РЕФЕРАТ

Выпускная квалификационная работа содержит 45 страниц, 21 рисунок, 5 использованных источников.

ДВИЖЕНИЕ РОБОТА БЕЗ ВНЕШНИХ ДВИЖИТЕЛЕЙ, ПОДВИЖНАЯ ВНУТРЕННЯЯ МАССА, АНИЗОТРОПНОЕ ТРЕНИЕ, ДВИЖЕНИЕ ПО ШЕРОХОВАТОЙ ПЛОСКОСТИ.

Выпускная квалификационная работа посвящена исследованию механической системы, состоящей из прямоугольного твердого тела, совершающего движение по шероховатой плоскости, и точки, двигающейся по окружности относительно прямоугольного твердого тела с постоянной угловой скоростью. Учтено условие различия коэффициента трения при движении вперед и движении назад.

Получены уравнения движения системы в размерном и безразмерном виде. Проведено численное интегрирование этих уравнений в среде MATLAB. Выделены пять типов движения корпуса. Построен ряд диаграмм при различных значениях отношения коэффициентов трения, где для диапазона остальных параметров задачи построены области, отвечающие этим типам движения.

По набору диаграмм сформулированы выводы об изменении областей типов движения в зависимости от отношения коэффициентов трения.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 4](#_Toc73871576)

[ОСНОВНАЯ ЧАСТЬ 6](#_Toc73871577)

[1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 7](#_Toc73871578)

[1.1 Постановка задачи 7](#_Toc73871579)

[1.2 Уравнения движения 8](#_Toc73871580)

[1.3 Безразмерные уравнения движения 9](#_Toc73871581)

[1.4 Примеры интегрирования 10](#_Toc73871582)

[1.5 Типы движения 13](#_Toc73871583)

[2. ПРАКТИЧЕСКАЯ ЧАСТЬ 20](#_Toc73871584)

[2.1 Построение диаграмм для разных значений b 20](#_Toc73871585)

[ЗАКЛЮЧЕНИЕ 29](#_Toc73871586)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 30](#_Toc73871587)

[ПРИЛОЖЕНИЯ 31](#_Toc73871588)

[Приложение 1. MY\_Equation.m 32](#_Toc73871589)

[Приложение 2. MY\_Main\_Program.m 33](#_Toc73871590)

[Приложение 3.Count\_of\_stops\_and\_napravlenie.m 35](#_Toc73871591)

[Приложение 4. Diagramm.m 37](#_Toc73871592)

[Приложение 5. PointDihotomy.m 45](#_Toc73871593)

# ВВЕДЕНИЕ

Роботы, движение которых осуществляется благодаря перемещению внутренней массы, являются примером робототехнических систем без внешних движителей. В данной научной работе проводится построение и анализ модели такой механической системы. Исследование таких систем представляет большой интерес с теоретической и практической точек зрения, вследствие того, что устройства такого типа могут иметь значительные преимущества по сравнению с робототехническими системами других типов [1].

Изучением таких систем занимается такая научная дисциплина как робототехника. Среди прочих научных дисциплин робототехника выделяется своей актуальностью и перспективностью применения в самых разных областях: в промышленности и медицине, в сельском хозяйстве и строительстве, в системах безопасности и даже в индустрии развлечений. Использование мобильных роботов позволяет снизить участие человека в рискованной и тяжелой работе – именно поэтому важность и актуальность этой прикладной науки крайне высока.

Роботы, не обладающие внешними подвижными элементами, могут использоваться в самых разных областях – например, для работы в космосе и исследования космических тел, в медицине [1], при работе под водой. В случае работы мобильного робота в жидкости отсутствие внешних подвижных элементов является значительным преимуществом, так как воздействие на окружающую среду оказывается минимальное, а кроме того – появляется возможность обеспечения полной гидроизоляции [3]. В общем случае, важнейшими преимуществами роботов данного типа являются: простота конструкции (как следствие, упрощается и ускоряется процесс их разработки), а также возможность функционирования в агрессивных средах (к примеру, появляется возможность работы на больших глубинах с высоким давлением).

В данной научной работе рассматривается механическая система, состоящая из корпуса, совершающего движение по шероховатой плоскости, и точки, двигающейся по окружности относительно корпуса с постоянной угловой скоростью. При этом коэффициент трения при движении вперед меньше, чем при движении назад. Получены уравнения движения, проведено численное интегрирование этих уравнений, выделены пять типов движения корпуса. Построен ряд диаграмм при различных значениях отношения коэффициентов трения, где для диапазона остальных параметров задачи построены области, отвечающие этим типам движения.

По набору диаграмм сформулированы выводы об изменении областей типов движения в зависимости от отношения коэффициентов трения.

# ОСНОВНАЯ ЧАСТЬ

# 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## 1.1 Постановка задачи

Рассмотрим механическую систему, состоящую из прямоугольного твердого тела массой 𝑀 находящегося на шероховатой горизонтальной поверхности, и подвижной массы – материальной точки массы m, движущейся внутри корпуса по окружности радиуса R. Центр окружности совпадает с геометрическим центром короба. Угловая скорость ω радиус-вектора, задающего положение подвижной массы, постоянна.

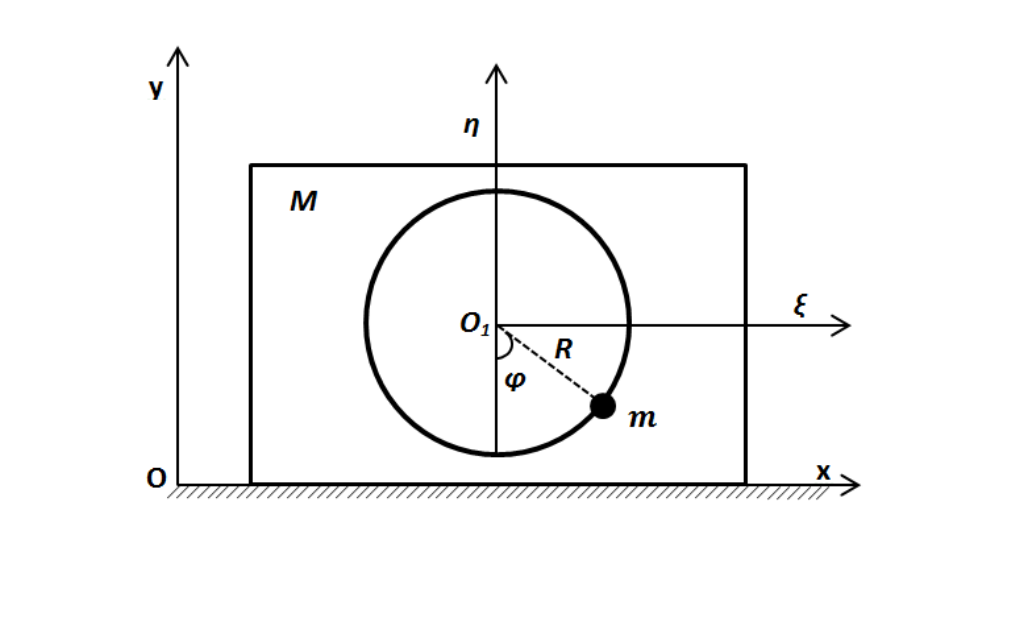


Рис. 1.1 Механическая система, представленная в задании

При движении в сторону положительного направления оси 𝑥 прямоугольный твердый корпус преодолевает действие силы трения с коэффициентом *k1*, а при движении в сторону отрицательного направления – с коэффициентом *k2,* причем *k1* равно *b\*k2*, где *b* – коэффициент пропорциональности. Будем считать, что *k1*<*k2*, т.е. вперед корпусу двигаться легче чем назад (учитывая это условие – очевидно, что *b*<1).

## 1.2 Уравнения движения

Для составления уравнений движения введем декартову прямоугольную систему координат 𝑂𝑥𝑦. Также введём систему координат Oξη, жёстко связанную с коробом. Центр системы совпадает с центром окружности. Тогда координаты подвижной массы в этой системе координат будут иметь вид:

Предполагаем, что в начальный момент времени подвижная материальная точка находилась в крайней нижней точке окружности. Из этого следует, что положения корпуса можно однозначно определить по координатам его центра масс, и уравнения движения будут записаны в виде:

Здесь сила трения вычисляется по следующей формуле:

## 1.3 Безразмерные уравнения движения

Введем безразмерные координаты:

Для удобства сохраним для новых координат прежние обозначения. Кроме того, будем предполагать, что прямоугольное твердое тело не может двигаться по вертикали (т.е. ускорение по оси y=0). В результате подстановки безразмерных координат в уравнения движения, получаем следующую систему уравнений:

(1.1)

Коэффициенты μ и *n* вычисляются по следующим формулам:

,

Безразмерная сила трения *f* будет равна:

В случае движения по вертикали безразмерная реакция опоры *n* в системе (1.1) обратится в ноль. Следовательно, для того, чтобы движение было безотрывным, необходимо, чтобы в любой момент времени выполнялось условие *n>0.*

Из второго уравнения системы получим условие безотрывного движения, но уже для параметра μ.

Т.к. при любом *t* и (из 2-го уравнения системы (1.1)), то:

, при любом *t*.

, при любом *t*.

.

## 1.4 Примеры интегрирования

Будем проводить исследование движения корпуса при помощи численных методов. С помощью встроенной в MATLAB функции ode45, реализующей одношаговые явные методы Рунге­-Кутта 4­-го и 5­-го порядков решения систем обыкновенных дифференциальных уравнений (ОДУ), проинтегрируем систему уравнений движения, полученную нами ранее. Результатом интегрирования будет являться матрица X, состоящая из 2-ух столбцов. В 1-ом столбце будут находиться значения координат *x*, а во 2-м столбце – значения скоростей *v*. Каждая координата *x* и каждая скорость *v* будут вычисляться в соответствии с некоторым значением времени t. Шаг по времени (интервалы, через которые программа ищет решение) определяется параметром grstep = 4π/1000.

Для интегрирования будем брать промежуток времени t = [0, 4π], за который подвижная материальная точка сделает два обращения по окружности.

Так как решение системы ДУ проводится численно, то значения координаты *x* и скорости *v* получаются лишь приближёнными, вычисленными с определённой погрешностью. Для того, чтобы избежать влияния этой погрешности на дальнейшие расчёты, будем считать, что:

1. Значение координаты *x* равно нулю, если модуль *x* меньше определённого значения Xtol*.* Примем Xtol=0.000000001.
2. Система находится в покое, если модуль скорости *v* меньше определённого значения Vtol. Примем Vtol=0.00000000001.

В среде MATLAB были составлены две функции: MY\_Equation.m и MY\_Main\_Program.m. Полный листинг кода этих функций представлен в разделе «Приложение 1» и «Приложение 2» соответственно.

Эти функции выполняют следующие действия:

1. ***MY\_Equation.m***

Представляет систему (1.1) в виде системы ДУ 1-го порядка в явной форме Коши:

Вычисляет правые части этой системы.

1. ***MY\_Main\_Program.m***

При помощи встроенной функции ode45 выполняет интегрирование системы ОДУ, полученных функцией MY\_Equation.m. При помощи встроенной функции plot выводит два графика зависимости x(t) и v(t) при заданных значениях параметров *μ, k2* и *b.*

Примеры построения графиков зависимости x(t) и v(t) программой MY\_Main\_Program.m:

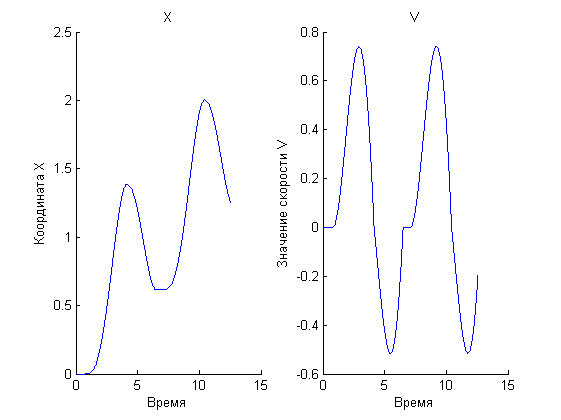


Рис. 1.2 b = 1, k2 = 0.3, μ = 1.7

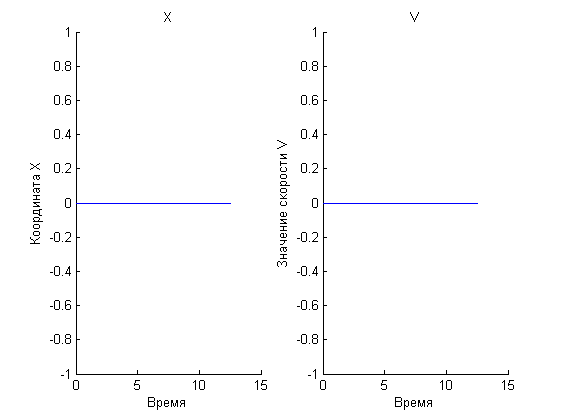


Рис. 1.3 b = 1, k2 = 0.75, μ = 1.7

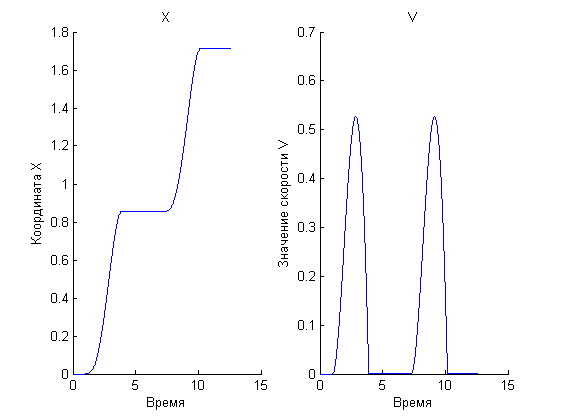


Рис. 1.4 b = 0.5, k2 = 0.75, μ = 1.7

## 1.5 Типы движения

Проведём качественный анализ типов движения корпуса по количеству продолжительных остановок (далее – остановок).

Случай движения без остановок возникает, когда трение недостаточно, чтобы остановить движение как в одном, так и в другом направлении. При этом наблюдаются мгновенные моменты нулевой скорости – в момент изменения направления движения.

Случаи движения с одной остановкой разделяются на два вида: с изменением направления движения через остановку (с возвратным движением) и без изменения направления (без возвратного движения).   
Первый вид возникает, когда сила трения в одном направлении недостаточна для того, чтобы остановить движение, а в другом – достаточна. При изменении направления движения с прямого (с меньшим коэффициентом трения) на обратное (с бо́льшим) происходит остановка, необходимая для того, чтобы движущая сила дошла до уровня, большего, чем сила трения скольжения в обратном направлении. При изменении направления движения с обратного на прямое такой остановки не происходит, поскольку к этому моменту движущая сила уже больше силы трения: скорость на мгновение принимает значение 0 и сразу начинает расти.

Второй вид движения с одной остановкой – без изменения направления движения – возникает, когда сила трения в обратном направлении достаточна для того, чтобы вообще не допустить движения тела в нём. Этот тип движения невозможен при изотропном трении (*b=*1 – случай, рассмотренный в работе А.С. Панёва [1]), так как в этом случае величина силы трения в обоих направлениях одинакова, и движение либо есть в обоих направлениях, либо его в обоих направлениях нет. Иными словами, при *b=*1 по мере увеличения *k* или *μ* движение в обоих направлениях прекращается одновременно.

Случай движения корпуса с двумя остановками возникает, когда сила трения в обоих направлениях достаточно велика, чтобы вызвать продолжительную остановку, но недостаточна, чтобы вообще не допустить движения.

Случай постоянного покоя возникает, когда сила трения покоя в обоих направлениях достаточно велика, чтобы вообще не допустить движения.

Теперь проведём качественный анализ типов движения по наличию и направлению смещения за полный период.

Случай, когда смещение за период равно 0, может возникнуть в двух случаях:

1. Движения вообще нет, корпус находится в покое.
2. Движение есть в обоих направлениях, причём смещение в прямом направлении равно смещению в обратном. Такой вариант движения возможен только в случае изотропной силы трения, то есть при . Этот случай мы видим в практической части на диаграмме для : все точки с синей заливкой имеют жёлтый обод. На диаграммах для других значений *b* жёлтый обод имеют только точки зоны постоянного покоя – с белой заливкой.

Случай, когда смещение за период направлено вперёд, возможен, если сила трения при движении вперёд меньше, чем при движении назад – то есть при .

Случай, когда смещение за период направлено назад, возможен, если сила трения при движении вперёд больше, чем при движении назад, то есть при .

Приведём графики координаты и скорости для указанных типов движения по количеству остановок:

1. Без остановок

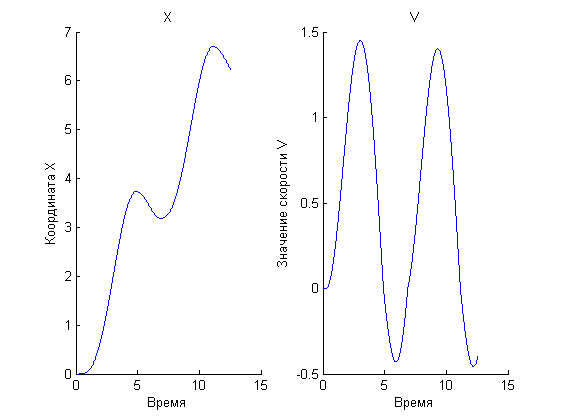


Рис. 1.5 b = 0.75, k2 = 0.1, μ =2.5

1. Одна остановка с возвратным движением

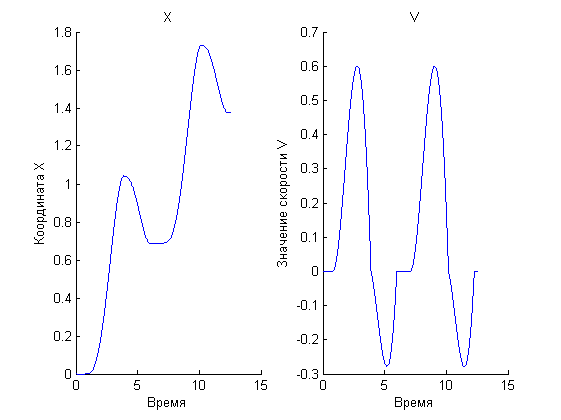


Рис. 1.6 b = 0.75, k2 = 0.3, μ =2.5

1. Две остановки

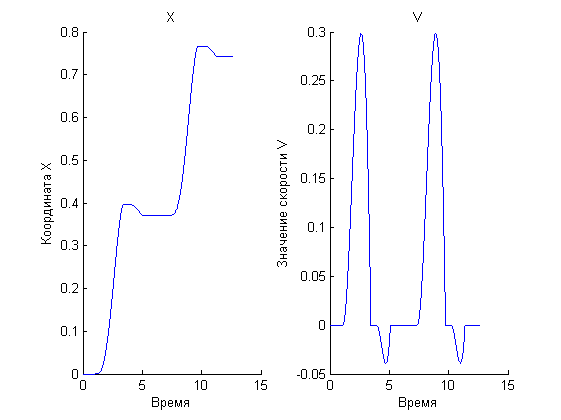


Рис. 1.7 b = 0.75, k2 = 0.4, μ =2.5

1. Одна остановка без возвратного движения

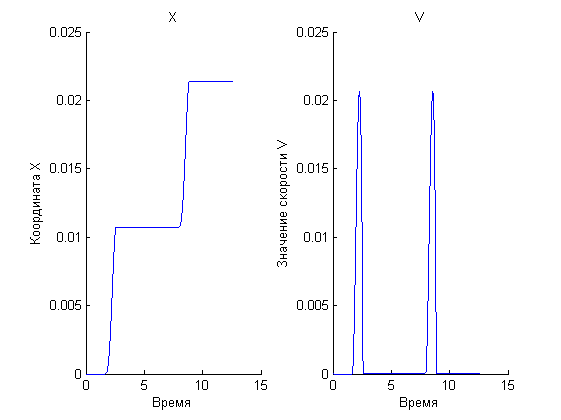


Рис. 1.8 b = 0.75, k2 = 0.55, μ =2.5

1. Покой

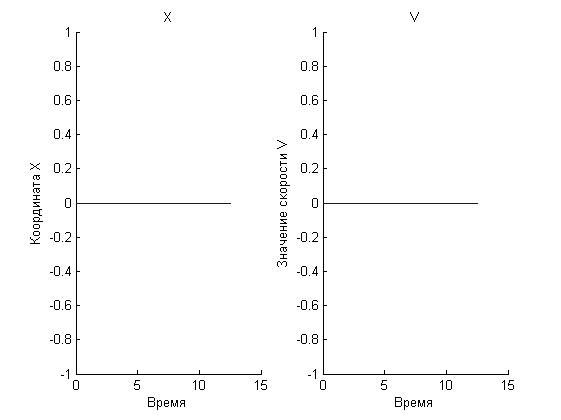


Рис. 1.9 b = 0.75, k2 = 0.8, μ =2.5

# 2. ПРАКТИЧЕСКАЯ ЧАСТЬ

## 2.1 Построение диаграмм для разных значений b

Построим ряд диаграмм типов движения при фиксированных значениях в координатах параметров *k* и *,* где *,* а *.*

Для *k* используем шаг0.02. Для используем шаг 0.08.

В данных диаграммах будем использовать следующие условные обозначения:

* Цвет заливки точки:

1. Красный – движение без остановок.
2. Зелёный – одна остановка с возвратным движением.
3. Синий – две остановки.
4. Оранжевый – одна остановка без возвратного движения.
5. Белый – состояние покоя.

* Цвет обода точки:

1. Жёлтый – движение с возвращением в исходное положение.
2. Голубой – движение вперед.
3. Фиолетовый – движение назад.

Для построения этих диаграмм были разработаны функция **count\_of\_stops\_and\_napravlenie.m** (Приложение 3) и класс **Diagramm** (файл **Diagramm.m**, Приложение 4) .

Функция **count\_of\_stops\_and\_napravlenie.m** выполняет следующие действия. Принимая в качестве параметров значения *k*, *μ* и *b*, она выполняет интегрирование системы ОДУ, полученных функцией MY\_Equation.m, а затем, проанализировав полученные последовательности координат и скоростей, возвращает в качестве своего значения типы движения по количеству остановок и по смещению за полный период.

Класс **Diagramm** вначале выполняет классификацию по типам движения точек координатной плоскости, расположенных в узлах сетки разбиения (метод **AllClassify**). Затем он строит диаграмму, выполняя отображение этих точек в соответствии с их классификацией (метод **Postroit**).

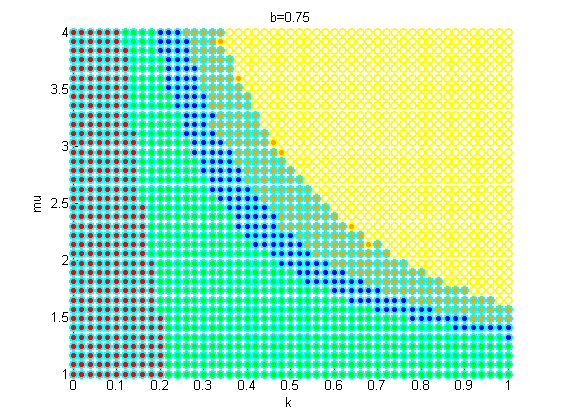


Рис. 2.1 Диаграмма типов движения при b = 0.75

Проведём уточнение границ между областями, используя метод дихотомии. Для этого после нанесения каждой точки на плоскость будем сравнивать тип движения в ней с типом движения у соседней с ней (сверху и слева - при наличии такого соседа). При обнаруженном отличии типа движения вызовем функцию дихотомии, которая методом половинного деления уточнит расположение точки на отрезке между этими двумя точками (то есть координаты точки *μ* и *k*). Найденные таким образом пары значений (*k, μ*) сохраняем в двумерных массивах (используя отдельный массив для каждого типа движения справа или сверху от найденной точки). После заполнения всех этих массивов они сортируются по *k*, и по ним строятся линии раздела между областями. Для построения линии раздела между областью постоянного покоя и соседней с ней такой метод не используется, так как формула для построения этой линии выводится аналитически: .

Для реализации описанного выше алгоритма разработана функция **PointDihotomy.m** (Приложение 5), выполняющая поиск более точной границы на отрезке между двумя точками координатной плоскости в координатах (*k, μ*) методом дихотомии. Также внесены соответствующие дополнения в класс **Diagramm.**

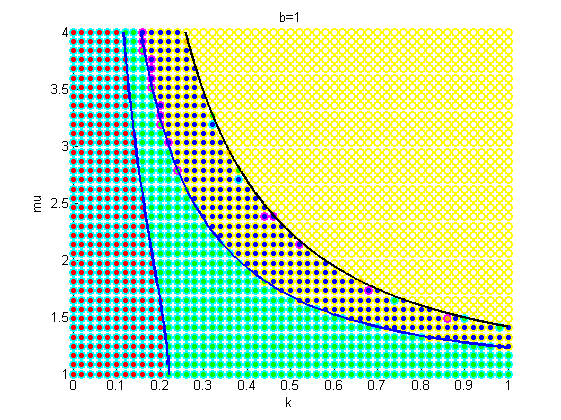


Рис. 2.2 Диаграмма типов движения при b = 1

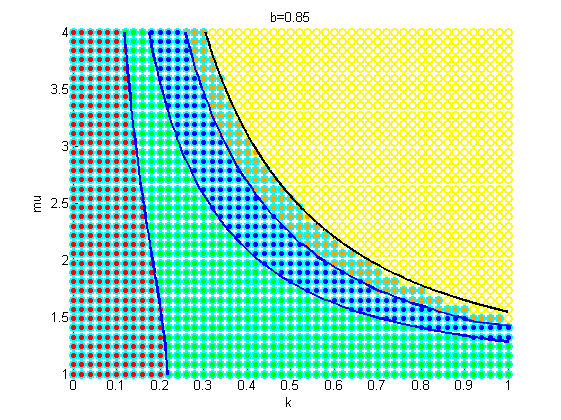


Рис. 2.3 Диаграмма типов движения при b = 0.85

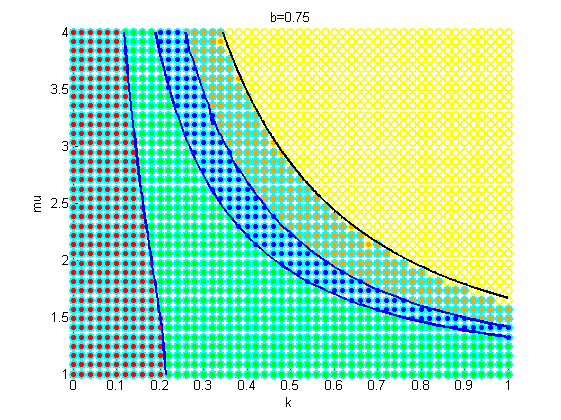


Рис. 2.4 Диаграмма типов движения при b = 0.75

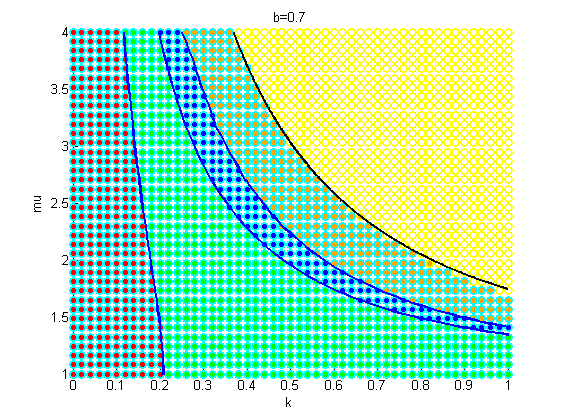


Рис. 2.5 Диаграмма типов движения при b = 0.7

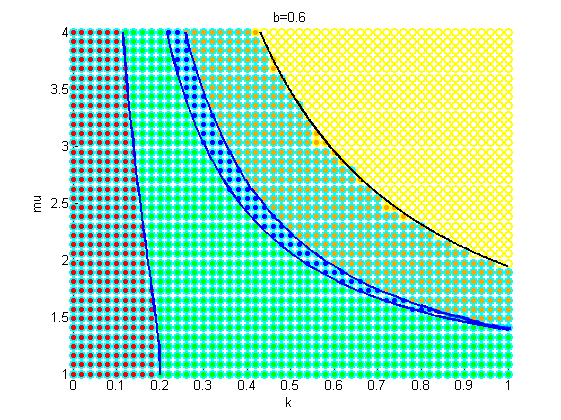


Рис. 2.6 Диаграмма типов движения при b = 0.6

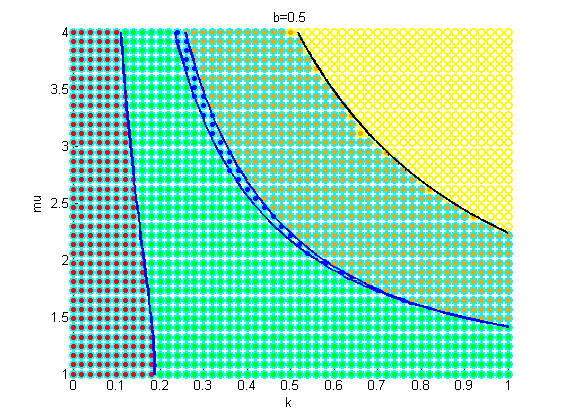


Рис. 2.7 Диаграмма типов движения при b = 0.5

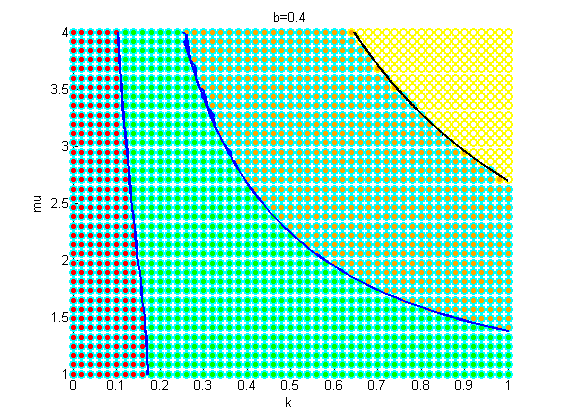


Рис. 2.8 Диаграмма типов движения при b = 0.4

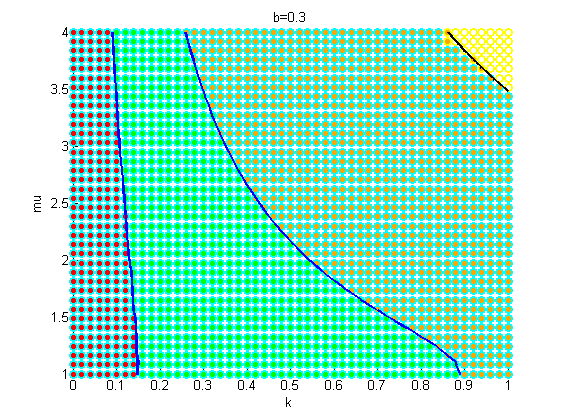


Рис. 2.9 Диаграмма типов движения при b = 0.3

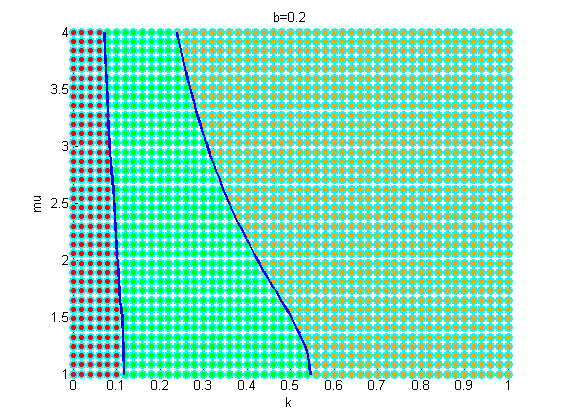


Рис. 2.10 Диаграмма типов движения при b = 0.2

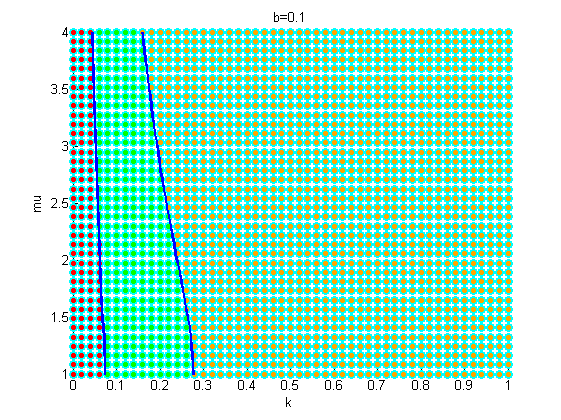


Рис. 2.11 Диаграмма типов движения при b = 0.1

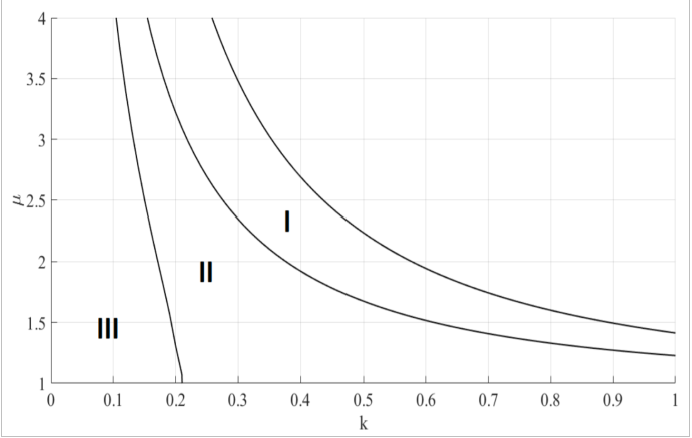


Рис. 2.12 Диаграмма, полученная в работе Панёва А.С.

Для случая b = 1 полученная диаграмма совпадает с полученной в работе Панёва [1].

# ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы:

1. Были получены безразмерные дифференциальные уравнения движения корпуса с движущейся относительно него по окружности подвижной массой.

2. Полученные уравнения численно проинтегрированы с помощью средств среды программирования MATLAB.

3. Сформулированы значимые критерии классификации полученных решений и составлена соответствующая программа.

4. Построены диаграммы областей, характеризующих различные типы движения в зависимости от отношения коэффициентов силы трения в прямом и обратном направлениях.

5. Уточнены границы областей с различными типами движения корпуса в зависимости от параметра коэффициента трения.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

* 1. Панёв А.С. Исследование движения тела по горизонтальной плоскости под влиянием перемещения внутренней массы. // Труды МАИ. 2015. Т. 85.
  2. Болотник Н.Н., Градецкий В.Г., Жуков А.А., Козлов Д.В., Смирнов И.П., Чащухин В.Г. Мобильный микроробот космического назначения: концепция и перспективы использования // Космические исследования. 2018
  3. Клековкин А. В. Исследование динамики движения в жидкости роботов с неизменяемой формой оболочки и управляемых внутренними роторами. // Ижевск: Ижевский государственный технический университет имени М.Т. Калашникова. 2020.
  4. Бардин Б.С., Панёв А.С. О периодических движениях тела с подвижной внутренней массой по горизонтальной поверхности. // Труды МАИ. 2015. T. 84.
  5. Маркеев А.П. Теоретическая механика: Учебник для университетов. 3-е изд.// 2007

# ПРИЛОЖЕНИЯ

## Приложение 1. MY\_Equation.m

function Xdot = MY\_equation(t,X)

global myu k1 k2 Vtol

n=cos(t)+myu;

s=-sin(t);

if abs(X(2)) < Vtol

X(2) = 0;

if s < -k1\*n

fc = -k1\*n;

elseif s > k2\*n

fc = k2\*n;

else

fc = s;

end

elseif X(2) > 0

fc = -k1\*n;

else

fc = k2\*n;

end

Xdot(1)=X(2);

Xdot(2)=fc-s;

Xdot=Xdot';

end

## Приложение 2. MY\_Main\_Program.m

global myu k1 k2 Xtol Vtol;

myu = 1.48 %1.08 %1.3; % 1.5;

k2 = 0.36 %0.5 %0.44 % 0.2; %0.6; %k1 и k2 - коэффициенты трения. Предполагаем, что отношение k1 к k2 равно b.

b = 0.75; %0.75;

k1 = b\*k2;

Vtol=0.0000000001;

Xtol=0.0000000001;

X0 = [0,0]; % Начальные условия (задача Коши) - вектор-столбец из величин s, fi, ds/dt, dfi/dt.

t0 = 0; % Начальное время интегрирования (сек)

n=10000;% Шаг выдачи результатов интегрирования (сек)

tfin=4\*pi;

grstep = tfin/n;

tout=t0:grstep:tfin;

opts = odeset('RelTol',1e-12,'AbsTol',1e-14); %создаем опцию с именем RelTol, которой будет соответствовать точность 1e-12

% также создаем опцию с именем AbsTol которой будет соответствовать точность 1e-14

tic

[t,X] = ode45(@MY\_equation,tout,X0,opts);

toc

figure

subplot(1,2,1) %производит разбивку графического окна на несколько подокон

hold on

plot(t,X(:,1));

title('X');

xlabel('Время')

ylabel('Координата X')

%whitebg('w')

subplot(1,2,2)

hold on

plot(t,X(:,2));

title('V');

xlabel('Время')

ylabel('Значение скорости V')

## Приложение 3.Count\_of\_stops\_and\_napravlenie.m

function [count, napr] = count\_of\_stops\_and\_napravlenie(current\_k, current\_myu, b)

global k myu tol k1 k2 Vtol;

k = current\_k;

k1=b\*k;

k2=k;

myu = current\_myu;

fprintf('count\_of\_stops\_and\_napravlenie b=%f k=%f myu=%f \n', b, k, myu);

tol = 1e-14;

Xtol=0.000000001;

Vtol=0.00000000001;

x0 = 0;

v0 = 0;

t\_start = 0;

t\_fin = 4 \* pi;

t\_step = 0.01;

t = t\_start:t\_step:t\_fin;

out = [x0, v0];

opts = odeset('RelTol',tol\*100,'AbsTol',tol);

[t, out] = ode45(@MY\_equation, t, out,opts);

X = out(:, 1);

V = out(:, 2);

if max(abs(V)) < Vtol

count = 4; % Покой нам только снился.

else

[isPositiveV, isNegativeV] = deal(false);

count = 0;

leng\_v = length(V);

i = 1;

while i < leng\_v

isPositiveV = isPositiveV || V(i) > Vtol;

isNegativeV = isNegativeV || V(i) < -Vtol;

if abs(V(i)) < Vtol && abs(V(i + 1)) < Vtol

count = count + 1;

while i < length(V) - 1 && abs(V(i)) < Vtol

i = i + 1;

end

end

i = i + 1;

end

count = floor(count/2);

if count == 1 && not (isNegativeV && isPositiveV)

count = 3;

end

end

leng\_x = length(X);

napr = sign(X(leng\_x) - X(floor(leng\_x / 2)));

if abs(X(leng\_x) - X(floor(leng\_x / 2))) < Xtol

napr = 0;

end

end

## Приложение 4. Diagramm.m

classdef Diagramm

%UNTITLED2 Summary of this class goes here

% Detailed explanation goes here

properties

Parameter;

Points;

ClassOfPoints;

kLength;

muLength;

end

methods

function D = Diagramm(parameter)

D.Parameter=parameter;

D.Points=[];

D.ClassOfPoints=[];

D.kLength=0;

D.muLength=0;

kNum=51;

muNum=38;

for I=1:kNum

k=(I-1)\*(1.0/(kNum-1));

for J=1:muNum

mu=1+(J-1)\*(3.0/(muNum-1));

fprintf('length(D.Points)= %d ', length(D.Points));

D.Points((I-1)\*muNum+J,1)=k;

fprintf(' %d ', length(D.Points));

D.Points((I-1)\*muNum+J,2)=mu;

fprintf(' %d \n', length(D.Points));

D.ClassOfPoints((I-1)\*muNum+J,1)=-1;

D.ClassOfPoints((I-1)\*muNum+J,2)=-1;

end

if D.muLength==0

D.muLength=muNum;

end

end

D.kLength=kNum;

fprintf('length(D.Points)= %d \n', length(D.Points));

% finish;

end

function D = OnlyOneClassify(D,I)

S=-1;

N=-1;

k=D.Points(I,1);

mu=D.Points(I,2);

[S, N] = count\_of\_stops\_and\_napravlenie(k, mu, D.Parameter);

D.ClassOfPoints(I,1)=S;

D.ClassOfPoints(I,2)=N;

end

function D = AllClassify(D)

for I=1:length(D.Points)

D=OnlyOneClassify(D,I);

end

end

function D = krest(k, myu)

plot(k, myu, '+', 'Color', [0 0 0], 'LineWidth', 1);

fprintf('krest k=%f myu=%f \n', k, myu);

end

function Postroit(D)

disp('Postroit');

index=@(kNum,muNum) (kNum-1)\*D.muLength+muNum;

figure

xlim([0 1])

ylim([1 4])

hold on

title(['b=',num2str(D.Parameter)]);

xlabel('k')

ylabel('mu')

FaceColors = [ [1 0 0]; [0 1 0]; [0 0 1]; [1, 165.0/255, 0]; [1 1 1]]; %red, green, blue, orange, white

ConturColors = [ [1 0 1]; [1 1 0]; [0 1 1] ]; %magenta (розовый), yellow, cyan (сине-зелёный)

[bp1,bp2,bp3,bp4] = deal([],[],[],[]);

[bc1,bc2,bc3,bc4] = deal(0);

for kNum=1:D.kLength

for muNum=1:D.muLength

I = index(kNum,muNum);

FaceColor = FaceColors(D.ClassOfPoints(I,1)+1,:);

ConturColor = ConturColors(D.ClassOfPoints(I,2)+2,:);

plot(D.Points(I,1),D.Points(I,2),'o','Color',ConturColor,'MarkerFaceColor',FaceColor,'LineWidth',2);

leftIndex=index(kNum-1,muNum);

if abs(D.Points(I,1)-0.850000) < 0.0001 && abs(D.Points(I,2)-1.52941) < 0.0001

disp(D.Points(I,2));

end

if not ((kNum<D.kLength && D.ClassOfPoints(index(kNum+1,muNum),1) == 4) || (muNum<D.muLength && D.ClassOfPoints(I+1,1) == 4))

dist=1;

if D.ClassOfPoints(I,1) == 3

dist = 2;

end

if kNum>1 && D.ClassOfPoints(I,1)~=4 && D.ClassOfPoints(I,1) > D.ClassOfPoints(leftIndex,1) && D.ClassOfPoints(I,1) <= dist+D.ClassOfPoints(leftIndex,1)

P = PointDihotomy(D.Parameter, D.Points(leftIndex,1), D.Points(leftIndex,2), D.ClassOfPoints(leftIndex,:), D.Points(I,1), D.Points(I,2), D.ClassOfPoints(I,:), 15);

%

if D.ClassOfPoints(I,1) == 0

bc1 = bc1+1;

[bp1(bc1, 1), bp1(bc1, 2)] = deal(P(1), P(2));

elseif D.ClassOfPoints(I,1) == 1

bc2 = bc2+1;

[bp2(bc2, 1), bp2(bc2, 2)] = deal(P(1), P(2));

elseif D.ClassOfPoints(I,1) == 2

bc3 = bc3+1;

[bp3(bc3, 1), bp3(bc3, 2)] = deal(P(1), P(2));

elseif D.ClassOfPoints(I,1) == 3

bc4 = bc4+1;

[bp4(bc4, 1), bp4(bc4, 2)] = deal(P(1), P(2));

end

if false && muNum>1 && D.ClassOfPoints(leftIndex,1) == D.ClassOfPoints(leftIndex-1,1)

P = PointDihotomy(D.Parameter, D.Points(leftIndex-1,1), D.Points(leftIndex-1,2), D.ClassOfPoints(leftIndex-1,:), D.Points(I,1), D.Points(I,2), D.ClassOfPoints(I,:), 16);

if D.ClassOfPoints(I,1) == 0

bc1 = bc1+1;

[bp1(bc1, 1), bp1(bc1, 2)] = deal(P(1), P(2));

elseif D.ClassOfPoints(I,1) == 1

bc2 = bc2+1;

[bp2(bc2, 1), bp2(bc2, 2)] = deal(P(1), P(2));

elseif D.ClassOfPoints(I,1) == 2

bc3 = bc3+1;

[bp3(bc3, 1), bp3(bc3, 2)] = deal(P(1), P(2));

elseif D.ClassOfPoints(I,1) == 3

bc4 = bc4+1;

[bp4(bc4, 1), bp4(bc4, 2)] = deal(P(1), P(2));

end

end

end

if muNum>1 && D.ClassOfPoints(I,1)~=4 && D.ClassOfPoints(I,1) > D.ClassOfPoints(I-1,1) && D.ClassOfPoints(I,1) <= dist+D.ClassOfPoints(I-1,1)

P = PointDihotomy(D.Parameter, D.Points(I-1,1), D.Points(I-1,2), D.ClassOfPoints(I-1,:), D.Points(I,1), D.Points(I,2), D.ClassOfPoints(I,:), 15);

if D.ClassOfPoints(I,1) == 0

bc1 = bc1+1;

[bp1(bc1, 1), bp1(bc1, 2)] = deal(P(1), P(2));

elseif D.ClassOfPoints(I,1) == 1

bc2 = bc2+1;

[bp2(bc2, 1), bp2(bc2, 2)] = deal(P(1), P(2));

elseif D.ClassOfPoints(I,1) == 2

bc3 = bc3+1;

[bp3(bc3, 1), bp3(bc3, 2)] = deal(P(1), P(2));

elseif D.ClassOfPoints(I,1) == 3

bc4 = bc4+1;

[bp4(bc4, 1), bp4(bc4, 2)] = deal(P(1), P(2));

end

end

end

end

end

if bc2 > 1

bp2 = sortrows(bp2,1);

plot2 = plot(bp2(:,1), bp2(:,2),'-','LineWidth',2);

end

if bc3 > 1

bp3 = sortrows(bp3,1);

plot3 = plot(bp3(:,1),bp3(:,2),'-','LineWidth',2);

end

if bc4 > 1

bp4 = sortrows(bp4,1);

plot4 = plot(bp4(:,1),bp4(:,2),'-','LineWidth',2);

end

disp('====== bp1 ======');

disp(bp1);

disp('====== bp2 ======');

disp(bp2);

disp('====== bp3 ======');

disp(bp3);

disp('====== bp4 ======');

disp(bp4);

k\_on\_mu = @(mu) 1/(D.Parameter\*sqrt(mu^2-1));

mu\_on\_k = @(k) sqrt(1+1/(D.Parameter\*k)^2);

x=k\_on\_mu(4):0.001:1;

y=1:length(x);

for I=1:length(x)

y(I)=mu\_on\_k(x(I));

end

plot(x,y,'-','Color', [0 0 0],'LineWidth',2);

end

end

end

## Приложение 5. PointDihotomy.m

function P = PointDihotomy( b, k\_begin, myu\_begin, C\_begin, k\_end, myu\_end, C\_end, steps)

k=(k\_begin+k\_end)/2;

myu = (myu\_begin+myu\_end)/2;

if steps > 0

C = count\_of\_stops\_and\_napravlenie(k, myu, b);

if (C(1)==C\_begin(1))

P=PointDihotomy(b, k, myu, C, k\_end, myu\_end, C\_end, steps-1);

else

P=PointDihotomy(b, k\_begin, myu\_begin, C\_begin, k, myu, C, steps-1);

end

else

P=[k,myu];

fprintf('Dihotomy k=%f myu=%f', k, myu);

end

end