EditFieldV';

app.EditFieldV.Editable = 'off';

app.EditFieldV.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 EditFieldV2max

app.EditFieldV2max = uieditfield(app.UIFigure, 'text');

app.EditFieldV2max.Tag = 'EditFieldV2max';

app.EditFieldV2max.Editable = 'off';

app.EditFieldV2max.Position = [371 429 60 22];

% Create EditFieldV1max

app.EditFieldV1max = uieditfield(app.UIFigure, 'text');

app.EditFieldV1max.Tag = 'EditFieldV1max';

app.EditFieldV1max.Editable = 'off';

app.EditFieldV1max.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.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.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.Position = [191 109 25 22];

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

% Create Label\_kd

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

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.Position = [231 109 50 22];

% Create EditField\_kd

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

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

% 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];

% 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