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** |
| по лабораторной работе №2 по дисциплине «Теория оптимального управления»  «СИНТЕЗ ОПТИМАЛЬНЫХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ ПРИНЦИПА МАКСИМУМА» |
| наименование темы  Вариант № 13 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил студент |  | АСУб-20-2 |  |  |  | А.В. Арбакова |
|  |  | шифр |  | подпись |  | И.О. Фамилия |
| Проверил |  |  |  |  |  | И.А. Серышева |
|  |  |  |  | подпись |  | И.О. Фамилия |
| Работа защищена с оценкой | | | |  | | |

Иркутск 2024 г.

**Содержание**

[1. Функциональная схема системы управления с данными варианта. 3](#_Toc158939220)

[2. Фазовые диаграммы углового движения спутника при μ = +1 с описанием процесса их построения. 4](#_Toc158939221)

[3. Фазовые диаграммы углового движения спутника при оптимальном управлении с описанием процесса их построения. 7](#_Toc158939222)

[4. Процесс вычисления Δt1 , Δt2. 8](#_Toc158939223)

[5. Описание процесса моделирования системы управления движением спутника. 10](#_Toc158939224)

[6. Листинг программы. 12](#_Toc158939225)

1. **Функциональная схема системы управления с данными варианта.**

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

**Вариант**: 13.

Пусть поставлена задача о синтезе регулятора, обеспечивающего оптимальное по быстродействию управление движением спутника вокруг центра масс по одной из осей. Функциональная схема системы управления представлена на рисунке 1.

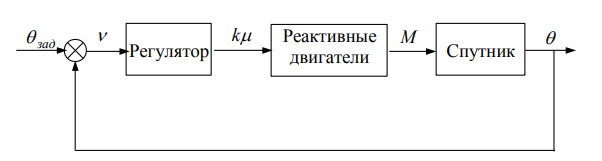


Рисунок 1 – Функциональная схема системы

Момент инерции спутника . Исполнительными органами управления являются реактивные двигатели с регулируемой тягой, развивающие максимальный момент .

При оптимальном управлении найти время, необходимое для перехода спутника в установившееся нулевое состояние, если в начальный момент времени его отклонение составляло , а угловая скорость (рисунок 2). Возмущения отсутствуют.



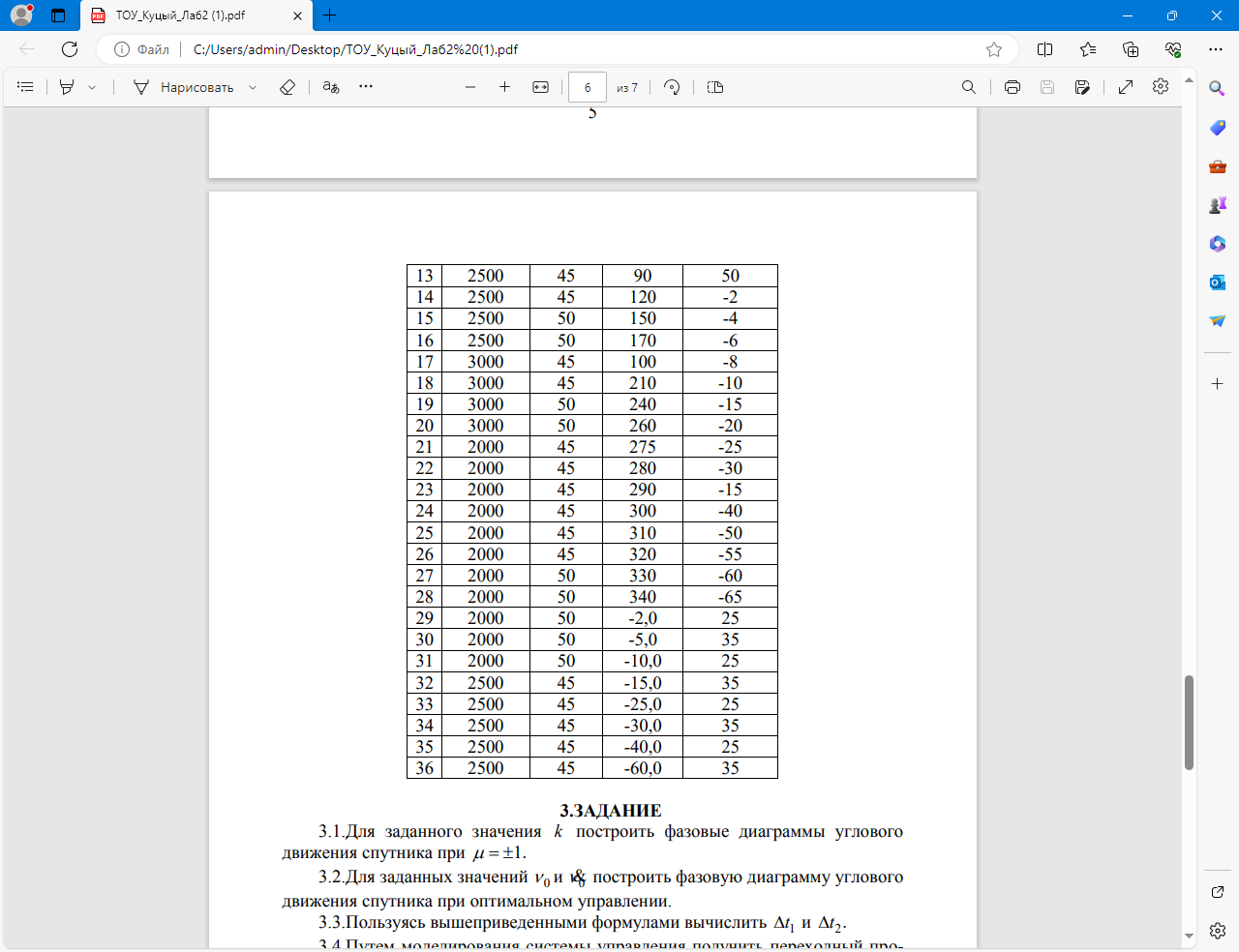


Рисунок 2 – Заданные значения

Задание:

1. Для заданного значения k построить фазовые диаграммы углового движения спутника при μ = ±1.
2. Для заданных значений ν0 и построить фазовую диаграмму углового движения спутника при оптимальном управлении.
3. Пользуясь вышеприведенными формулами вычислить Δt1 и Δt2
4. Путем моделирования системы управления получить переходный процесс спутника из заданного положения в нулевое.
5. Сравнить вычисленные значения Δt1 и Δt2 со значениями и полученными из графика переходного процесса.
6. **Фазовые диаграммы углового движения спутника при μ = -1 с описанием процесса их построения.**

В поставленной задаче оптимального управления мы имеем входные и выходные угловые координаты и три основных компонента функциональной структуры системы (рисунок 3).

1. Регулятор – элемент, который определяет управляющее воздействие на систему и обеспечивает достижение заданных требований к управлению.
2. Ограничитель – элемент, который ограничивает допустимые значения управляющего воздействия, чтобы избежать нарушения ограничений на параметры системы или оборудования.
3. Передаточная функция системы – математическая модель, которая описывает динамику системы и связь между входным управляющим воздействием и выходной реакцией системы.

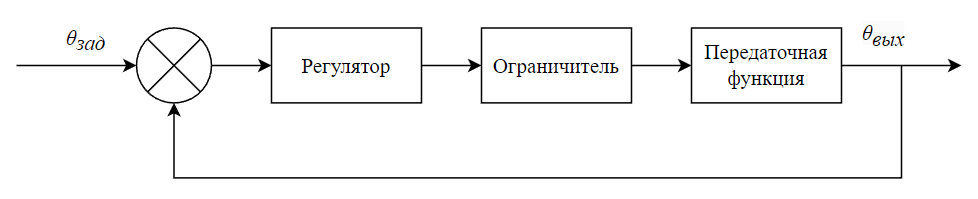


Рисунок 3 – ­ Компоненты функциональной структуры системы

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

– момент управления спутником, определяющий величину крутящего момента, необходимого для изменения угловой скорости объекта, который описывается реактивными двигателями с регулируемой тягой.

– отклонение в начальный момент времени.

– угловая скорость в начальный момент времени.

Уравнение движения спутника, описываемый с помощью второго закона Ньютона для вращательного движения, при отсутствующих возмущениях на систему имеет вид:

Для решения задачи синтеза оптимальных систем мы внедрим принцип Максимума Понтрягина, поскольку поведение спутника может быть описано системой дифференциальных уравнений в нормальной форме. Данный метод позволит найти оптимальное управление, оптимизирующее заданный функционал качества – быстродействие системы, то есть скорость достижения требуемой точности и стабильности в работе.

Для получения предельного быстродействия (минимума времени переходного процесса) необходимо проектировать автоматические системы с учетом ограничений, наложенных на ее координаты, из условия минимума критерия качества вида:

При применении принципа максимума по быстродействию необходимо также учитывать ограничения на момент управления, который ограничен по времени действия

Учитывая условие минимума, можно сделать вывод, что:

Закон управления μ, полученный на основе принципа Максимума Понтрягина, представляет собой релейный закон, который может быть реализован в виде переключателя между двумя установленными значениями.

Первый случай ():

Второй случай ():

На фазовой плоскости получим уравнение . При различных значениях мы наблюдаем разнообразные фазовые траектории, представляющие собой параболы. Заметим, что фазовая траектория, проходящая через начало координат, имеет .

Для нахождения фазовой диаграммы введем обозначения:

Но так как двигатель может развивать различный управляющий момент, то уравнение движение может быть записано в виде:

Где – нормированная функция управления.

Начальные координаты движущегося объекта задаются в градусах, что означает необходимость их перевода в радианы:

Где – положения объекта в момент времени , – скорость объекта в момент времени , – конструктивный параметр пропорциональности предельного момента управления к моменту инерции движущегося объекта, являющийся константой, зависящей только от исходных данных.

В соответствии со вторым законом Ньютона получаем следующие уравнения движения объекта:

Решаем динамическую систему с двумя фазовыми переменными :

Из решения системы дифференциальных уравнений следует, что система условий и дифференциальных уравнений, задающая линию переключения, выглядит описывается образом:

Для получения более точных результатов можно использовать численный метод Рунге-Кутта по решению дифференциальных уравнений (Данная задача оптимального управления подразумевает решения дифференциального уравнения второго порядка):

Для определения оптимального времени перехода объекта из заданного положения в оптимальное (нулевое положение) выведем следующие формулы:

Построим фазовые диаграммы углового движения спутника (рисунок 4).

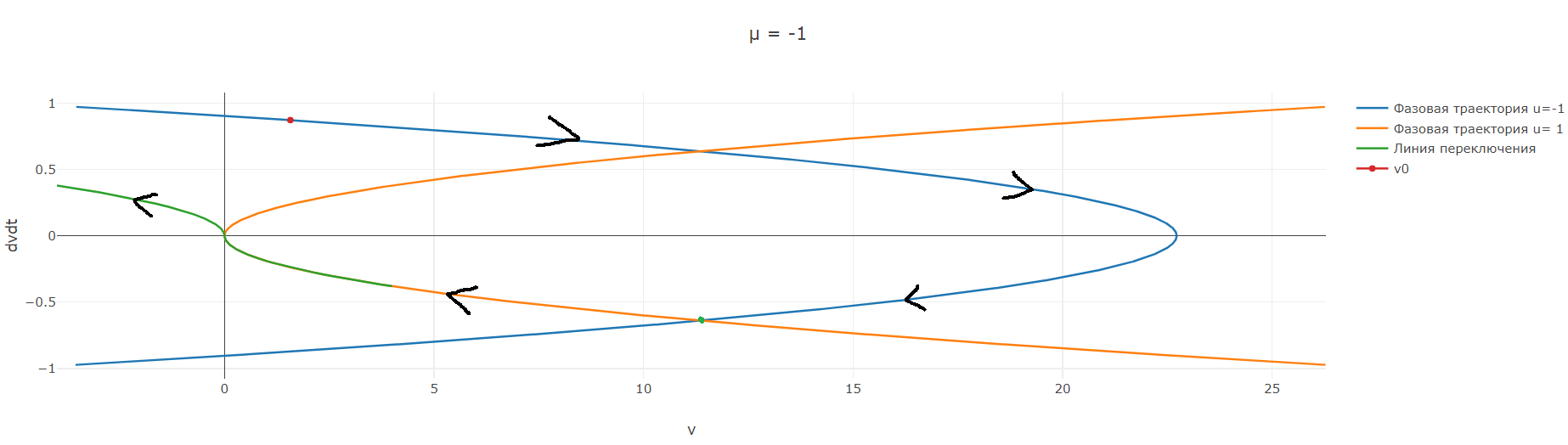


Рисунок 4 – Фазовая диаграмма углового движения спутника при µ = -1

На рисунке 4 красным цветом помечена стартовая точка, а зеленым цветом помечена точка переключения.

1. **Фазовые диаграммы углового движения спутника при оптимальном управлении с описанием процесса их построения.**

Из рисунка 4 следует, что чтобы спутник пришел в точку (0, 0) следует взять µ=-1. На рисунке 5 изображена фазовая диаграмма углового движения спутника при оптимальном управлении.

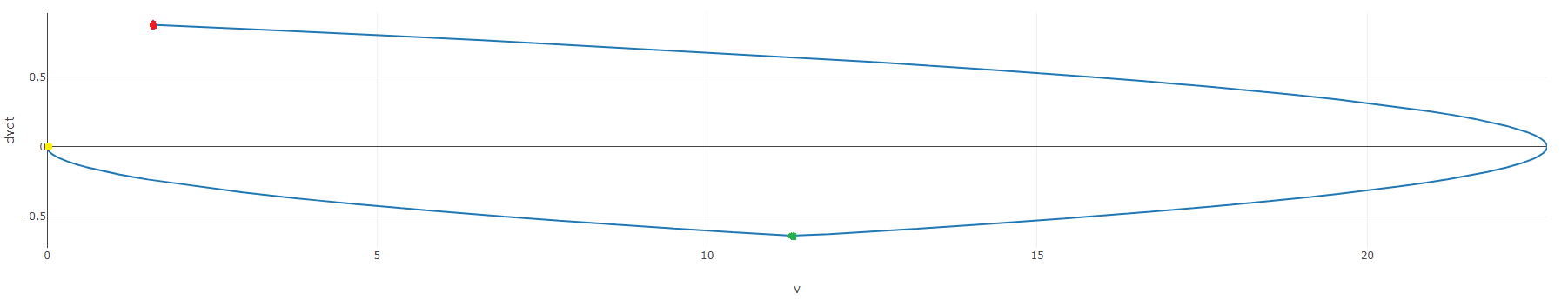


Рисунок 5 – Фазовая диаграмма углового движения спутника при оптимальном управлении

На рисунке 5 красным цветом отмечена стартовая точка, зеленным цветом отмечено переключение, а желтым цветом отмечена конечная точка траектории.

На рисунке 6 изображен график изменения угла от времени.

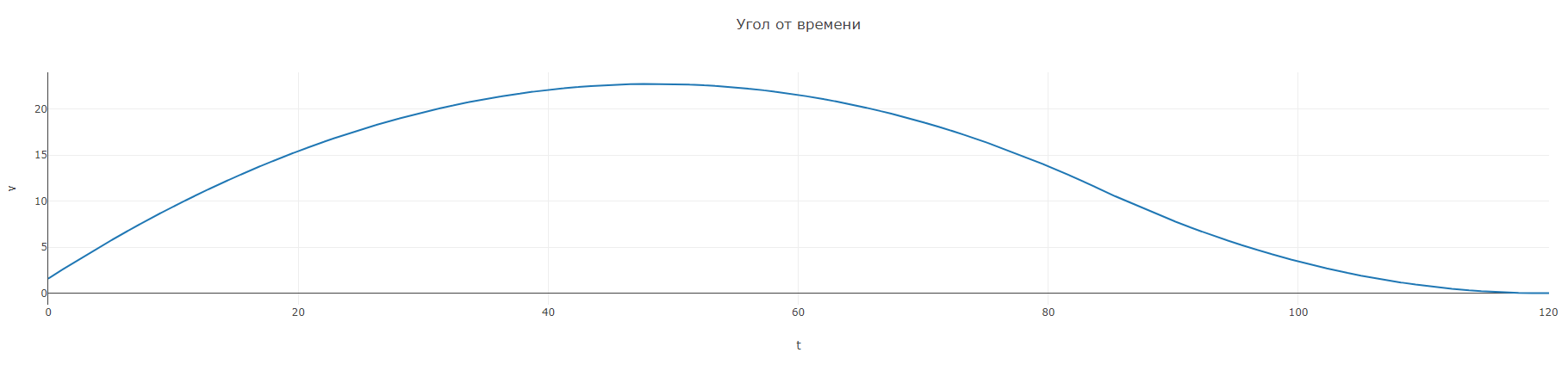


Рисунок 6 – Изменение угла от времени

На рисунке 7 изображен график изменения скорости от времени.

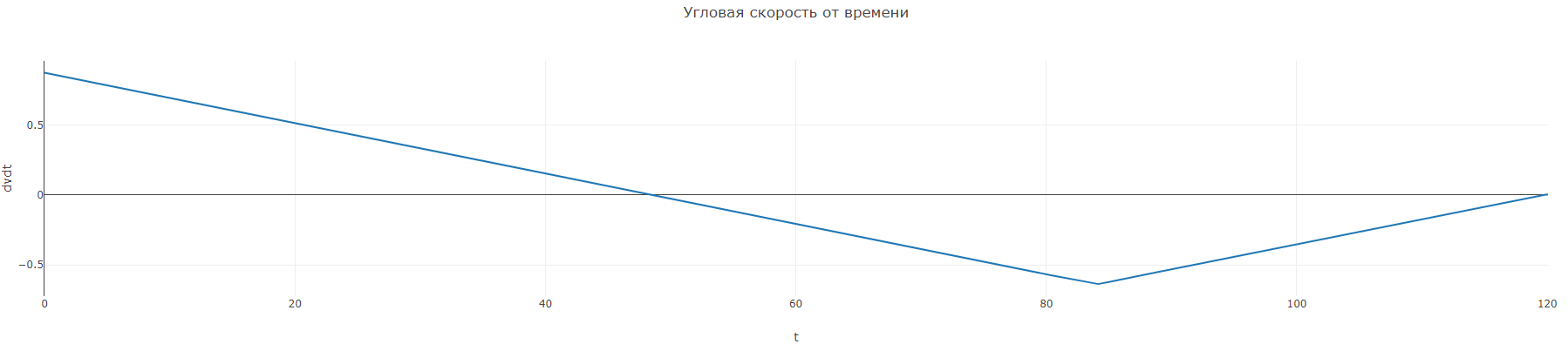


Рисунок 7 – Изменение скорости от времени

1. **Процесс вычисления Δt1 , Δt2.**

Предварительно преобразуем и в радианную меру

= 900 = 1,57 рад

= 50 градс-1 = c-1

Найдём , чтобы понять, где находится точка (выше или ниже линии переключения)

Значит, мы начинаем ниже линии переключения => из «-1» будем переключаться на «+1».

Поскольку переход совершается в два этапа и значения в начале ( = 50 град/с) и в конце ( 0) заданы, то для вычисления времени на каждом этапе необходимо найти в точке D. Эта точка лежит на пересечении кривых, описываемых уравнениями (при начальном )

При пересечении этих парабол получается равенство:

Т.к. переключение происходит в отрицательной области значений

Значение C0 определим исходя из начальных условий. Тогда:

Подстановка C0 и k дает

Для того, чтобы найти возьмем уравнение:

Подставив =:

Получим координаты точки переключения на фазовой плоскости: (; -)

Для расчета выбираем знак “–“ так как переключение происходит в области отрицательных значений.

Найдем формулы для расчета .

В результате интегрирования получаем:

Подставляя , находим время движения на первом и втором участках.

Рассчитываем время движения на первом участке:

Рассчитываем время движения на втором участке:

1. **Описание процесса моделирования системы управления движением спутника.**

Уравнения движения спутника при отсутствии возмущений имеют вид:

По условию задачи момент M должен быть сформулирован так, чтобы спутник переходил из любого отклоненного положения в нормально ориентированное за минимальное время. Поскольку тяга реактивных исполнительных двигателей конечна, то управляющий момент M ограничен

Для решения задачи введём обозначения

Тогда получим

Для моделирования системы управления движением спутника применим метод Рунге-Кутта к данным уравнениям:

Моделируем систему пока спутник не придет в

А когда выполнится условие

Сработает переключение на

Время движения на первом участке:

Время движения на втором участке:

Полученные значения путем моделирования (рисунок 8).

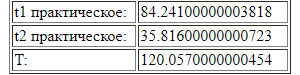


Рисунок 8 – Полученные значения путем моделирования

|  |  |  |  |
| --- | --- | --- | --- |
| Значение | Аналитическое | Смоделированное | Погрешность |
|  |  | 84.24100 | 0.477768 |
|  |  | 35.81600 | 0.386101 |
|  |  | 120.05700 | 0.863869 |

1. **Листинг программы.**
2. <html lang="ru">
3. <head>
4. <meta charset="utf-8">
5. <title>тоу 2</title>
6. <script src="https://cdn.plot.ly/plotly-2.20.0.min.js" charset="utf-8"></script>
7. </head>
8. <body>
9. <div id="plt-1"></div>
10. <div id="plt-2"></div>
11. <div id="plt-3"></div>
12. <div id="plt-4"></div>
13. <div id="info-table" style="display: flex; justify-content: center;"></div>
14. <script>
15. function degreeToRadian(deg) {
16. return (deg \* Math.PI) / 180;
17. }
18. function drawPlots() {
19. J = 2500
20. M = 45
21. v0 = 90
22. dv\_dt0 = 50
23. v0 = degreeToRadian(v0);
24. dv\_dt0 = degreeToRadian(dv\_dt0);
26. const k = M / J;
28. const c0\_mu = (mu) => dv\_dt0 \*\* 2 - mu \* 2 \* k \* v0;
29. const dv\_dt\_i = (k, c1) => -Math.sqrt(2 \* k \* Math.abs(c1)) \* Math.sign(c1);
30. const dv = 0.001;
31. const time\_line = [];
32. const v\_line = [];
33. const dvdt\_line = [];
34. const parabola\_x\_line = [];
35. const primary\_parabola\_y\_line = [];
36. const static\_parabola\_y\_line = [];
37. const switch\_x\_line = [];
38. const switch\_y\_line = [];
39. let selected\_mu;
40. let selected\_mu\_value;
41. let t1\_practice = 0;
42. let t2\_practice = 0;
43. function drawSwitchLine() {
44. let v\_start = -4;
45. let v\_end = -v\_start;
46. while (v\_start <= v\_end) {
47. switch\_y\_line.push(dv\_dt\_i(k, v\_start));
48. switch\_x\_line.push(v\_start);
49. v\_start += dv;
50. }
51. selected\_mu = dv\_dt0 < dv\_dt\_i(k, v0);
52. selected\_mu\_value = selected\_mu ? 1 : -1;
53. }
54. function drawMainParabola() {
55. let v0\_start = -(Math.abs(dv\_dt0) + 0.1);
56. let v0\_end = -v0\_start;
57. while (v0\_start <= v0\_end) {
58. parabola\_x\_line.push(v0\_start);
59. primary\_parabola\_y\_line.push((-selected\_mu\_value \* (c0\_mu(selected\_mu\_value) - v0\_start \*\* 2)) / 2 / k);
60. v0\_start += dv;
61. }
62. }
63. function drawSecondaryParabola() {
64. let v0\_start = -(Math.abs(dv\_dt0) + 0.1);
65. let v0\_end = -v0\_start;
66. while (v0\_start <= v0\_end) {
67. static\_parabola\_y\_line.push((-selected\_mu\_value \* v0\_start \*\* 2) / 2 / k);
68. v0\_start += dv;
69. }
70. }
71. function logic(t, t\_prev, v, v\_prev) {
72. const dvdt = (v - v\_prev) / (t - t\_prev);
73. const dvdt\_p = dv\_dt\_i(k, v);
74. if (dvdt < dvdt\_p || (dvdt === dvdt\_p && dvdt < 0)) {
75. return 1;
76. }
77. return -1;
78. }
79. function model(mu, v, dvdt, dt) {
80. let k1 = dvdt \* dt;
81. let m1 = mu \* k \* dt;
82. let k2 = (dvdt + m1 / 2) \* dt;
83. let m2 = mu \* k \* dt;
84. let k3 = (dvdt + m2 / 2) \* dt;
85. let m3 = mu \* k \* dt;
86. let k4 = (dvdt + m3) \* dt;
87. let m4 = mu \* k \* dt;
88. const new\_v = v + (k1 + 2 \* k2 + 2 \* k3 + k4) / 6;
89. const new\_dvdt = dvdt + (m1 + 2 \* m2 + 2 \* m3 + m4) / 6;
90. return [new\_v, new\_dvdt];
91. }
92. function modelSystem() {
93. const L = 130;
94. const dt = 0.001;
95. let t\_prev = 0;
96. let t = 0;
97. let mu = selected\_mu ? 1 : -1;
98. let v = v0;
99. let v\_prev = v0;
100. let dvdt = dv\_dt0;
101. while (t < L) {
102. time\_line.push(t);
103. v\_line.push(v);
104. dvdt\_line.push(dvdt);
105. mu\_prev = mu;
106. mu = logic(t, t\_prev, v, v\_prev);
107. t\_prev = t;
108. v\_prev = v;
109. [v, dvdt] = model(mu, v, dvdt, dt);
110. if (Math.abs(v) <= 1e-3 && Math.abs(dvdt) <= 1e-3) {
111. break;
112. }
113. mu === selected\_mu\_value ? (t1\_practice += dt) : (t2\_practice += dt);
114. t += dt;
115. }
116. }
117. function drawTable() {
118. const v1 = c0\_mu(selected\_mu\_value) / (2 \* k);
119. const \_dv\_dt1 = -Math.sqrt(k \* Math.abs(v1)) \* Math.sign(v1);
120. const t1 = Math.abs((\_dv\_dt1 - dv\_dt0 \* -selected\_mu\_value) / k);
121. const t2 = Math.abs(-\_dv\_dt1 / k);
122. let table = document.createElement('table')
123. let cellJ = "<tr><td>J:</td><td>"+J+"</td></tr>"
124. let cellM = "<tr><td>M:</td><td>"+M+"</td></tr>"
125. let cellK = "<tr><td>k:</td><td>"+k+"</td></tr>"
126. let cellC0Plus = "<tr><td>c0\_mu, mu = +1:</td><td>"+c0\_mu(+1)+"</td></tr>"
127. let cellC0Min= "<tr><td>c0\_mu, mu = -1:</td><td>"+c0\_mu(-1)+"</td></tr>"
128. let cellV0 = "<tr><td>v0:</td><td>"+v0+"</td></tr>"
129. let cellV1 = "<tr><td>v1:</td><td>"+v1/2+"</td></tr>"
130. let cellDV0 = "<tr><td>dv\_dt0:</td><td>"+dv\_dt0+"</td></tr>"
131. let cellDV1 = "<tr><td>dv\_dt1:</td><td>"+\_dv\_dt1+"</td></tr>"
132. let cellT1A = "<tr><td>t1 теоритическое:</td><td>"+t1+"</td></tr>"
133. let cellT2A = "<tr><td>t2 теоритическое:</td><td>"+t2+"</td></tr>"
134. let cellT1T = "<tr><td>t1 практическое:</td><td>"+t1\_practice+"</td></tr>"
135. let cellT2T = "<tr><td>t2 практическое:</td><td>"+t2\_practice+"</td></tr>"
136. let cellT = "<tr><td>T:</td><td>"+(t1\_practice + t2\_practice)+"</td></tr>"
137. table.innerHTML = cellJ+cellM+cellK+cellC0Plus+cellC0Min+cellV0+cellV1+cellDV0+cellDV1+cellT1A+cellT2A+cellT1T+cellT2T+cellT;
138. table.setAttribute('border', 1)
139. document.getElementById('info-table').appendChild(table)
140. }
141. function outputPlots() {
142. const start\_point = {
143. name: 'v0',
144. mode: 'trace',
145. x: [v0],
146. y: [dv\_dt0],
147. };
148. const faza\_legend = {
149. title: `μ = ${selected\_mu\_value}`,
150. xaxis: {
151. title: 'v',
152. },
153. yaxis: {
154. title: 'dvdt',
155. },
156. };
157. const v\_t\_legend = {
158. title: 'Угол от времени',
159. xaxis: {
160. title: 't',
161. },
162. yaxis: {
163. title: 'v',
164. },
165. };
166. const dvdt\_t\_legend = {
167. title: 'Угловая скорость от времени',
168. xaxis: {
169. title: 't',
170. },
171. yaxis: {
172. title: 'dvdt',
173. },
174. };
175. const dvdt\_v\_legend = {
176. title: '',
177. xaxis: {
178. title: 'v',
179. },
180. yaxis: {
181. title: 'dvdt',
182. },
183. };
184. const primary\_parabola\_plot = {
185. name: 'Фазовая траектория u=-1',
186. mode: 'lines',
187. x: primary\_parabola\_y\_line,
188. y: parabola\_x\_line,
189. };
190. const static\_parabola\_plot = {
191. name: 'Фазовая траектория u= 1',
192. mode: 'lines',
193. x: static\_parabola\_y\_line,
194. y: parabola\_x\_line,
195. };
196. const v\_t\_plot = {
197. name: 'угол',
198. mode: 'lines',
199. x: time\_line,
200. y: v\_line,
201. };
202. const dvdt\_t\_plot = {
203. name: 'скорость',
204. mode: 'lines',
205. x: time\_line,
206. y: dvdt\_line,
207. };
208. const dvdt\_v\_plot = {
209. name: 'скорость',
210. mode: 'lines',
211. x: v\_line,
212. y: dvdt\_line,
213. };
214. const switch\_plot = {
215. name: 'Линия переключения',
216. mode: 'lines',
217. x: switch\_x\_line,
218. y: switch\_y\_line,
219. };
220. Plotly.newPlot('plt-1', [primary\_parabola\_plot, static\_parabola\_plot, switch\_plot, start\_point], faza\_legend);
221. Plotly.newPlot('plt-2', [v\_t\_plot], v\_t\_legend);
222. Plotly.newPlot('plt-3', [dvdt\_t\_plot], dvdt\_t\_legend);
223. Plotly.newPlot('plt-4', [dvdt\_v\_plot], dvdt\_v\_legend);
224. }
225. drawSwitchLine();
226. drawSecondaryParabola();
227. drawMainParabola();
228. modelSystem();
229. drawTable();
230. outputPlots();
231. }
232. drawPlots();
233. </script>
234. <script src="https://cdn.plot.ly/plotly-2.20.0.min.js" charset="utf-8"></script>
235. </body>
236. </html>