Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт информационных технологий
и анализа данных

наименование института

Отчет

к лабораторной работе №2

«Методы оптимизации»

по дисциплине

«Синтез оптимальных систем с использованием принципа максимума»

наименование темы

Вариант №20

Выполнил студент	АСУб-21-2		О. Э. Кудряшов
	шифр группы	подпись	И.О. Фамилия
Проверил			И. А. Серышева
	должность	подпись	И.О. Фамилия

Иркутск 2024г.

1. Функциональная схема системы управления с данными варианта

Цель работы: ознакомление с методикой синтеза оптимальных систем регулирования с использованием средств вычислительной техники.

Условия задачи: в данной лабораторной работе необходимо с помощью регулятора обеспечить оптимальное по быстродействию управление движения спутника вокруг центра масс по одной из осей. Возмущения отсутствуют. Задачу решить, используя принцип максимума Л.С. Понтрягина. На рисунке 1 представлена функциональная схема системы управления.

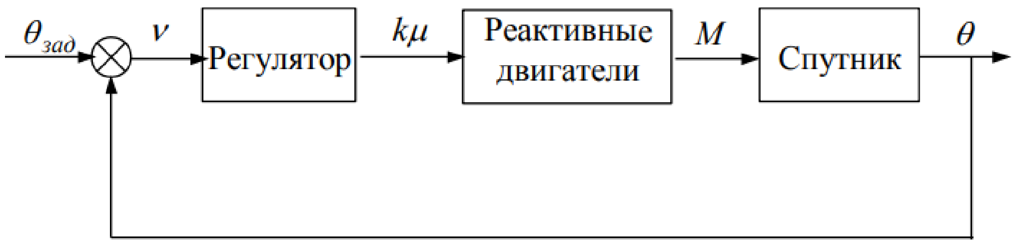

Рисунок 1 – Функциональная схема системы управления

Регулятор – элемент, который определяет управляющее воздействие на систему и обеспечивает достижение заданных требований к управлению.

Здесь регулятор принимает отклонение v и выдает управляющий сигнал μ, который может принимать значение +1 или -1.

Реактивные двигатели являются исполнительными органами управления, в зависимости от результата работы регулятора, он выдает управляющий момент М, определяющий величину крутящего момента, необходимого для изменения угловой скорости объекта, который описывается реактивными двигателями с регулируемой тягой. Значения управляющего момента ограничены $M \le M_{Max} \le M_m$

Уравнения движения спутника (1), по условию задачи, должен находить момент М, который позволит спутнику переходить из любого отклоненного положения в нормально ориентированное за минимальное время.

(1) $$J\frac{d^2v}{dt^2} = M$$

где J – момент инерции спутника, характеризующий инертность твердого тела относительно вращения вокруг определенной оси.

Момент инерции спутника J = 3000 *кгм*^{2}. Исполнительными органами управления являются реактивные двигатели с регулируемой тягой, развивающей максимальный момент M_m = 50 *Нм*. При оптимальном управлении найти время, необходимое для перехода спутника в установившееся нулевое состояние, если в начальный момент времени его отклонение составляло 260 градусов, а угловая скорость -20 *градс*$^{-1}$. Возмущения отсутствуют. Задачу решить используя принцип максимума Л.С. Понтрягина.

2. **Фазовые диаграммы углового движения спутника при $u = \pm 1$ с описанием процесса их построения**

Для решения задачи преобразуем уравнение (1), введя обозначения:

$$x_1 = v_{\text{рад}};\ \ x_2 = \frac{dv_{\text{рад}}}{dt} = \dot{v}_{\text{рад}}; k = \frac{M}{J} = \frac{50}{3000} = 0{,}017$$

Преобразуем уравнение движения спутника в систему дифференциальных уравнений первого порядка:

$$(2)\ \begin{cases} \frac{dx_1}{dt} = x_2 \\ \frac{dx_2}{dt} = k\mu \end{cases}, k\mu - \text{ ускорение движения}$$

Здесь u – это нормированная функция управления $|u| \leq 1$, $v_{\text{рад}} = v_{\text{град}} * \frac{\pi}{180} = \ 4{,}538$, $\dot{v}_{\text{рад}} = \dot{v}_{\text{град}} * \frac{\pi}{180} = \ -0{,}349$

Теперь составим гамильтониан по формуле (3):

(3) $\quad H = \sum_{i=1}^{n} \Psi_i f_i(x_1, x_2, \mu).$

Для совокупности уравнений (2) гамильтониан будет:

(4) $\quad H = \Psi_1 x_2 + \Psi_2 k\mu$

Максимум этой функции с учетом ограничения на управляющий сигнал μ и обеспечивает оптимальность системы по быстродействию. Очевидно, что при наложенных ограничениях максимум H имеет место, если управляющий сигнал μ формируется исходя из алгоритма:

$$\mu = sign\psi_2$$

Таким образом, оптимальное по быстродействию управление будет осуществляться в том случае, если регулятор будет переключать исполнительное устройство по релейному закону в соответствии со знаком вспомогательной функции ψ_2. Для того чтобы найти ψ_2, запишем изначальные системы через гамильтоновы функции.

$$(5)\ \begin{cases} \frac{d\psi_1}{dt} = -\frac{\partial H}{\partial x_1} = -\frac{\partial(\psi_1 x_2 + \psi_2 k\mu)}{\partial x_1} = 0 \\ \frac{d\psi_2}{dt} = -\frac{\partial H}{\partial x_2} = -\frac{\partial(\psi_1 x_2 + \psi_2 k\mu)}{\partial x_2} = -\psi_1 \end{cases}$$

Далее необходимо проинтегрировать эти выражения. Сделав это, получим ψ_2 и ψ_1:

$$(6)\ \int \frac{d\psi_1}{dt} = \int 0 => \psi_1 = C_1$$

$$(6)\ \int \frac{d\psi_2}{dt} = -\int C_1 dt => \psi_2 = -C_1 t + C_2$$

Где C_1 и $C_2 - const$

Подставляя (6) в уравнение (4), получаем:

$$(7)\ \ H = C_1 x_2 + (C_2 - C_1 t)\mu k.$$

Максимум H будет зависеть от знака коэффициента при μ, то есть:

$$(8)\quad \mu = sign(\Psi_2) = sign(C_2 - C_1 t)$$

Задача по быстродействию будет иметь не более одного решения u. И, так как $C_2 - C_1 t$ – является прямой линией, то знак меняется не более одного раза на промежутке $t_0 \le t \le t_1$. Из этого следует, что $\mu(t)$ имеет 2 интервала постоянства.

Тогда управляющий сигнал должен менять знак при достижении линии переключения, чтобы перейти на её траекторию:

Рассмотрим 2 случая – когда $\mu = 1$ и $\mu = -1$:

1 случай:

$\mu = 1$

$$\begin{cases} \dfrac{dx_1}{dt} = x_2 \\ \dfrac{dx_2}{dt} = k \Rightarrow x_2 = k\left(t + \tilde{C}\right) \end{cases}$$

$$\begin{cases} x_1 = \dfrac{x_2^2}{2k} + \mathrm{C} \\ x_2 = k\left(t + \tilde{C}\right) \end{cases} \quad (1)$$

2 случай:

$\mu = -1$

По аналогии с первым случаем получаем:

$$\begin{cases} \dfrac{dx_1}{dt} = x_2 \\ \dfrac{dx_2}{dt} = -k \end{cases} \Rightarrow \begin{cases} x_1 = -\dfrac{x_2^2}{2k} + C \\ x_2 = -k\left(t + \tilde{C}\right) \end{cases} \quad (2)$$

Этим уравнениям соответствуют параболы, каждая из которых симметрична относительно оси x1 = v.

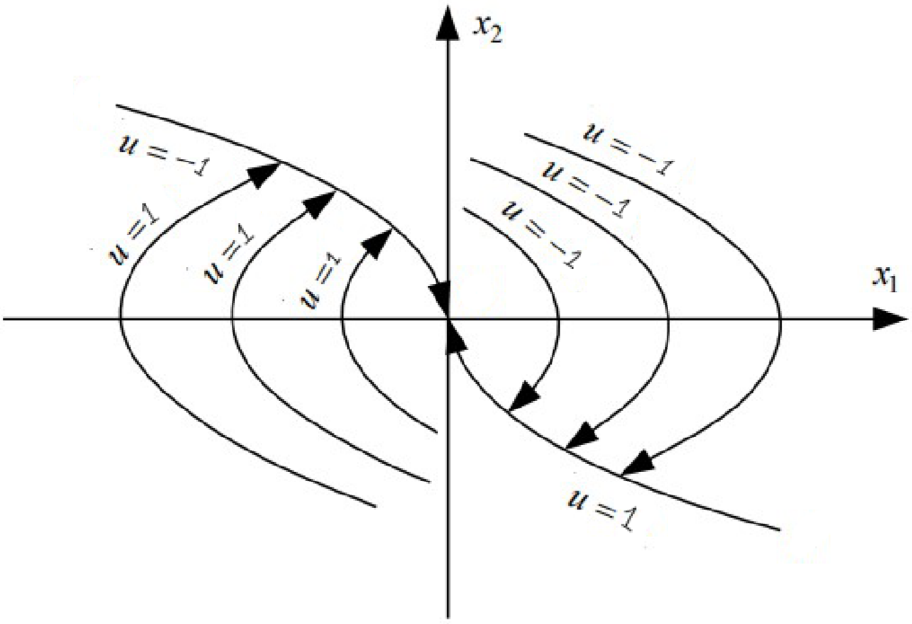

Рисунок 2 – Параболы с направлениями движения

Спутник придёт и будет оставаться в исходном положении, если изображающая точка на фазовой плоскости попадёт в начало координат:

$$x_1 = \nu \equiv 0, x_2 = \frac{d\nu}{dt} \equiv 0$$

Для этого при произвольном начальном отклонении изображающая точка должна достичь линии переключения, выделенной красным цветом на рисунке 3. После этого сменить траекторию с помощью изменения знака управляющей функции:

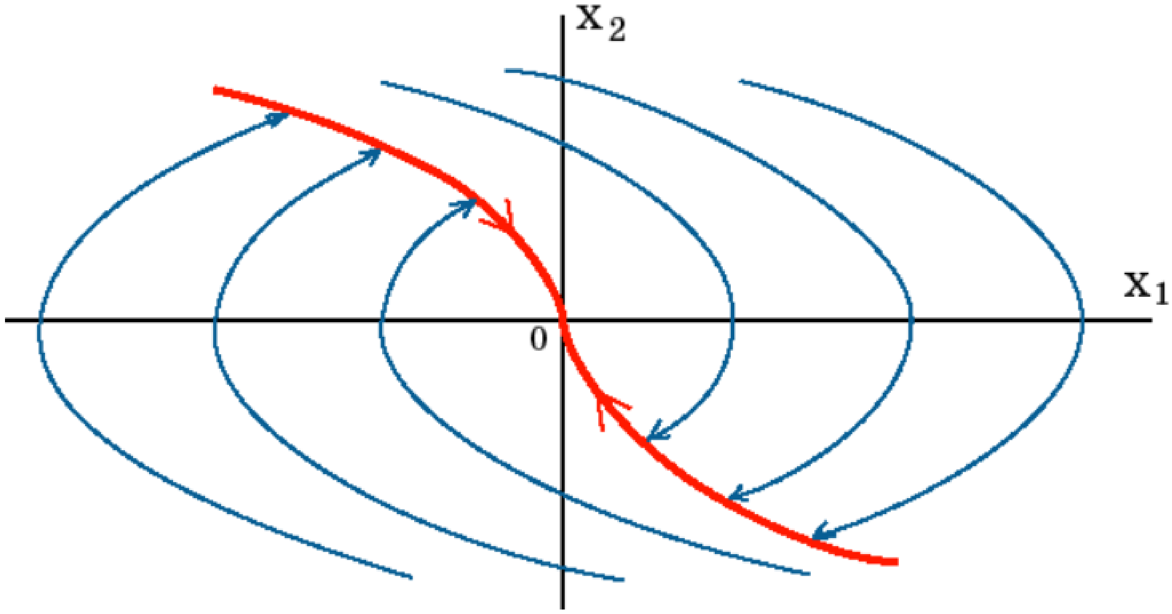

Рисунок 3 – параболы и линия переключения

Таким образом, управляющий сигнал должен менять свой знак при попадании на линию переключения согласно условиям:

$$\mu(\nu) = \begin{cases} +1 \text{ при } \dfrac{d\nu}{dt} < \left(\dfrac{d\nu}{dt}\right)_{\text{п}} \text{ и при } \dfrac{d\nu}{dt} = \left(\dfrac{d\nu}{dt}\right)_{\text{п}} < 0, \\ -1 \text{ при } \dfrac{d\nu}{dt} > \left(\dfrac{d\nu}{dt}\right)_{\text{п}} \text{ и при } \dfrac{d\nu}{dt} = \left(\dfrac{d\nu}{dt}\right)_{\text{п}} > 0. \end{cases}$$

Таким образом регулятор должен состоять из устройства, дифференцирующего ошибку v по времени, устройства вычисляющего скорость $\left(\frac{dv}{dt}\right)_{\text{п}}$ на линии переключения и логического устройства, которое получая на вход выходы из двух других устройств сравнивает полученные значения и принимает решение, какой выходной управляющий сигнал формировать ($\mu = \pm 1$), основываясь на системе, описанной выше. На рисунке 4 представлена функциональная схема такого регулятора.

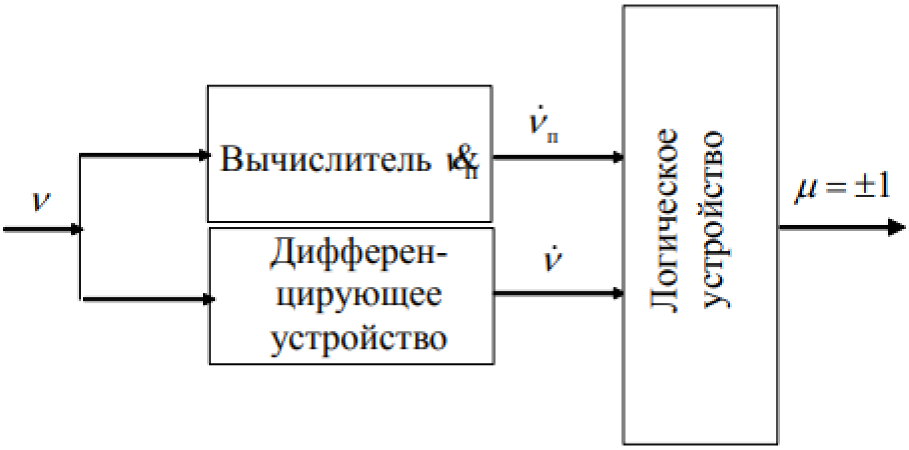

Рисунок 4 – Функциональная схема регулятора

3. Фазовые диаграммы углового движения спутника при оптимальном управлении

Изобразим процесс движения на фазовой плоскости, исключив из совокупности уравнений dt. Для $\mu = \pm 1$ получим:

$$\begin{cases} x_2 dx = \pm k dx_1 \\ \dfrac{x_2^2}{2} = \pm k x_1 + C \end{cases}$$

Возьмем $x_1 = v, x_2 = \frac{dv}{dt}$, тогда получим что $\frac{\frac{dv^2}{dt}}{2} = \pm kv \Rightarrow \frac{dv}{dt} = \pm\sqrt{2kvsign(v)}$.

Тогда уравнение линии переключения (значение скорости x_2 для линии переключения на фазовой плоскости в начальный момент времени, то есть в начальном положении спутника) имеет вид:

$$\dot{v}_п = \left(\frac{dv}{dt}\right)_п = \pm\sqrt{2kvsignv}.$$

Подставляя числовые значения, получим значение скорости x_2 для линии переключения на фазовой плоскости в начальный момент времени, то есть в начальном положении спутника:

$$\dot{v}_п = \left(\frac{dv}{dt}\right)_п = \sqrt{2 * 0.01667 * 4.5378 * sign(4.5378)} \approx 0.389$$

Так как $\dot{v}_n < \dot{v}_0$, то $\mu = -1$. Найдем точку пересечения траектории спутника до достижения линии переключения и линии переключения.

Уравнение параболы до переключения имеет вид $\frac{x_2^2}{2k} = C - x_1$. Выразим С из этого уравнения и получим $C = \frac{x_2^2}{2k} + kx_1$. Подставим в него начальные условия:

$$C = \frac{x_2^2}{2k} + x_1 = \frac{0.3491^2}{2 * 0.01667} + (4.5378) \approx 8.1933$$

Теперь можно найти скорость точки переключения, приравняв уравнение параболы, по которой движется спутник и уравнение линии переключения:

$$x_1 = -\frac{x_2^2}{2k} + C \text{ и } x_1 = sign(\, x_1)\frac{x_2^2}{2k} \Rightarrow \ x_1 = \ x_1 \Rightarrow -\frac{x_2^2}{2k} + \mathrm{C} = \frac{x_2^2}{2k} \Rightarrow x_2^2 = C * k \Rightarrow x_2 = -\sqrt{C * k}.$$

Знак точки переключения получается «-», так как переключение происходит в области отрицательных значений x_2.

Подставив соответствующие значения получим:

$$\dot{v}_1 = -\sqrt{8.1933 * 0.01667} = -0.3695.$$

Для расчета минимального времени необходимого спутнику для достижения нулевого состояния, запишем уравнения в виде системы:

$$\begin{cases} \dfrac{dx_1}{dt} = x_2, \\ \dfrac{dx_2}{dt} = k\mu. \end{cases}$$

Тогда оптимальное время перехода спутника из заданного положения в нулевое может быть определено следующим образом:

$$\frac{dx_2}{dt} = \frac{d\dot{v}}{dt} = k\mu.$$

Проинтегрируем полученное уравнение:

$$\int_{t_1}^{t_2} k\mu dt = \int_{t_1}^{t_2} \frac{d\dot{v}}{dt} dt \ \Rightarrow k\mu(t_2 - t_1) = \dot{v}_2 - \dot{v}_1.$$

Отсюда следует что:

$\triangle\, t_1 = t_1 - t_0 = \frac{\dot{v}_1 - \dot{v}_0}{k\mu}$, μ = -1,

$\triangle\, t_2 = t_2 - t_1 = \frac{\dot{v}_2 - \dot{v}_1}{k\mu}$, μ = 1, так как произошло переключение на линию АОВ.

Подставив значения получим:

$$\triangle\, t_1 = \frac{-0.3695 + 0{,}349}{0.01667 * (-1)} = 1.2298\text{с},$$

$$\triangle\, t_2 = \frac{0 - (-0.409)}{0.018 * (1)} = 22.1655\text{с}.$$

Тогда общее время $\triangle = \triangle\, t_1 + \triangle\, t_2 = 1.2298 + 22.1655 = 23.3953\text{с}$.

4. Моделирование

Алгоритм работы моделирования

1. Вычисляем скорость линии переключения по заданному x_1.
2. Устанавливаем значение μ по следующему правилу:

$$\mu(\nu) = \begin{cases} +1 \text{ при } \nu < \nu_n \text{ или при } \nu = \nu_n < 0, \\ -1 \text{ при } \nu > \nu_n \text{ или при } \nu = \nu_n > 0. \end{cases}$$

3. Моделируем новые значения x_1 и x_2 используя формулы, полученные методом Рунге-Кутта.

4. Проверяем, равны ли x_1 и x_2 нулю. Если да, то завершение работы алгоритма, иначе возвращаемся к пункту 1.

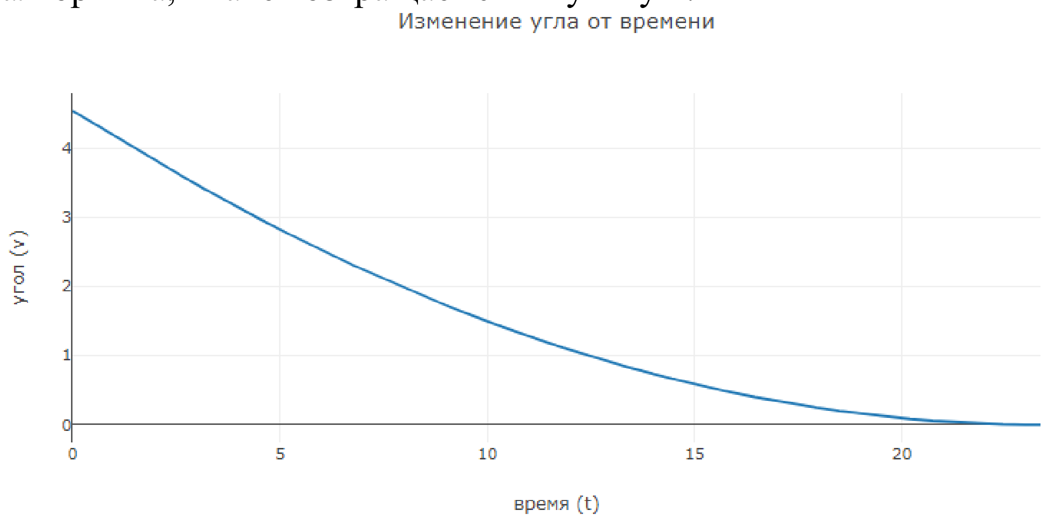

Рисунок 5 – График результатов решения дифференциальных уравнений с использованием метода Рунге-Кутта

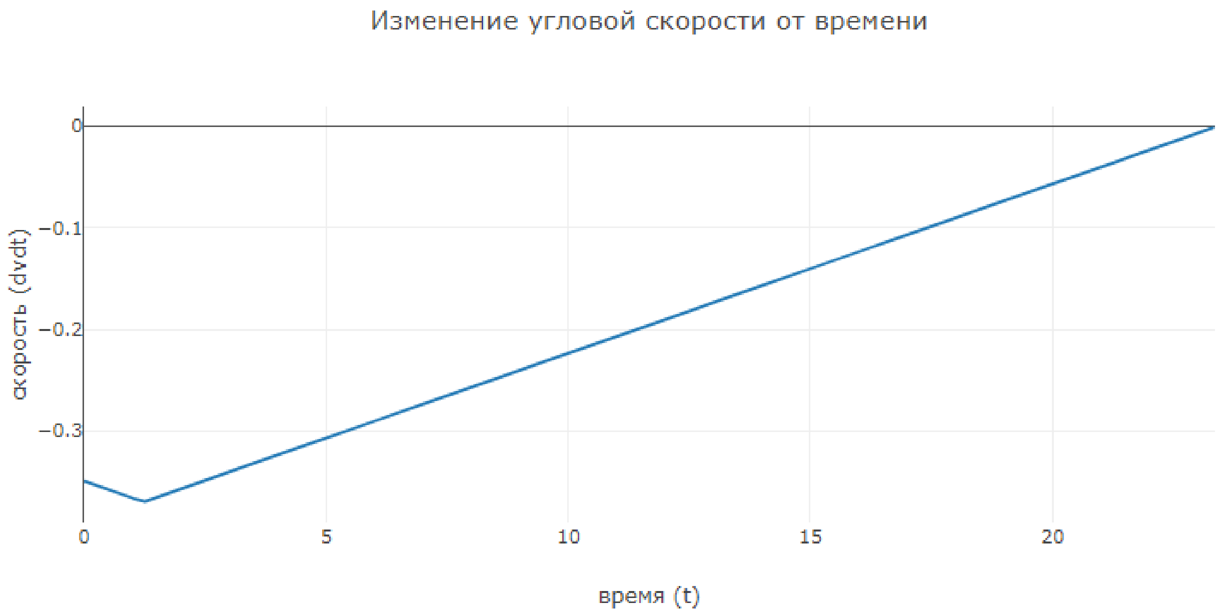

Рисунок 6 - График результатов решения дифференциальных уравнений с использованием метода Рунге-Кутта

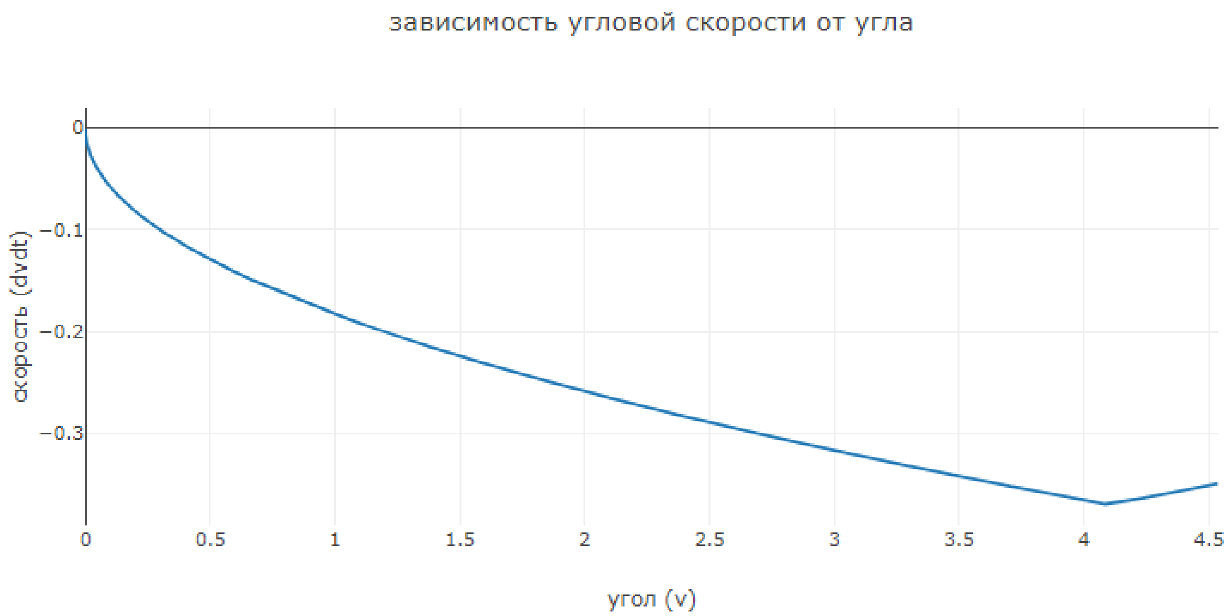

Рисунок 7 – График процесса перехода спутника из заданного положения в нулевое

5. Сравнение вычисленных $\Delta t_1, \Delta t_2$ с моделируемыми $\widetilde{\Delta t_1}, \widetilde{\Delta t_2}$.

Покажем временные промежутки и общее время переходного процесса при моделировании системы.

```
t1: 1.2280111255383885
t2: 22.171962149470342
общее  t: 23.341000000004016
```

Рисунок 8 – время переходного процесса через моделирование

Покажем временные промежутки и общее время переходного процесса при аналитическом решении:

$$\triangle\, t_1 = \frac{-0.3695+0{,}349}{0.01667*(-1)} = \mathbf{1.2298}\text{с},$$

$$\triangle\, t_2 = \frac{0-(-0.409)}{0.018*(1)} = \mathbf{22.1655}\text{с}.$$

Тогда общее время $\triangle = \triangle\, t_1 + \triangle\, t_2 = 1.2298 + 22.1655 = \mathbf{23.3953}$с.

Как видим, решения совпали с точностью до сотых, но точность программного метода можно было бы увеличить, задав большую точность.

6. Листинг

```
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>L2</title>
    <link rel="stylesheet" href="style.css" />
```

```
    <link                                                        rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css" />
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></
script>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"></scrip
t>
    <script src="https://cdn.plot.ly/plotly-2.20.0.min.js" charset="utf-8"></script>
  </head>
  <body>
    <details open>
      <summary>Графики</summary>
      <section>
        <details open>
          <summary>Фазовые портреты</summary>
          <div id="plot-mu"></div>
        </details>
        <details open>
          <summary>Зависимости</summary>
          <div id="plot-v(t)"></div>
          <div id="plot-dvdt(t)"></div>
          <div id="plot-dvdt(v)"></div>
        </details>
      </section>
    </details>

    <script src="plotly-latest.min.js"></script>
    <script>
      function drawPlots() {
        const degree2radian = (degree) => (degree * Math.PI) / 180;
        const dv_dt_i = (k, c1) => -Math.sqrt(2 * k * Math.abs(c1)) * Math.sign(c1);

        let [J, Mm, v0, dv_dt0] = [3000, 50, 260, -20];
        v0 = degree2radian(v0);
        dv_dt0 = degree2radian(dv_dt0);
        const k = Mm / J;
        const c0_mu = (mu) => dv_dt0 ** 2 - mu * 2 * k * v0;

        const dv = 0.001;

        const time_line = [];
        const v_line = [];
```

```
const dvdt_line = [];

const parabola_x_line = [];
const primary_parabola_y_line = [];
const static_parabola_y_line = [];

const switch_x_line = [];
const switch_y_line = [];

let selected_mu;
let selected_mu_value;

let t1_practice = 0;
let t2_practice = 0;

function modelSwitchLine() {
  let v_start = -4;
  let v_end = -v_start;
  while (v_start <= v_end) {
    switch_y_line.push(dv_dt_i(k, v_start));
    switch_x_line.push(v_start);

    v_start += dv;
  }

  selected_mu = dv_dt0 < dv_dt_i(k, v0);
  selected_mu_value = selected_mu ? 1 : -1;
}

function modelPrimaryParabola() {
  let v0_start = -(Math.abs(dv_dt0) + 0.1);
  let v0_end = -v0_start;
  while (v0_start <= v0_end) {
    parabola_x_line.push(v0_start);
    primary_parabola_y_line.push((-selected_mu_value *
(c0_mu(selected_mu_value) - v0_start ** 2)) / 2 / k);

    v0_start += dv;
  }
}

function modelStaticParabola() {
  let v0_start = -(Math.abs(dv_dt0) + 0.1);
  let v0_end = -v0_start;
  while (v0_start <= v0_end) {
```

```
    static_parabola_y_line.push((-selected_mu_value * v0_start ** 2) / 2 / k);

    v0_start += dv;
  }
}

function regulator(t, t_prev, v, v_prev) {
  const dvdt = (v - v_prev) / (t - t_prev);
  const dvdt_p = dv_dt_i(k, v);
  if (dvdt < dvdt_p || (dvdt === dvdt_p && dvdt < 0)) {
    return 1;
  }
  return -1;
}

function sputnik(mu, v, dvdt, dt) {
  let k1 = dvdt * dt;
  let m1 = mu * k * dt;
  let k2 = (dvdt + m1 / 2) * dt;
  let m2 = mu * k * dt;
  let k3 = (dvdt + m2 / 2) * dt;
  let m3 = mu * k * dt;
  let k4 = (dvdt + m3) * dt;
  let m4 = mu * k * dt;

  const new_v = v + (k1 + 2 * k2 + 2 * k3 + k4) / 6;
  const new_dvdt = dvdt + (m1 + 2 * m2 + 2 * m3 + m4) / 6;

  return [new_v, new_dvdt];
}

function modelSystem() {
  const L = 200;
  const dt = 0.001;
  let t_prev = 0;
  let t = 0;
  let mu = selected_mu ? 1 : -1;

  let v = v0;
  let v_prev = v0;
  let dvdt = dv_dt0;

  while (t < L) {
    time_line.push(t);
    v_line.push(v);
```

```
      dvdt_line.push(dvdt);

      mu_prev = mu;
      mu = regulator(t, t_prev, v, v_prev);

      t_prev = t;
      v_prev = v;

      [v, dvdt] = sputnik(mu, v, dvdt, dt);

      if (Math.abs(v) <= 1e-3 && Math.abs(dvdt) <= 1e-3) {
        break;
      }

      mu === selected_mu_value ? (t1_practice += dt) : (t2_practice += dt);

      t += dt;
    }
  }

  function analiticBlock() {
    const v1 = c0_mu(selected_mu_value) / (2 * k);
    const _dv_dt1 = -Math.sqrt(k * Math.abs(v1)) * Math.sign(v1);

    const t1 = Math.abs((_dv_dt1 - dv_dt0 * -selected_mu_value) / k);
    const t2 = Math.abs(-_dv_dt1 / k);

    console.log('J:', J);
    console.log('Mm:', Mm);
    console.log('k:', k);

    console.log('c0_mu, mu = +1:', c0_mu(+1));
    console.log('c0_mu, mu = -1:', c0_mu(-1));

    console.log('v0:', v0);
    console.log('v1:', v1);
    console.log('dv_dt0:', dv_dt0);
    console.log('dv_dt1:', _dv_dt1);

    console.log('аналитическое t1:', t1);
    console.log('аналитическое t2:', t2);

    const proportion = (main, addition) => Math.abs(100 - (addition * 100) /
main).toFixed(4);
```

```
      console.log('практическое t1:', t1_practice);
      console.log('практическое t2:', t2_practice);
      console.log('общее практическое t:', t1_practice + t2_practice);
       console.log('аналитическое t:', t1, t2, t1 + t2);
       console.log('практическое t:', t1_practice, t2_practice, t1_practice + t2_practice);
      console.log('погрешность:', `${proportion(t1, t1_practice)}%`, `${proportion(t2, t2_practice)}%, ${proportion(t1 + t2, t1_practice + t2_practice)}%`);
    }

    function outputPlots() {
      const start_point = {
        name: 'Начальная точка',
        mode: 'trace',
        x: [v0],
        y: [dv_dt0],
      };

      const faza_legend = {
        title: `Фазовая диаграмма при μ = ${selected_mu_value}`,
        xaxis: {
          title: 'угол (v)',
        },
        yaxis: {
          title: 'скорость (dvdt)',
        },
      };

      const v_t_legend = {
        title: 'Изменение угла от времени',
        xaxis: {
          title: 'время (t)',
        },
        yaxis: {
          title: 'угол (v)',
        },
      };

      const dvdt_t_legend = {
        title: 'Изменение угловой скорости от времени',
        xaxis: {
          title: 'время (t)',
        },
        yaxis: {
```

```
    title: 'скорость (dvdt)',
  },
};

const dvdt_v_legend = {
  title: 'зависимость угловой скорости от угла',
  xaxis: {
    title: 'угол (v)',
  },
  yaxis: {
    title: 'скорость (dvdt)',
  },
};

const primary_parabola_plot = {
  name: 'Изображающая линия',
  mode: 'lines',
  x: primary_parabola_y_line,
  y: parabola_x_line,
};

const static_parabola_plot = {
  name: 'Изображающая линия',
  mode: 'lines',
  x: static_parabola_y_line,
  y: parabola_x_line,
};

const v_t_plot = {
  name: 'угол',
  mode: 'lines',
  x: time_line,
  y: v_line,
};

const dvdt_t_plot = {
  name: 'скорость',
  mode: 'lines',
  x: time_line,
  y: dvdt_line,
};

const dvdt_v_plot = {
  name: 'скорость',
  mode: 'lines',
```

```
          x: v_line,
          y: dvdt_line,
        };

        const switch_plot = {
          name: 'Линия переключения',
          mode: 'lines',
          x: switch_x_line,
          y: switch_y_line,
        };

        Plotly.newPlot('plot-mu', [primary_parabola_plot, static_parabola_plot, switch_plot, start_point], faza_legend);
        Plotly.newPlot('plot-v(t)', [v_t_plot], v_t_legend);
        Plotly.newPlot('plot-dvdt(t)', [dvdt_t_plot], dvdt_t_legend);
        Plotly.newPlot('plot-dvdt(v)', [dvdt_v_plot], dvdt_v_legend);
      }

      modelSwitchLine();
      modelStaticParabola();
      modelPrimaryParabola();
      modelSystem();
      analiticBlock();
      outputPlots();
    }

    setTimeout(() => {
      drawPlots();
      document.addEventListener(
        'keypress',
        function (ev) {
          if (ev.code === 'Enter') {
            drawPlots();
          }
        },
        false
      );
    }, 0);
  </script>
 </body>
</html>
```