Министерство образования и науки Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И  
ОПТИКИ

Факультет систем управления и робототехники

**Отчет по лабораторной работе №7 «Разработка системы управления для неполноприводного робота»**

**по дисциплине «Введение в профессиональную деятельность»**

Выполнили: студенты гр. **R3142**

Рогозина В. С.

Петрищев А. С.

Подзоров А.В.

Лоскутова И.В.

Преподаватель:

Перегудин А. А.,

ассистент фак.СУиР

Санкт-Петербург   
2022

1. Цель работы

Получить опыт составления модели вход-состояние-выход для относительно сложного электромеханического устройства. Познакомиться с понятием П-регулятора состояния, расчетом его коэффициентов и принципами работы неполноприводных роботов, находящихся под его управлением.

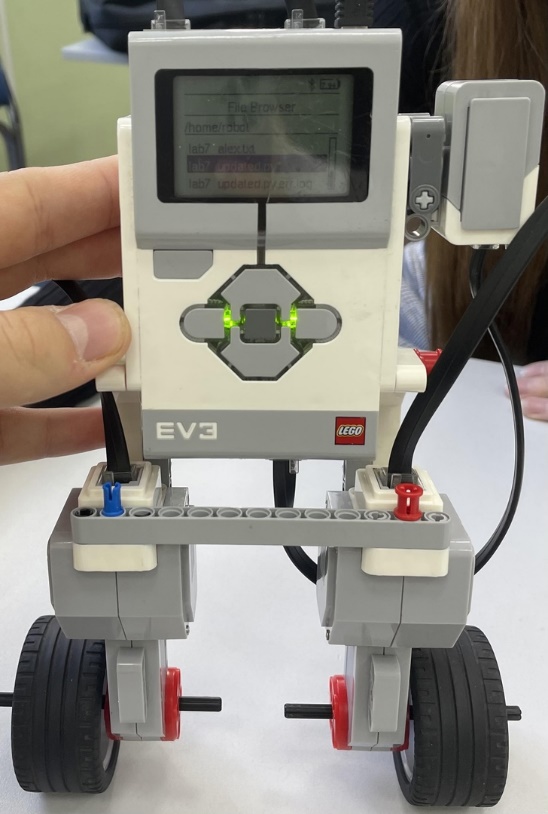
1. Материалы работы

2.0. Сборка робота.

Наш робот был собран аналогично методичке. Конструкция робота полностью соответствует представленному механизму.

2.1. Фотографии собранного робота-segway

Изображение выглядит как устройство, счетчик, фрезерный станок

Автоматически созданное описание 

2.2. Параметры собранного робота

Параметры робота-segway представлены в таблице:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | M, кг | J, | L,м | R,м |
| колесо (х2) | 0,023 | 0,000317 | 0,05 | 0,028 |
| мотор (х2) | 0,08 | 0,000359 | 0,04 | - |
| брик | 0,286 | 0,000374 | 0 | - |
| гиро сенсор | 0,014 | 0,000359 | 0,06 | - |
| всё тело | 0,506 | 0,000789 | - | - |

М – масса детали, которая была посчитана с помощью кухонных весов.

J – момент инерции, который был посчитан по теореме Штейнера

L – расстояние от центра детали до центра масс, которое было измерено с помощью линейки.

R – радиус, который был измерен с помощью линейки

Так же в данной лабораторной работе нам понадобились дополнительные параметры мотора и брика:

km = ke = 0.488

R = 8.2 Om

U = 8.02 В

2.3. Математическая модель робота

2.3.1. Линейная модель

**Математическая модель**

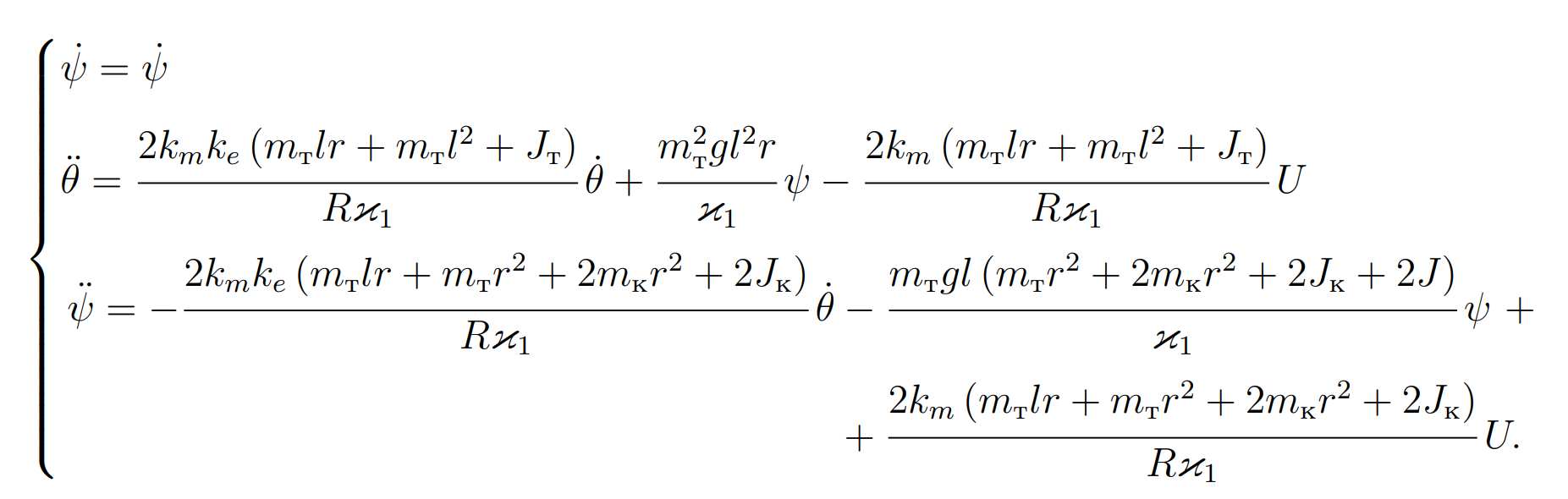
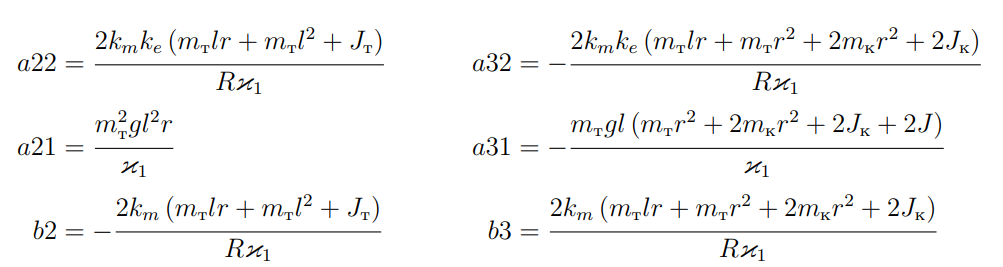
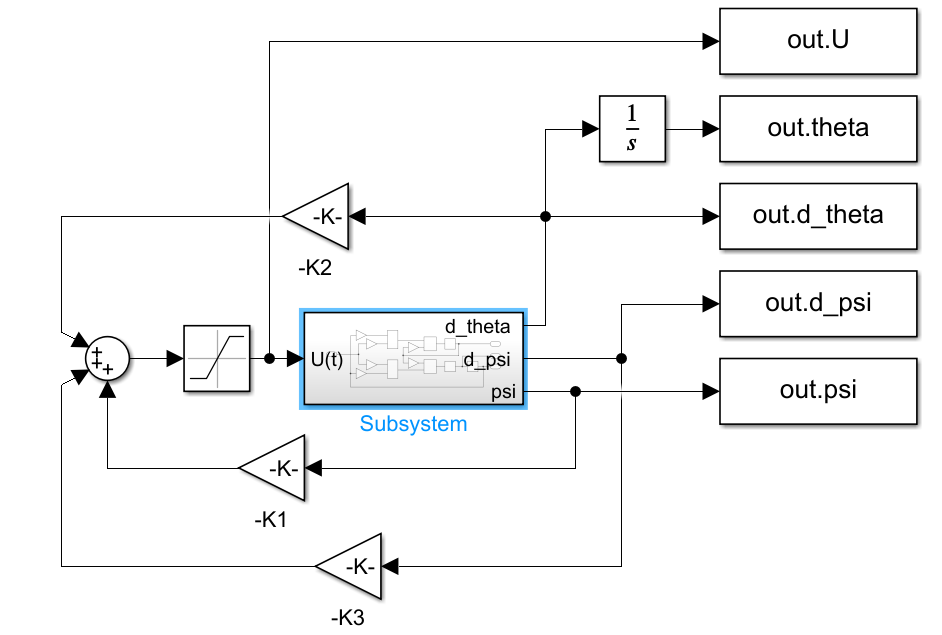
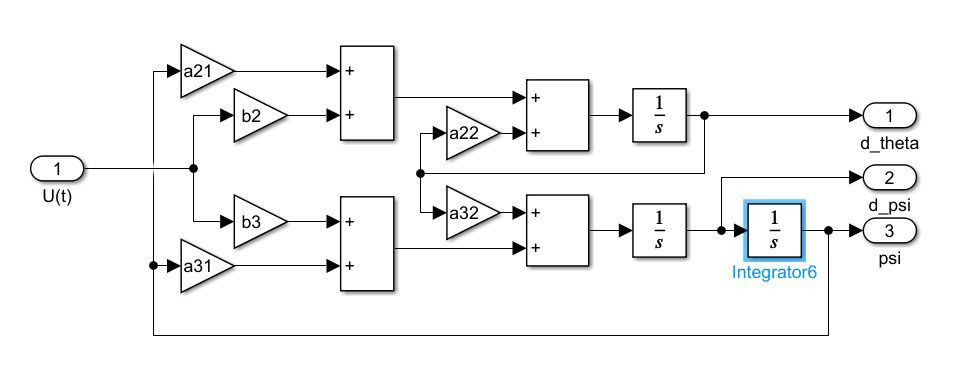
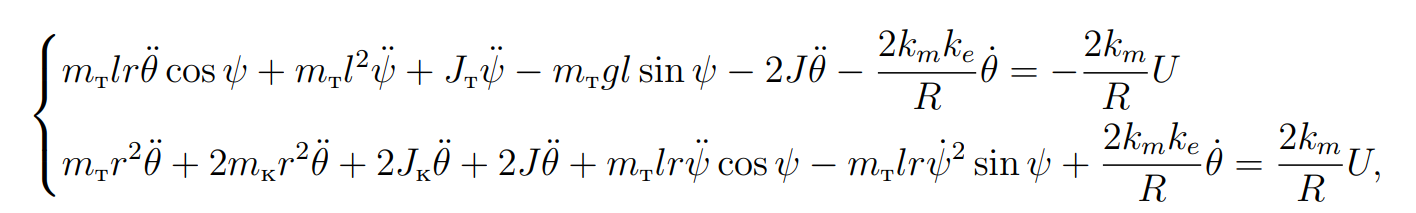
Линейная модель робота была дана нам в методичке. Она была получена с помощью линеаризации полной нелинейной модели. Уравнения, по которым строится линейная модель робота:  
  
**Модель робота в Simulink**  
  
Для нее используются «коэффициенты», которые стоят перед основными параметрами модели. Они представлены ниже:  


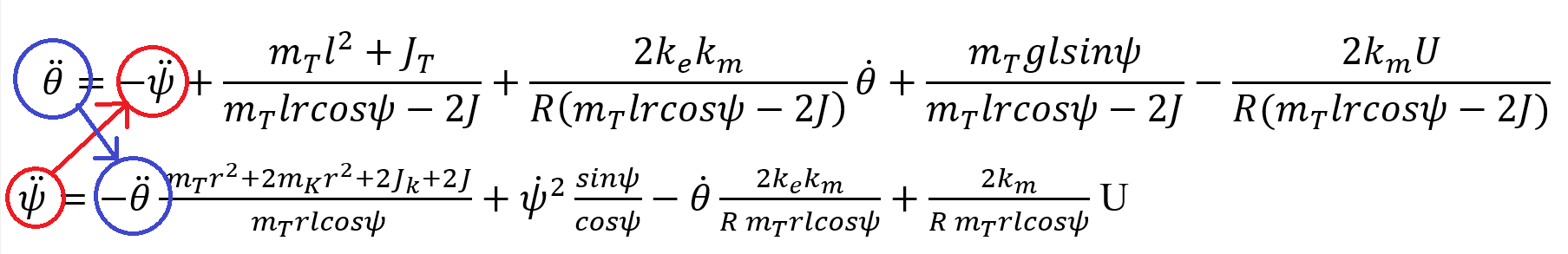
Схема моделирования:  
  
Блок Subsystem:  


2.3.2. Нелинейная модель  
**Математическая модель**

Полная нелинейная математическая модель для нашего робота-сигвея была выведена нами с помощью системы уравнений данных нам в методических пособиях:   


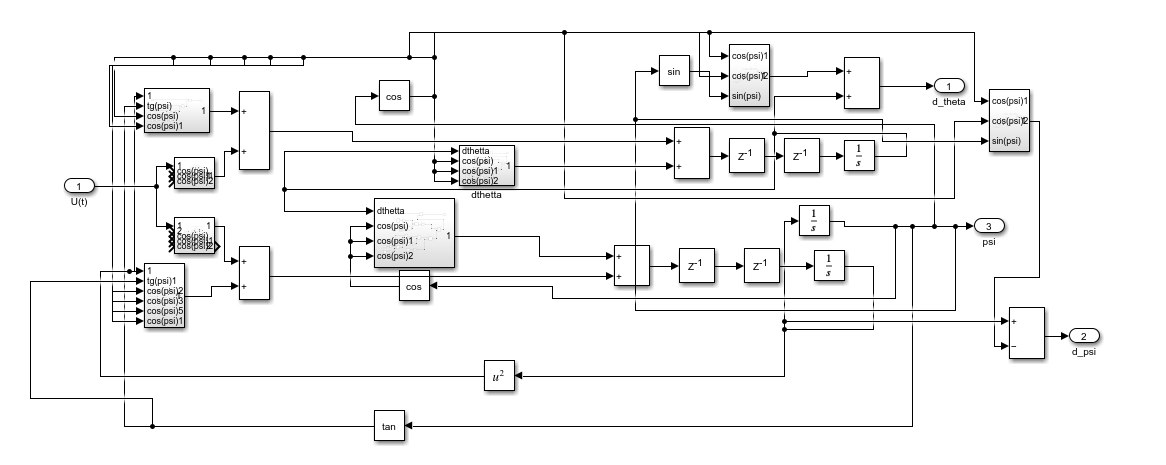
Мы выразили из этих уравнений.

U

Затем, мы подставили одно уравнение в другое  


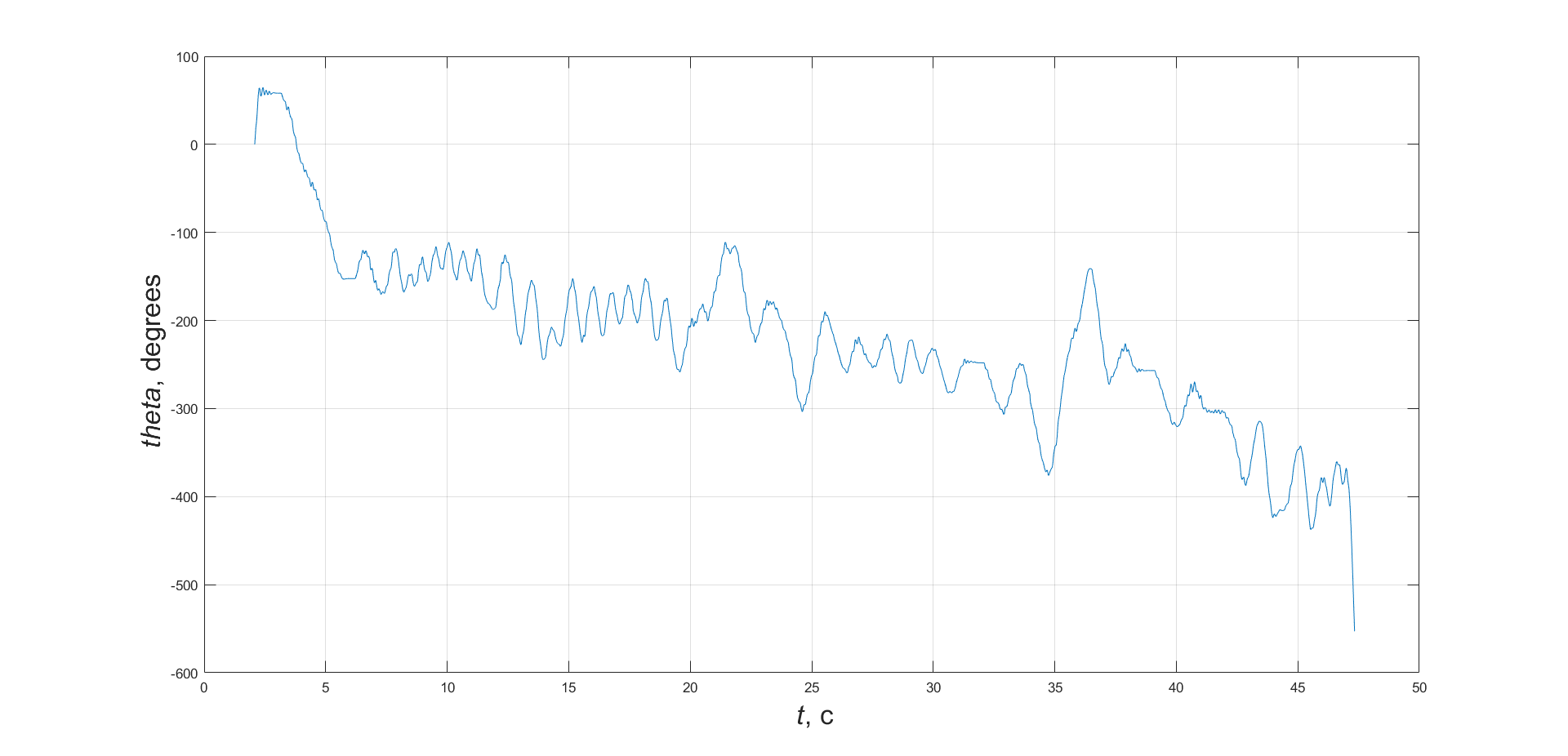
И в итоге, сократив всё, что можно было сократить, мы получили уравнения полной нелинейной математической модели нашего сигвея:­­­­

**Модель робота Simulink**

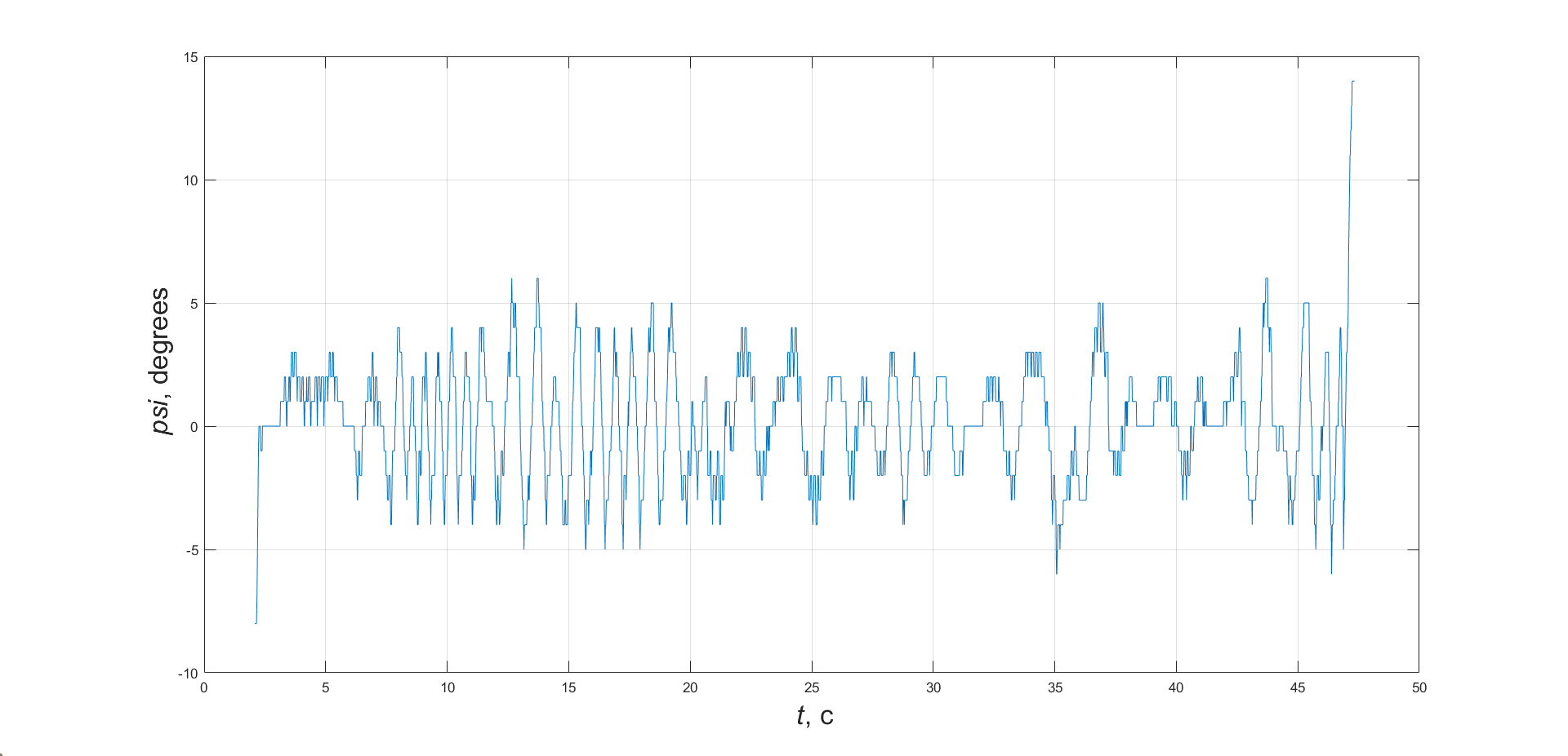


В данной модели в суперблоках находятся нелинейные функции, которые в уравнении стоят перед каждым из параметров, описывающих движение сигвея.­­­

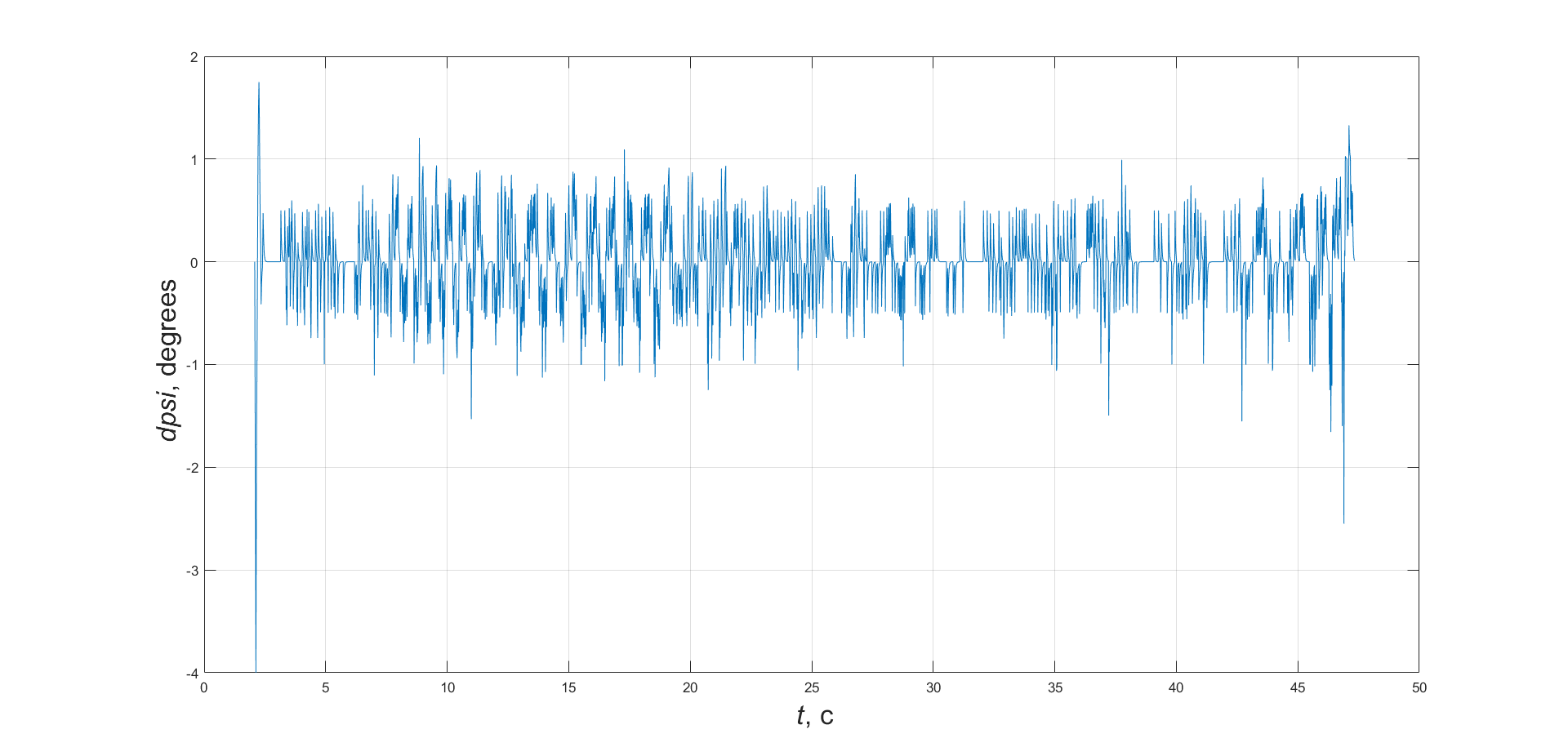
2.4. Графики эксперимента, полученные в ходе выполнения лабораторной работы



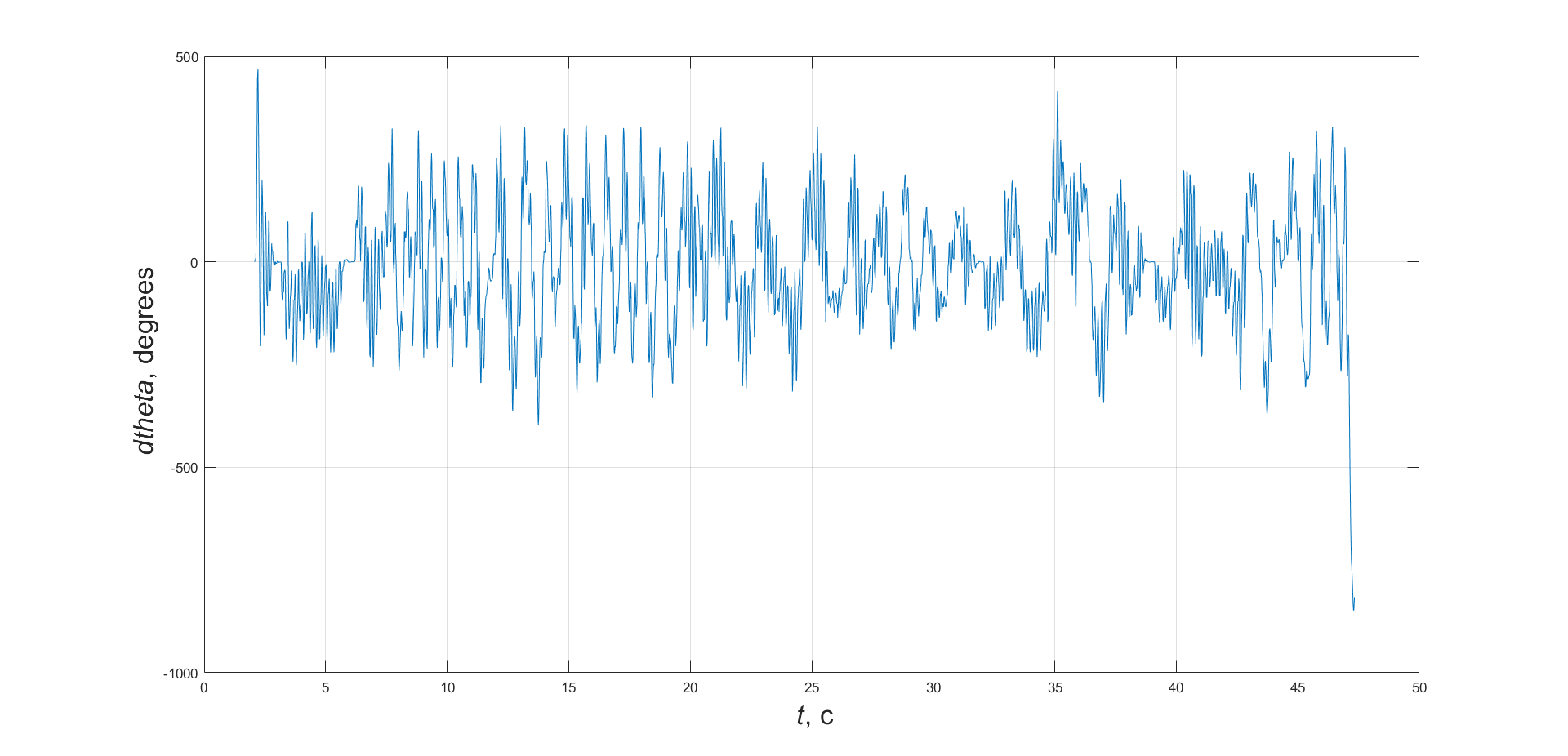
*Рисунок 1. График зависимости угла поворота моторов от времени t*

**

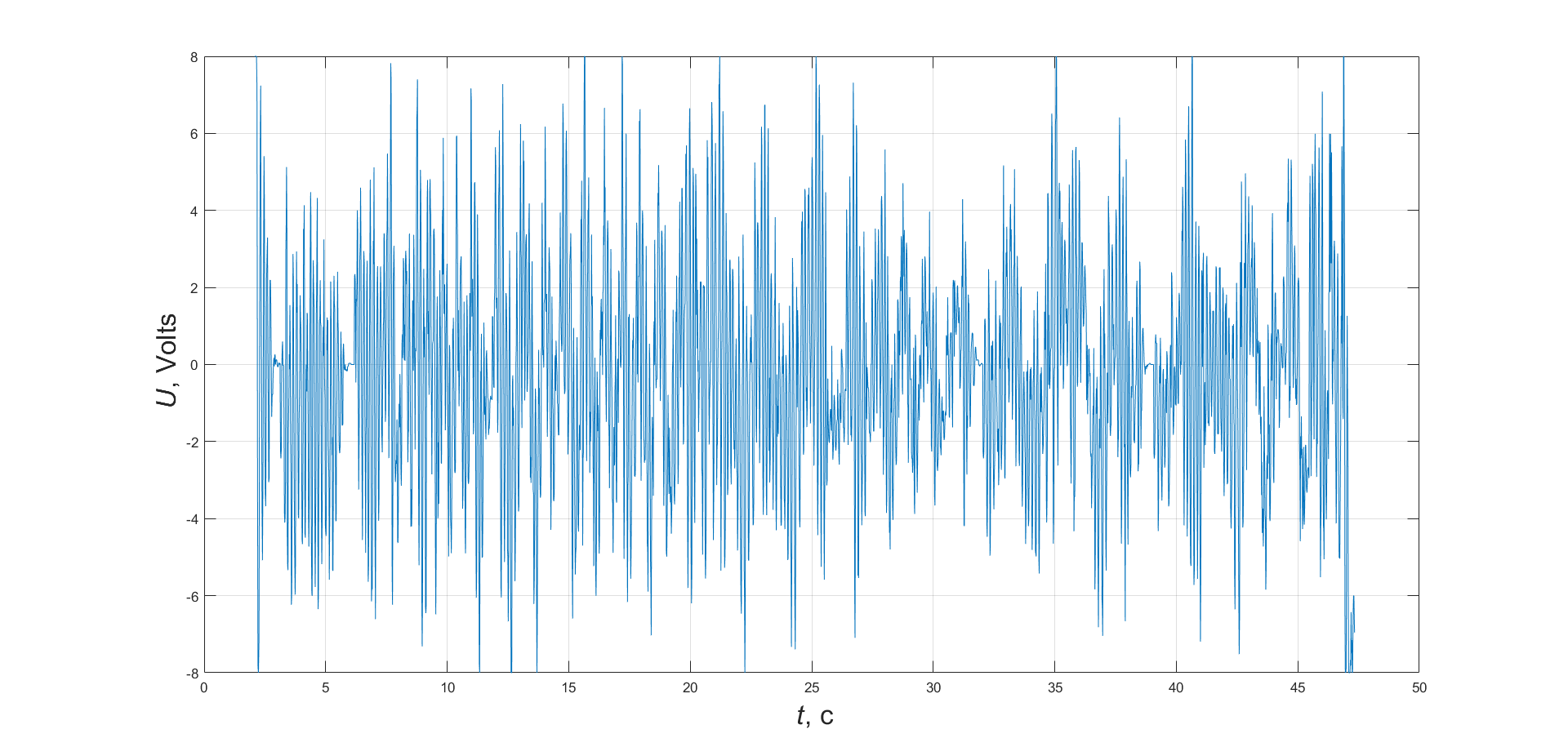
*Рисунок 2. График зависимости угла отклонения от времени t*

**

*Рисунок 3. График зависимости от времени t*

**

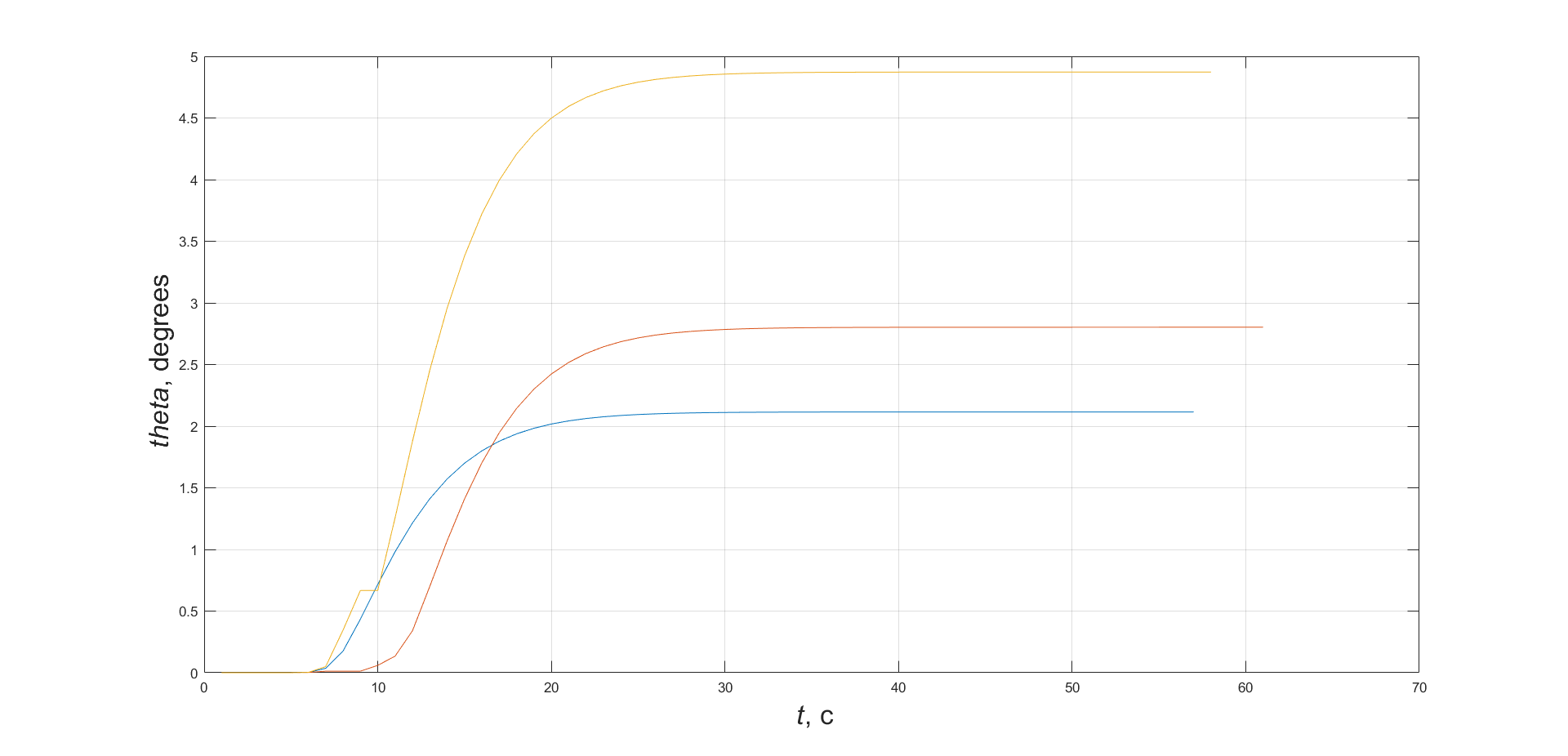
*Рисунок 4. График зависимости от времени t*

**

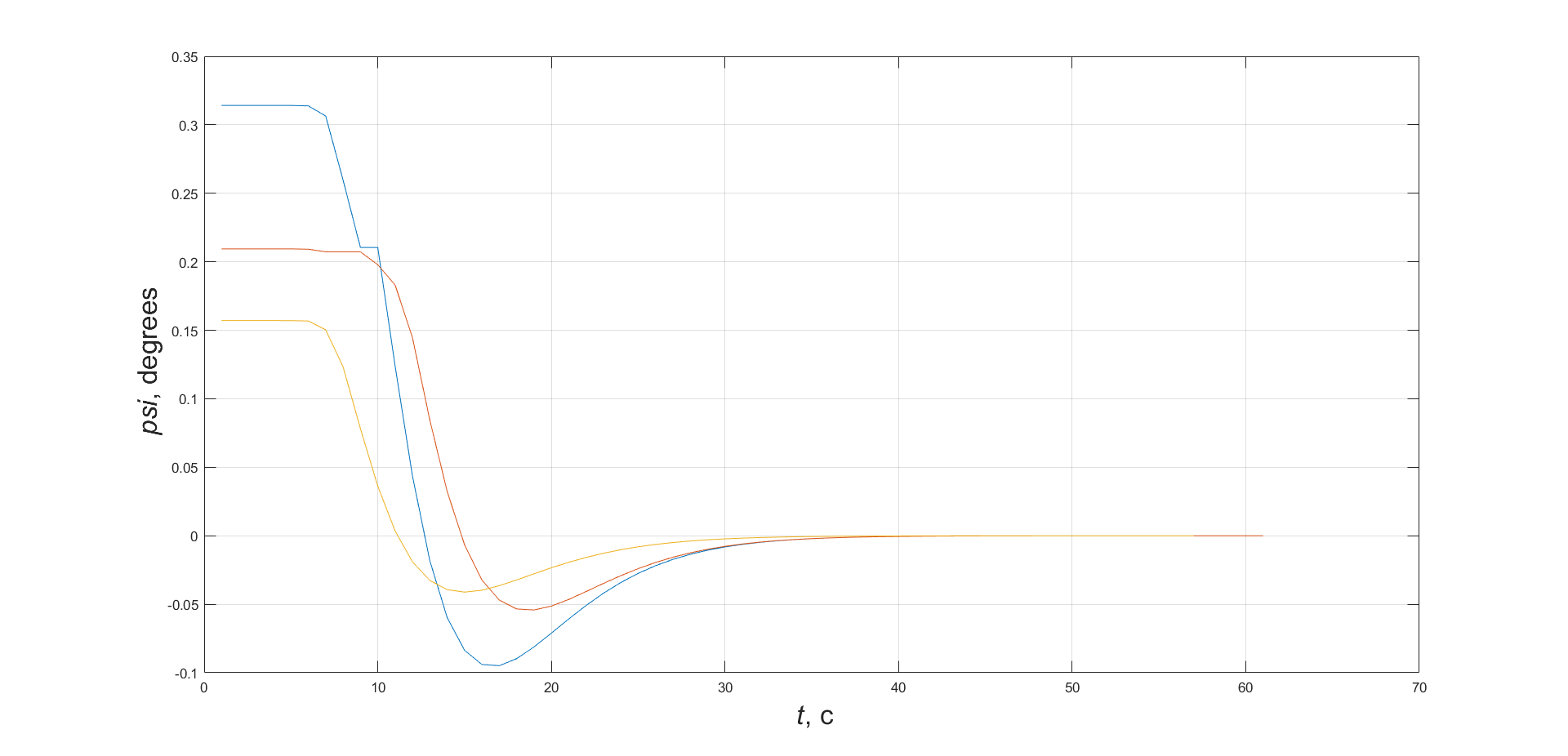
*Рисунок 5. График зависимости напряжения U от времени t*

2.6. Графики модели

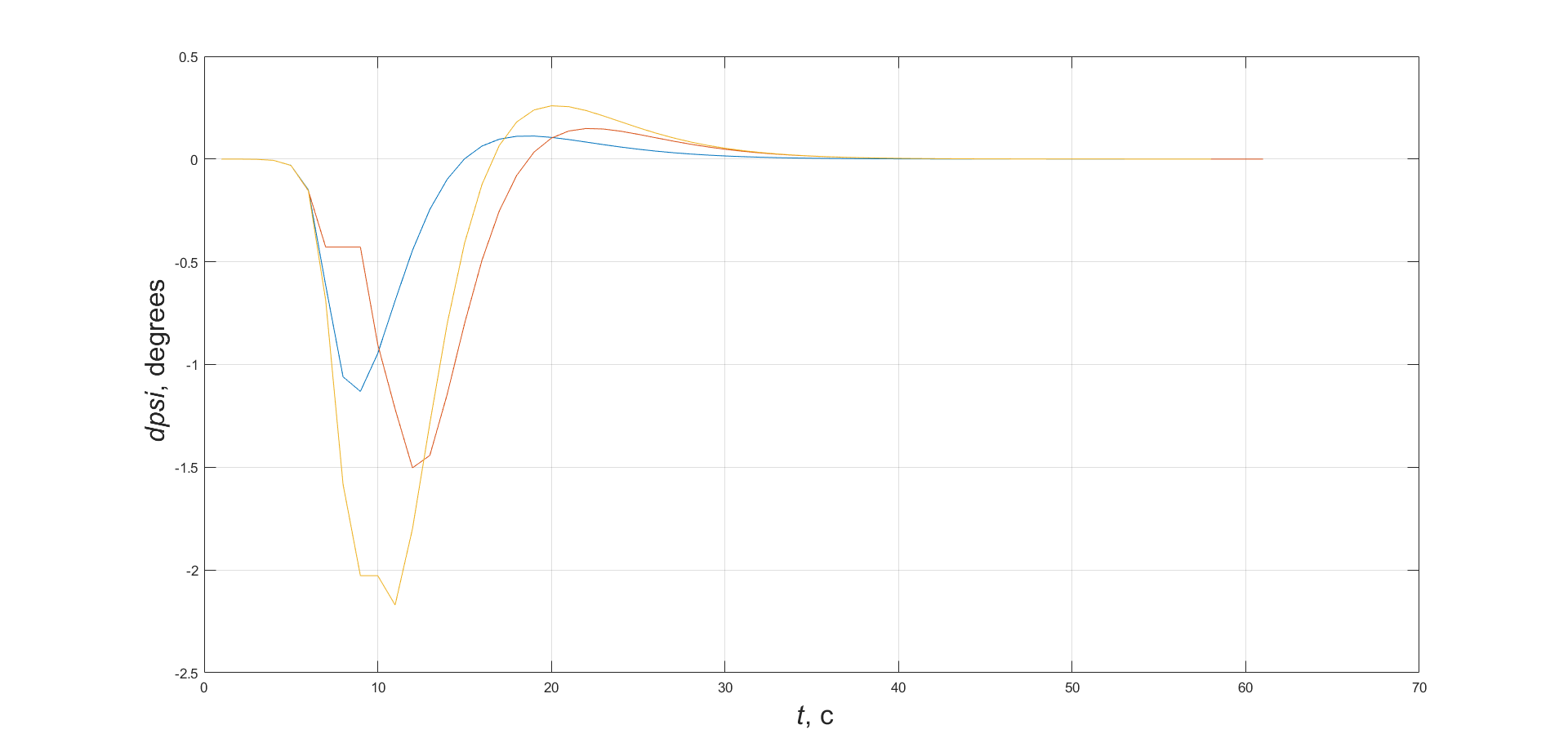
2.6.1. Линейная модель



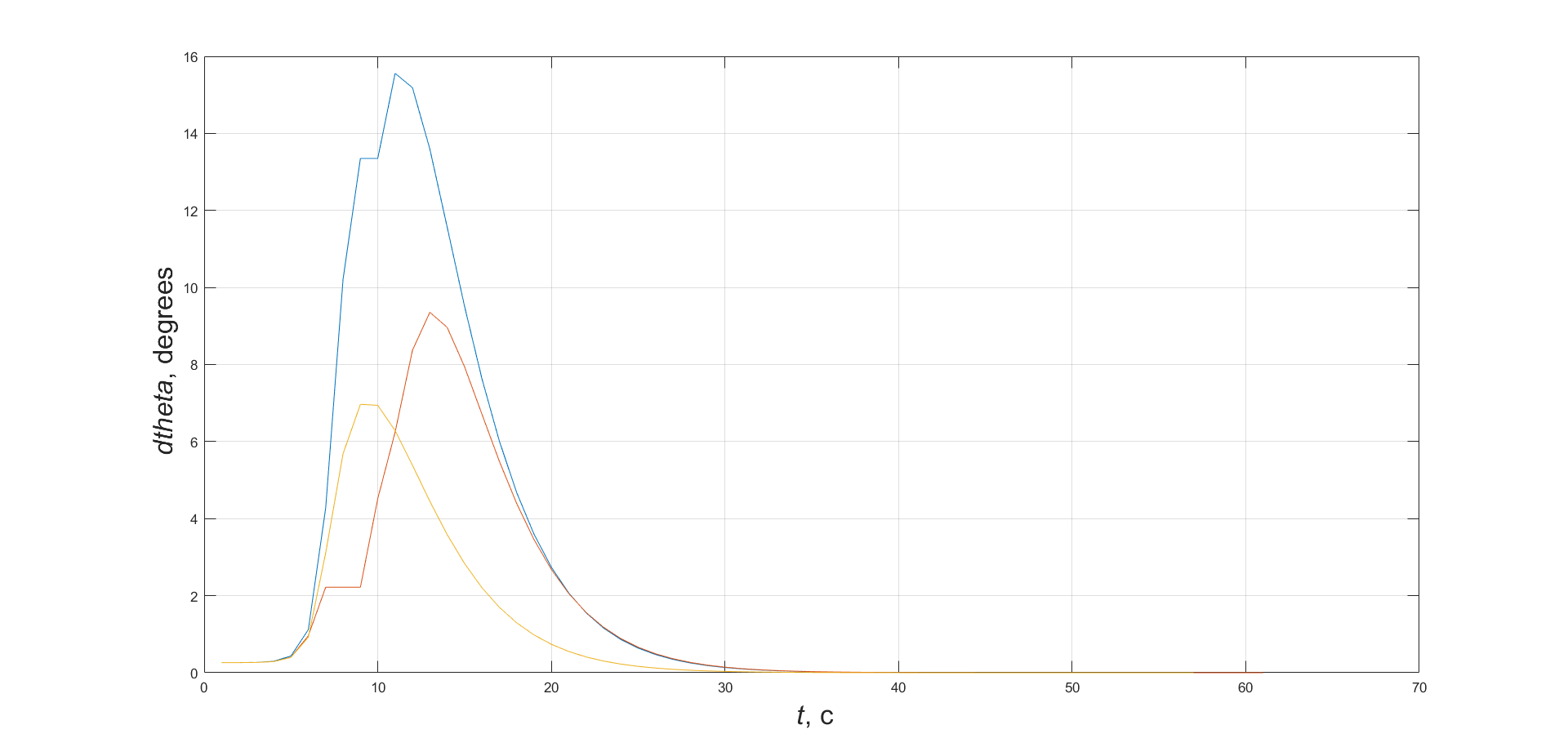
*Рисунок 6. График модели, отражающей зависимости угла поворота моторов от времени t при разных входных параметрах*

**

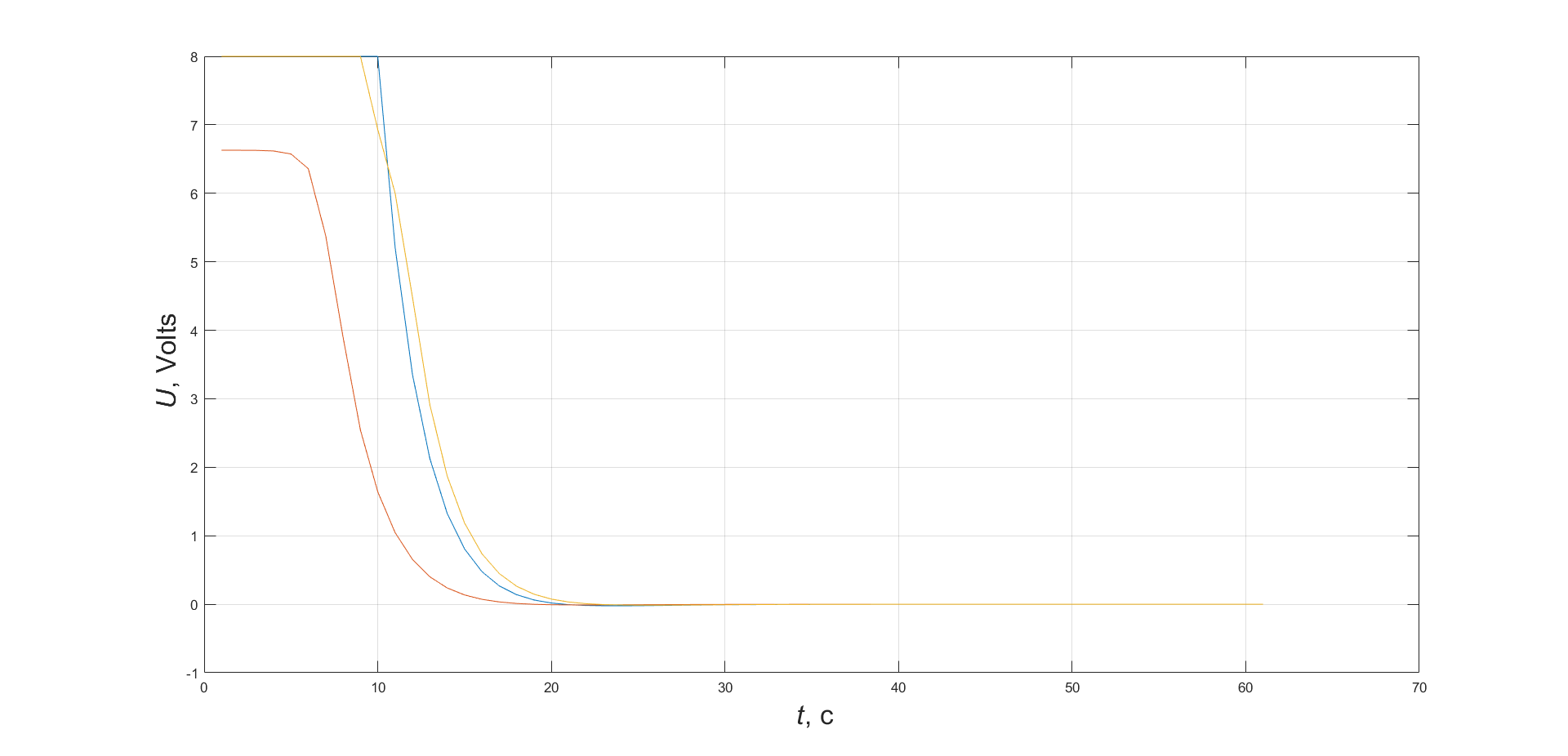
*Рисунок 7. График модели, отражающей зависимости угла отклонения от времени t при разных входных параметрах*

**

*Рисунок 8. График модели, отражающей зависимости от времени t при разных входных параметрах*

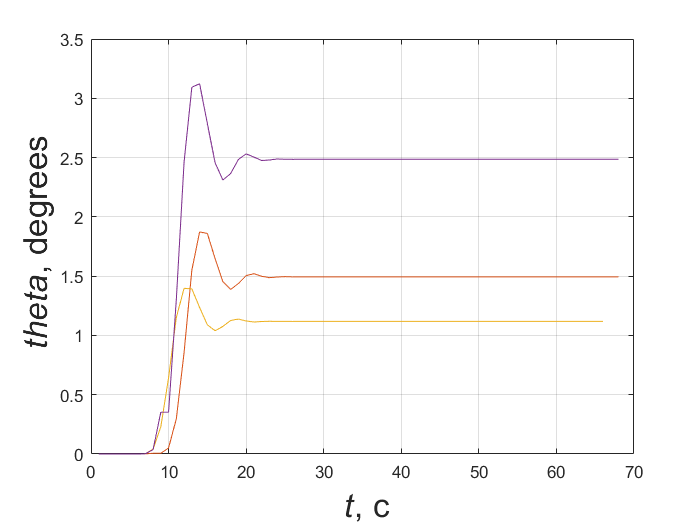
**

*Рисунок 9. График модели, отражающей зависимости от времени t при разных входных параметрах*

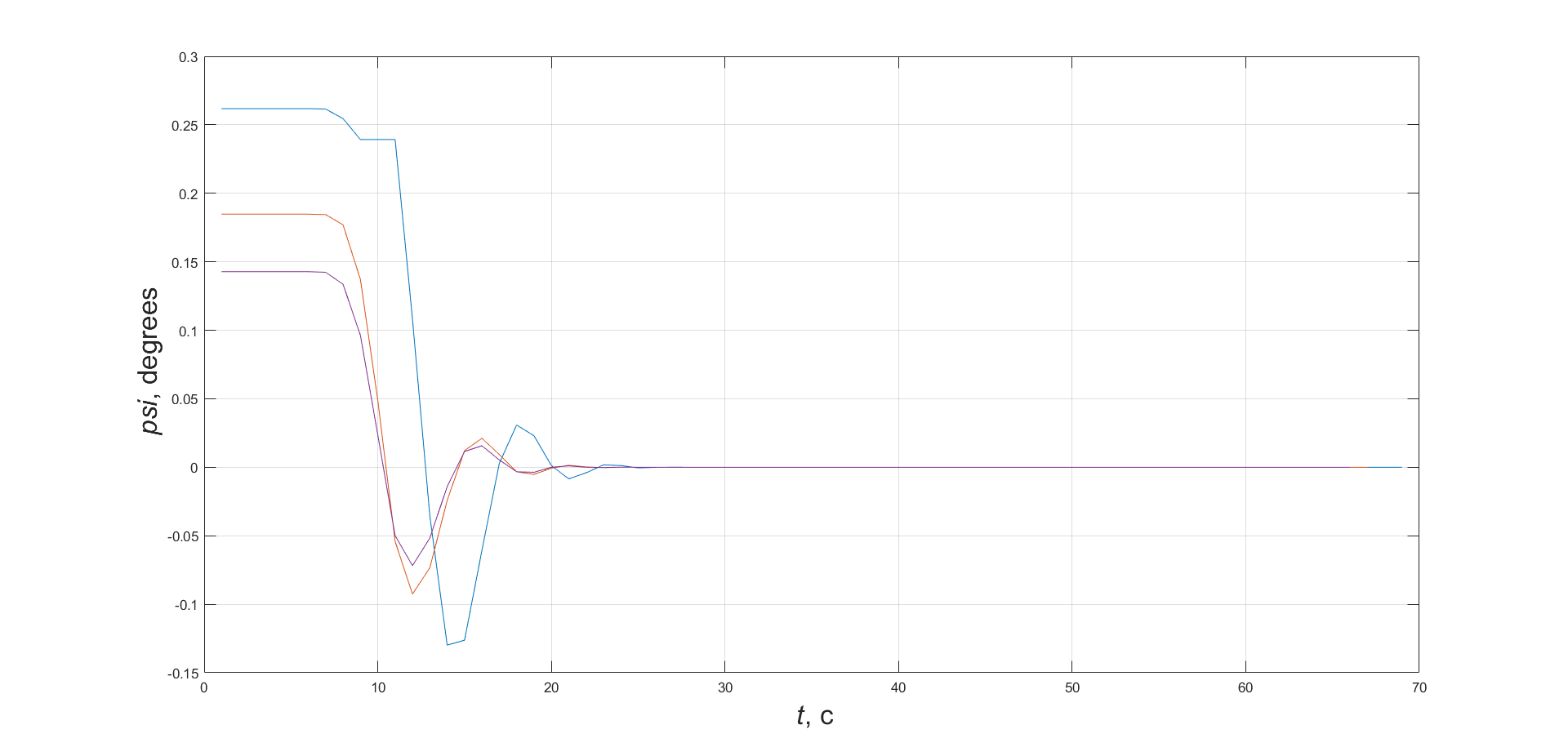
**

*Рисунок 10. График модели, отражающей зависимости напряжения U от времени t при разных входных параметрах*

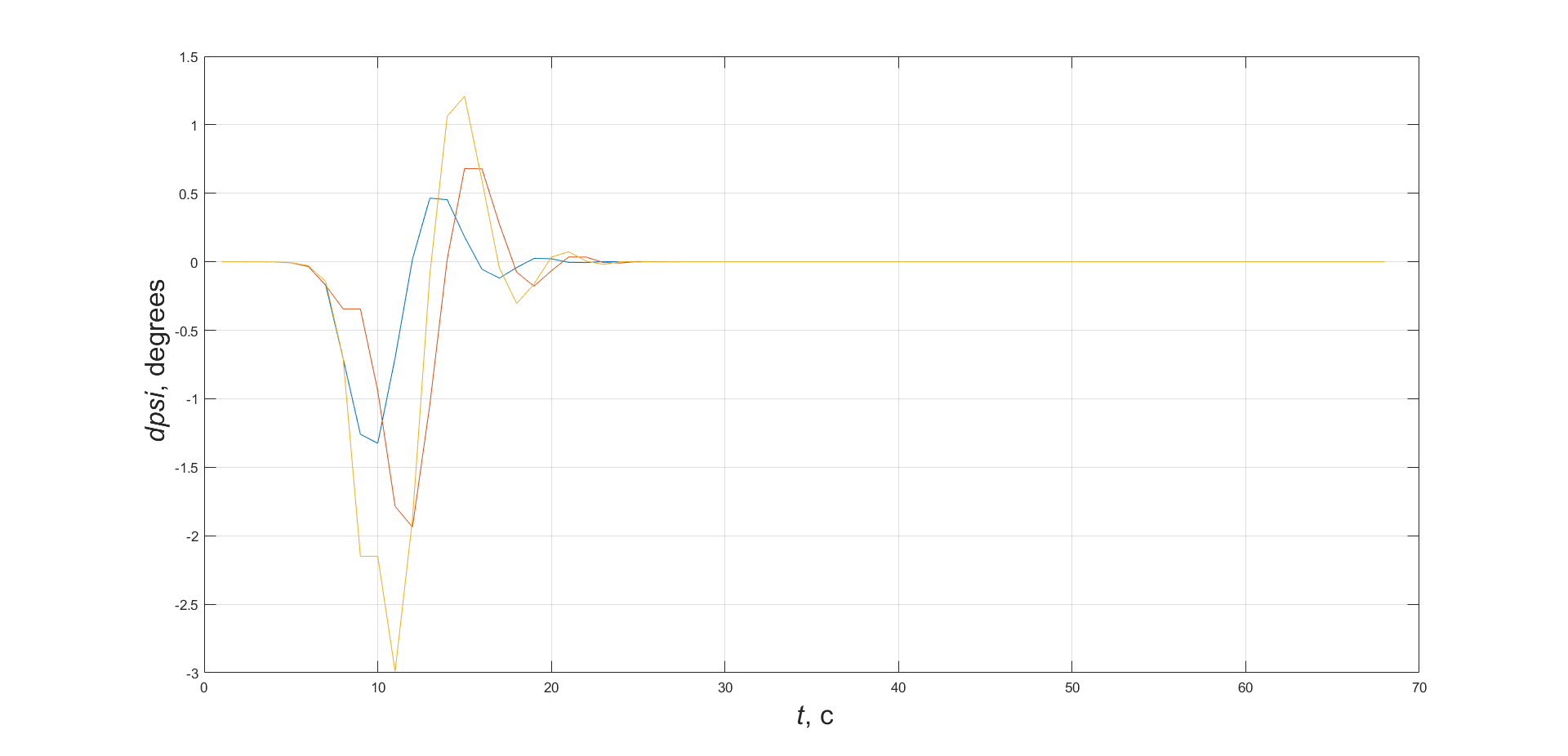
2.6.2. Нелинейная модель



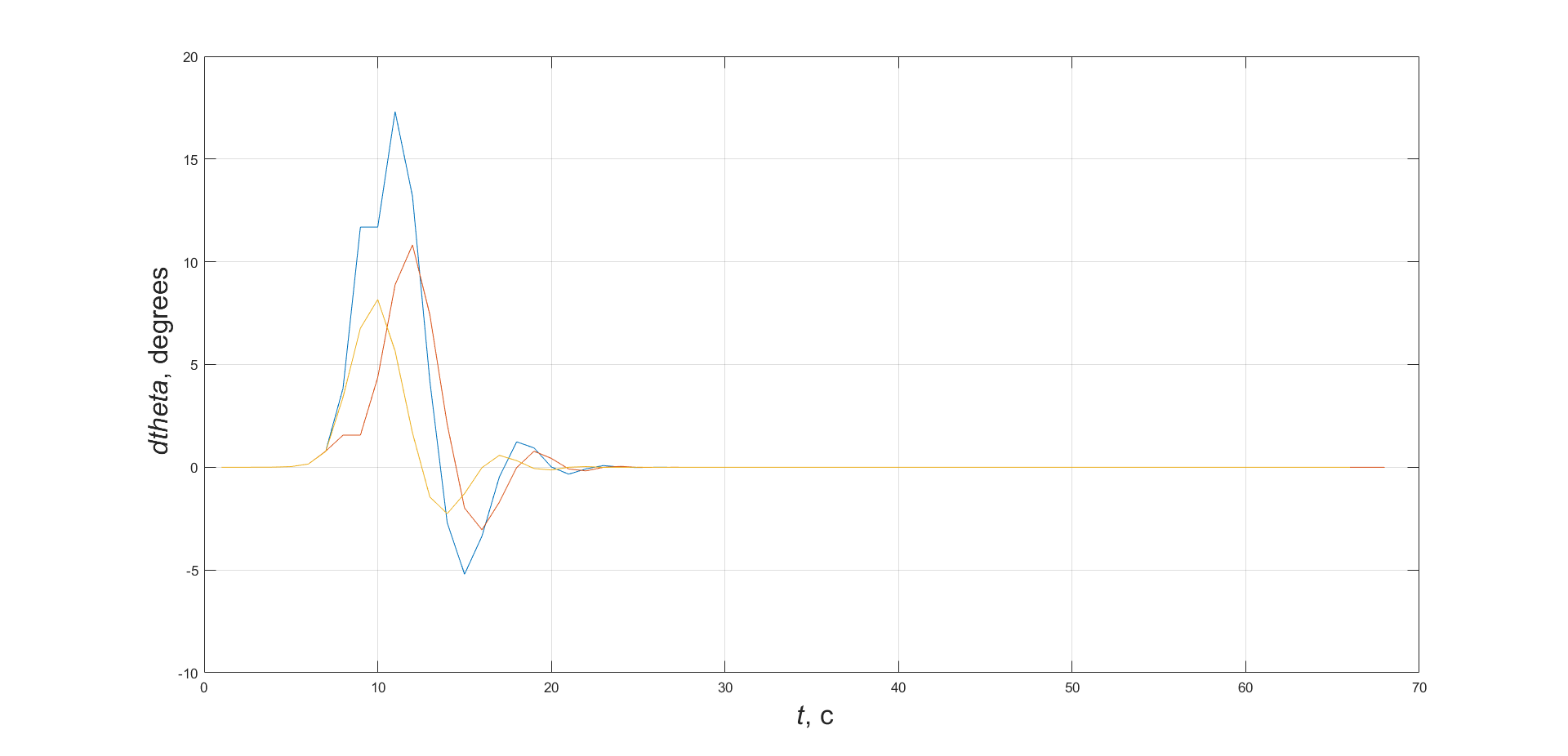
*Рисунок 11. График модели, отражающей зависимости угла поворота моторов от времени t при разных входных параметрах*

**

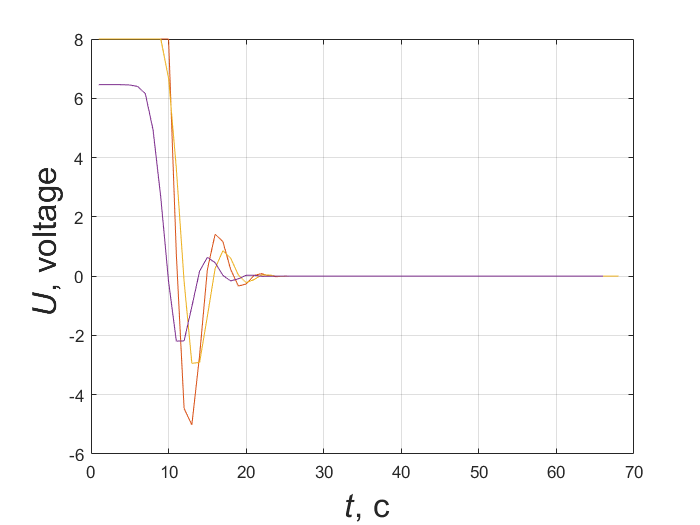
*Рисунок 12. График модели, отражающей зависимости угла отклонения от времени t при разных входных параметрах*

**

*Рисунок 13. График модели, отражающей зависимости от времени t при разных входных параметрах*

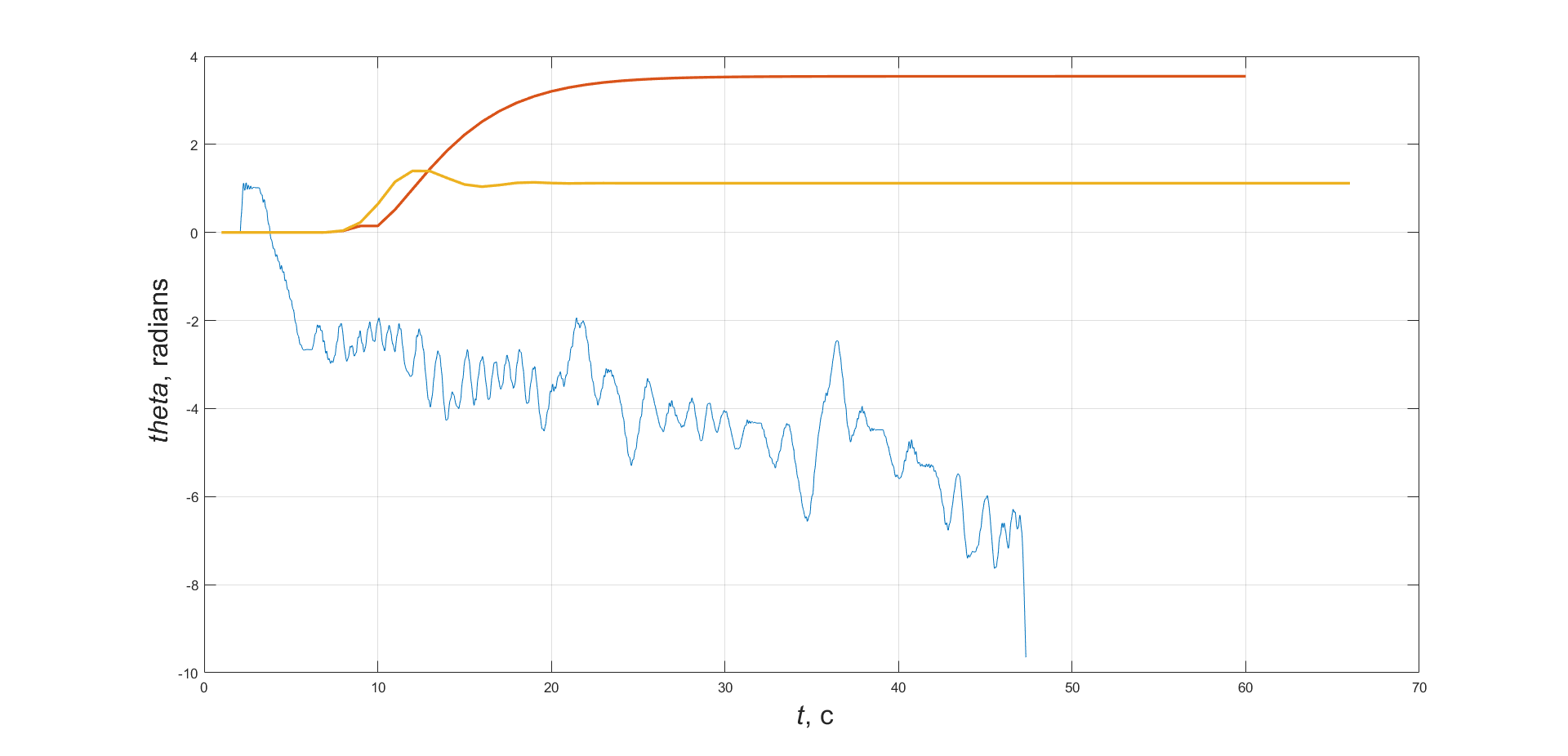
**

*Рисунок 14. График модели, отражающей зависимости от времени t при разных входных параметрах*

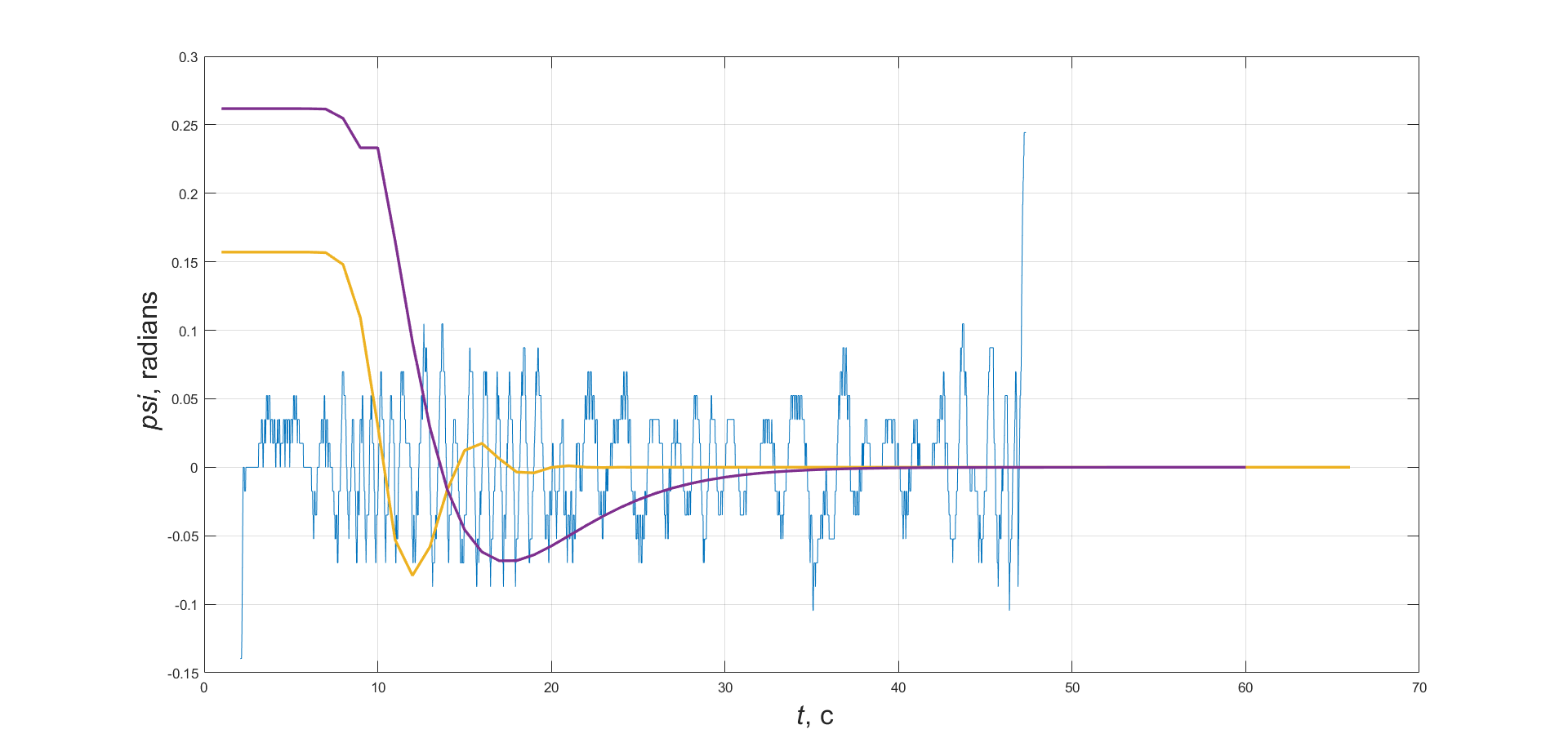
**

*Рисунок 15. График модели, отражающей зависимости напряжения U от времени t при разных входных параметрах*

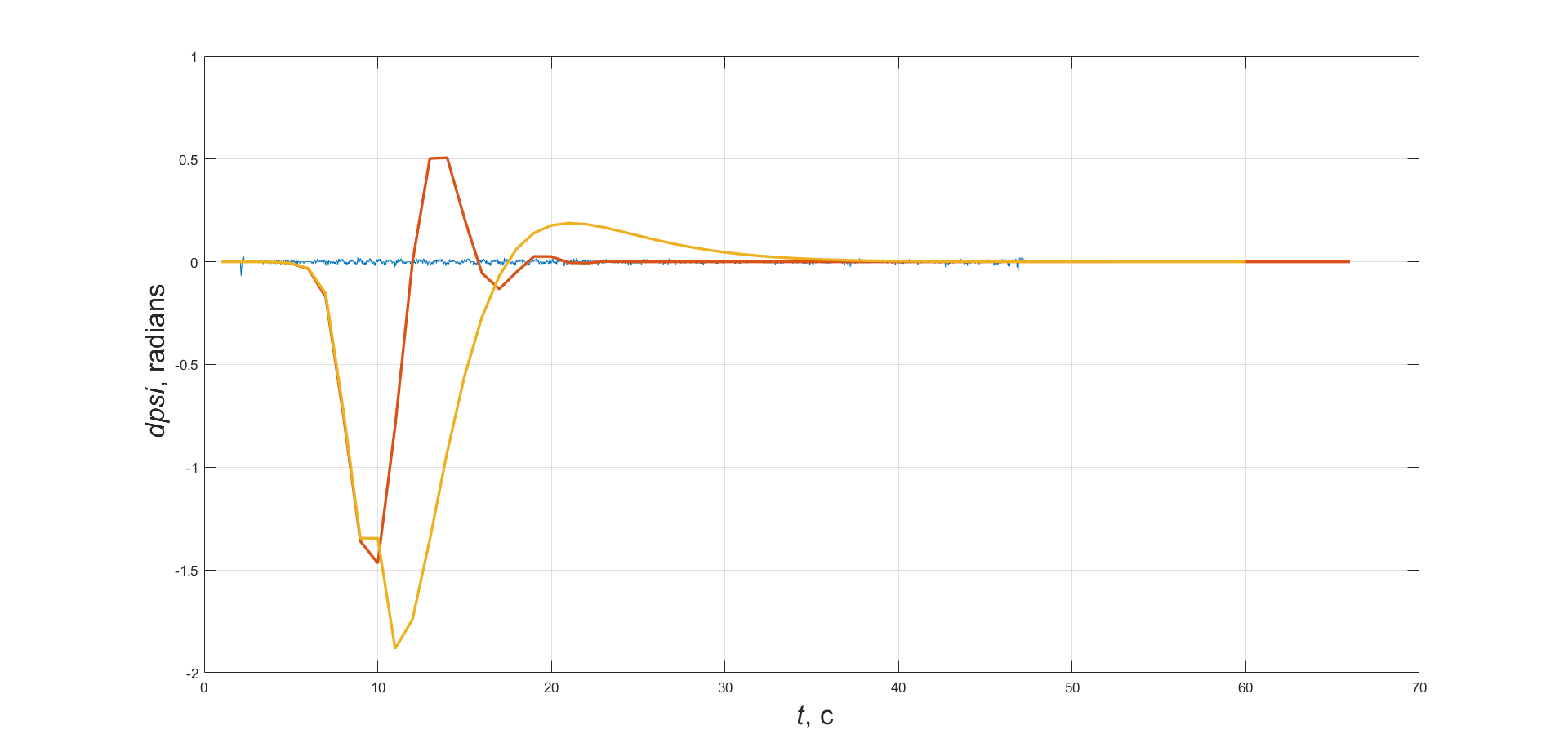
2.7. Графики эксперимент + модель



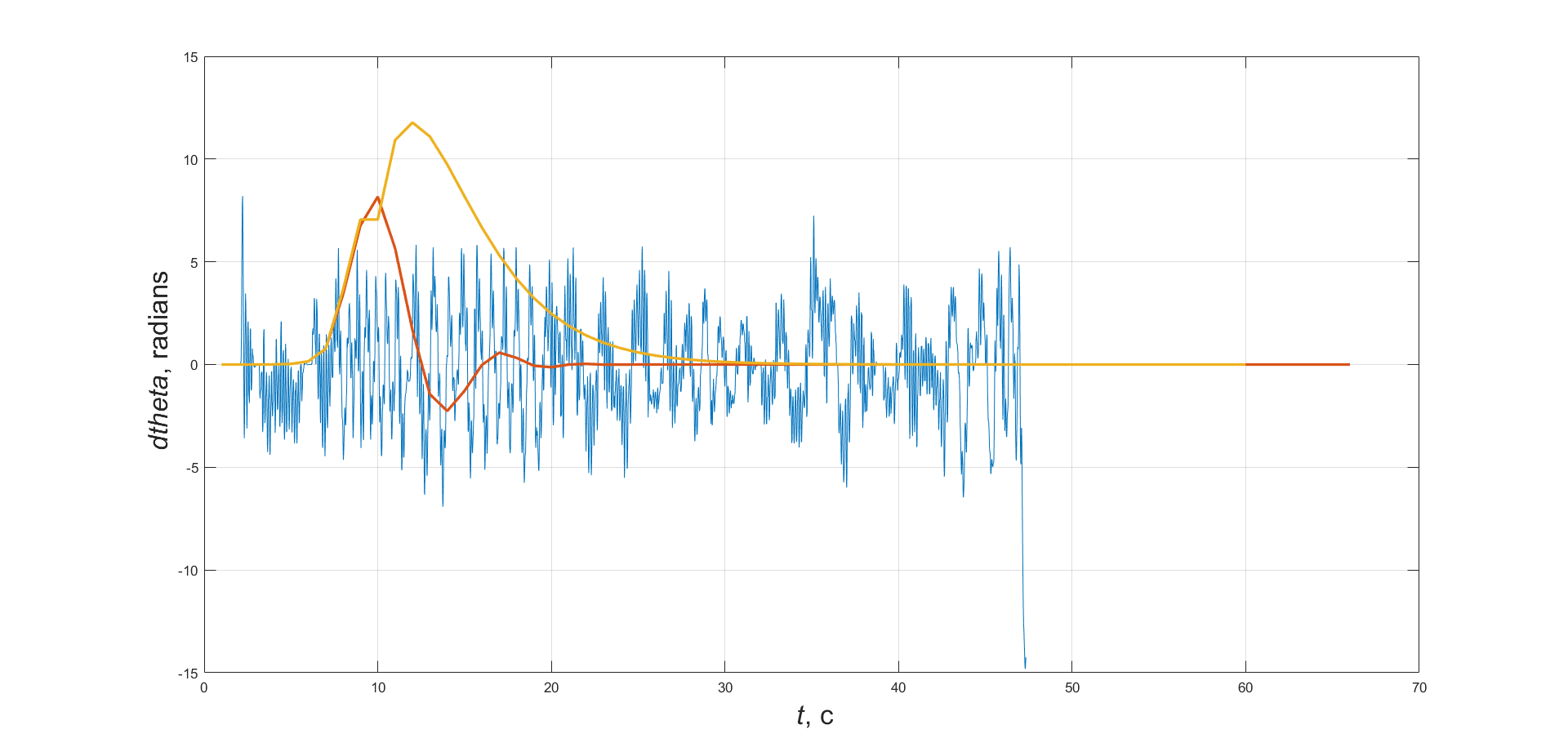
*Рисунок 16. График зависимости угла поворота моторов от времени t*

**

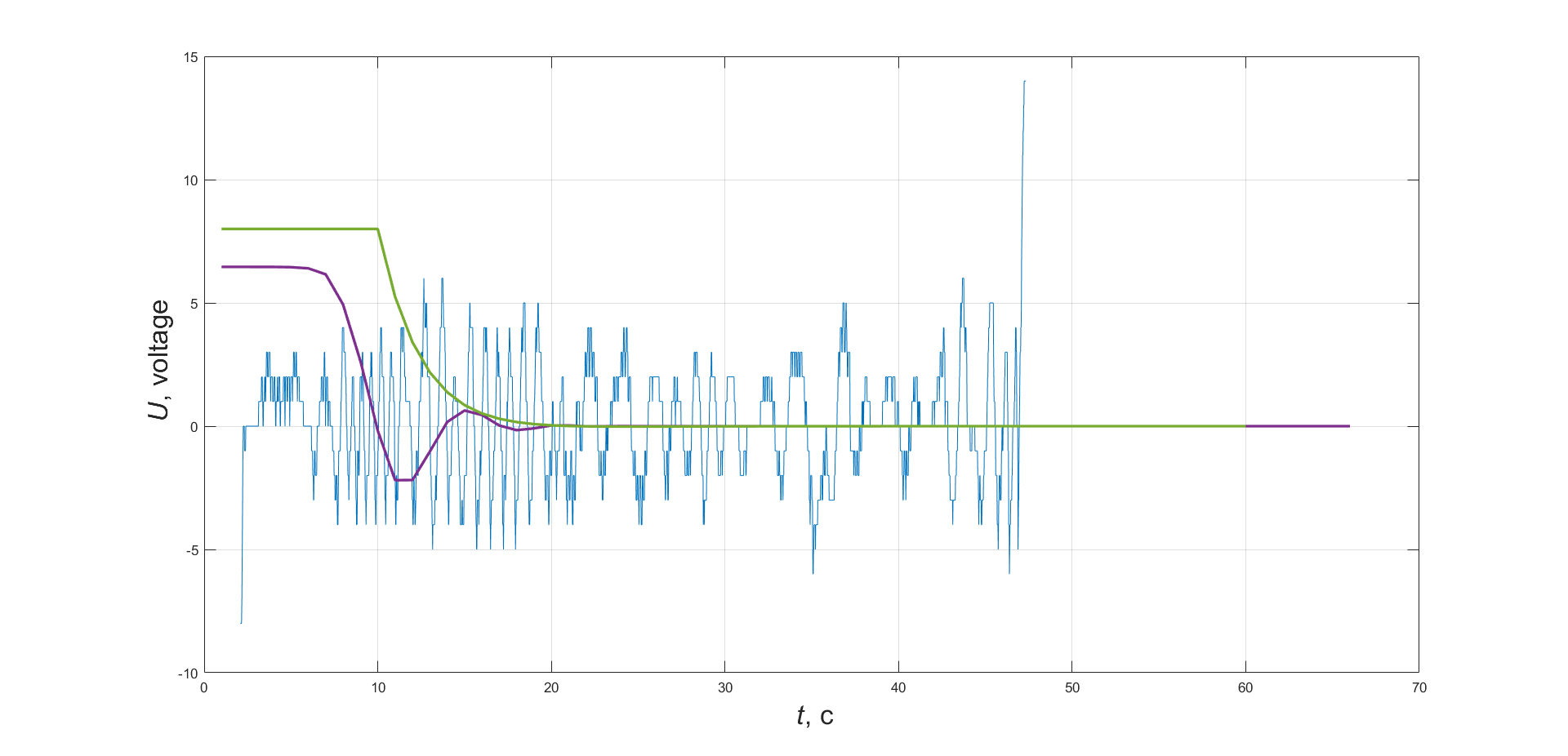
*Рисунок 17. График зависимости угла отклонения от времени t*

**

*Рисунок 18. График зависимости от времени t*

**

*Рисунок 19. График зависимости от времени t*

**

*Рисунок 20. График зависимости напряжения U от времени t*

1. Код расчета коэффициентов для LQR регулятора

В этой лабораторной работе мы использовали линейно-квадратичный регулятор. По заданию нужно было использовать ПИД регулятор, но у него есть некоторые недочеты. Основное преимущество ЛКР-регулятора перед ПИД-регулятором — это критерий оптимальности. Этот критерий является результатом минимизации определенного интеграла. Критерий оптимальности позволяет получить максимально быструю сходимость, не выходя при этом за физические ограничения управления. Грубо говоря, мы ставим только критерий скорости, и добавляем щепотку ограничений на контроль, до тех пор, пока значения не впишутся в рамки.  
Настройка линейно-квадратичного регулятора ведется с помощью изменения матриц Q и R соответственно. Этот функционал может дать перерегулирование, а может и нет, смотря как выбраны матрицы. Но одним из главных преимуществ ЛКР-регулятора является то, что всегда можно подобрать оптимальные для нашего функционала матрицы, то есть всегда можно хорошо и качественно настроить наш регулятор.

**Код программы:**

import numpy as np

from control import lqr

mt = 0.623;

mk = 0.02;

l = 0.16;

r = 0.028;

g = 9.82;

Jt = 0.001287;

Jk = 1/2 \* mk \* r\*\*2;

km = 0.488;

ke = 0.488;

R = 8.209;

J = 0.0024;

k1 = mt\*l\*r\*(mt\*l\*r - 2\*J) - (mt\*l\*\*2 + Jt) \*(mt\*r\*\*2 + 2\*mk\*r\*\*2 + 2\*Jk + 2\*J)

A = np.matrix ([[0, 0, 1],

[mt\*\*2 \* g \* l\*\*2 \* r / k1, 2 \* km \* ke \* (mt \* l\* r + mt \* l\*\*2 + Jt) / (R \* k1), 0],

[-mt \* g \* l\*(mt\* r\*\*2 + 2 \* mk \* r\*\*2 + 2\*Jk + 2\* J) / k1, -2 \* km \* ke \* (mt \* l \* r + mt\*r\*\*2 + 2\*mk\*r\*\*2 + 2\*Jk) / (R \* k1), 0]])

B = np.matrix([[0],

[-2 \* km \* (mt \*l\*r + mt \* l\*\*2 + Jt)/(R\*k1)],

[2\*km\*(mt\*l\*r + mt\*r\*\*2 + 2\* mk \*r\*\*2 + 2\*Jk)/(R\*k1)]])

Q = np.matrix([

[1, 0, 0],

[0, 1, 0],

[0, 0, 1]

])

R\_cost = 1

# x' = Ax + Bu

# u = -kx

# x' = (A-kB)x

k, \*\_ = lqr(A, B, Q, R\_cost)

print(k)

Код Python

#!/usr/bin/env python3  
from ev3dev2.motor import \*  
from ev3dev2.sensor.lego import \*  
from ev3dev2.power import PowerSupply  
import math  
import time  
motorA = LargeMotor('outA')  
motorB = LargeMotor('outD')  
Gyro = GyroSensor('in3')  
brick = PowerSupply()  
max\_voltage = brick.measured\_volts  
Gyro.mode = Gyro.MODE\_GYRO\_ANG  
t\_start = time.time()  
t\_cur = time.time() - t\_start  
t\_prev = t\_cur  
dt = 0  
k1 = -240.50872697  
k2 = -2.37783527  
k3 = -35.76041424  
k = 0.9  
psi\_cur = 0  
dpsi = 0  
psi\_prev = 0  
dtetha = 0  
Gyro.calibrate()  
GyroZero = Gyro.angle  
Gyro.rate  
pwr = 0  
fh = open('lab7.txt', 'w')  
b = 0.5  
filter\_prev = 0  
dpsi\_prev = 0  
def filter(value):  
 global b, filter\_prev, dpsi\_prev  
 out = (1-b)\*filter\_prev + b \* dpsi\_prev  
 filter\_prev = out  
 dpsi\_prev = value  
 return out  
print("START!")  
while (True):  
 t\_cur = time.time() - t\_start  
 #dt = t\_cur - t\_prev  
 #psi\_cur = ((psi\_prev + dpsi \* dt) \* (1 - b) + (Gyro.angle - GyroZero) \* b) / 180 \* 3.14 psi\_cur = (Gyro.angle - GyroZero) / 180 \* math.pi  
 dpsi = filter(psi\_cur-psi\_prev)  
 dtetha = (motorA.speed + motorB.speed) / 2 / 180 \* 3.14  
 pwr = k1 \* psi\_cur + k2 \* dtetha + k3 \* dpsi  
 pwr = pwr / max\_voltage \* 100  
 if(psi\_cur > math.pi/10):  
 pwr = -100  
 if(psi\_cur < -math.pi/10):  
 pwr = 100  
 if (pwr > 100):  
 pwr = 100  
 if (pwr < -100):  
 pwr = -100  
 motorB.run\_direct(duty\_cycle\_sp=pwr)  
 motorA.run\_direct(duty\_cycle\_sp=pwr)  
 fh.write(str(t\_cur) + " " + str(psi\_cur) + " " + str(dpsi) + " " + str(dtetha) + " " + str(pwr) + str(t\_cur) + "\n")  
 t\_prev = t\_cur  
 psi\_prev = psi\_cur  
fh.close()

1. Код Matlab

k\_m = 0.29;

k\_e = k\_m;

R = 4.7;

m\_t = 0.539;

m\_k = 0.023;

r = 0.028;

l = 0.09;

J = 0.0023;

J\_k = 0.000009016;

J\_t = 0.000360083;

g = 9.8;

Um = 8;

t = 0.7;

t\_zvezda = 6.3;

w\_0 = t\_zvezda/t;

x\_1 = m\_t\*l\*r\*(m\_t\*l\*r-2\*J)-(m\_t\*l^2+J\_t)\*(m\_t\*r^2+2\*m\_k\*r^2+2\*J\_k+2\*J);

a11 = 0;

a12 = 0;

a13 = 1;

a21 = (m\_t^2\*g\*r\*l^2)/x\_1;

a22 = (2\*k\_m\*k\_e\*(m\_t\*l\*r + m\_t\*l^2 + J\_t))/(R\*x\_1);

a23 = 0;

a31 = (-m\_t\*g\*l\*(m\_t\*r^2 + 2\*m\_k\*r^2 + 2\*J\_k + 2\*J))/x\_1;

a32 = (-2\*k\_m\*k\_e\*(m\_t\*l\*r + m\_t\*r^2 + 2\*m\_k\*r^2 + 2\*J\_k))/(R\*x\_1);

a33 = 0;

b1 = 0;

b2 = (-2\*k\_m\*(m\_t\*l\*r + m\_t\*l^2 + J\_t))/(R\*x\_1);

b3 = (2\*k\_m\*(m\_t\*l\*r + m\_t\*r^2 + 2\*m\_k\*r^2 + 2\*J\_k))/(R\*x\_1);

A = [a11 a12 a13; a21 a22 a23; a31 a32 a33];

B = [b1; b2; b3];

C = A\*B;

D = A^2\*B;

Y = [B(1,1) C(1,1) D(1,1); B(2,1) C(2,1) D(2,1); B(3,1) C(3,1) D(3,1)];

E = [0 b2 b3; b3 0 a32\*b2-a22\*b3; a32\*b2-a22\*b3 a21\*b3-a31\*b2 0];

W = [3\*w\_0 + a22; 3\*w\_0^2 + a31; w\_0^3 - a22\*a31 + a21\*a32];

K = E\W;

k\_1 = K(1,1);

k\_2 = K(2,1);

k\_3 = K(3,1);

U=out.psi.data;

plot(U,LineWidth=2)

grid on;

xlabel('{\it t}, c','Fontsize',20);

ylabel('{\it psi}, radians','Fontsize',20)

hold on;

1. Небольшое дополнительное задание

Мы вывели модель относительно угловой и линейной скорость нашего сигвея.

Изображение выглядит как текст, квитанция

Автоматически созданное описание

1. Вывод

В данной лабораторной работе мы смогли добиться устойчивости нашего сигвея и удерживать его в таком состоянии продолжительное время. Так же мы вывели полную нелинейную модель робота. Дополнительно был реализован вывод полной нелинейной модели относительно угловой и линейной скоростей.  
В итоге мы получили опыт и освоили навыки работы с неполноприводными системами.