

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 №2

«Идентификация трения в сочленении манипулятора. Расчет коэффициентов
регуляторов»

по дисциплине «Введение в профессиональную деятельность»

Выполнили: студенты гр. R3135

Дупак А. А.,

Щтенников Р. А.,

Зорькина А. А.

Преподаватель: Перегудин А. А.

Санкт-Петербург
2019

Цель работы

Ознакомиться с методами идентификации характеристик системы и расчета коэффициентов регуляторов.

Материалы работы

Задачи решаемые при выполнении работы

1. Подавая на сочленение напряжение $u(t) = u_m \sin \omega t$, при трех разных ω снять показания углов поворота и соответствующих значений времени. U_m — максимальное напряжение.

2. Аппроксимировать полученные данные следующей формулой по A:

$$q_i(t) = \frac{B(A^2(1 - \cos \omega t) + \omega^2(1 - e^{-At}) - A\omega \sin \omega t)}{A\omega(A^2 + \omega^2)} \quad \text{где} \quad A = \frac{K_f}{J_m} + \frac{K_e \cdot K_m}{J_m \cdot R}, B = \frac{K_m \cdot U_m}{J_m \cdot R},$$

K_f — коэффициент трения, K_e , K_m — конструктивные постоянные, R — сопротивление, J_m — приведенный момент инерции.

3. Вычислить коэффициент трения K_f для каждой ω . Рассчитать итоговое значение как среднее арифметическое.

4. Рассчитать коэффициенты ПИД-регулятора угла поворота сочленения подстановкой полинома Ньютона и Баттерворта.

5. Рассчитать коэффициенты ПИ-регулятора угловой скорости сочленения подстановкой полинома Ньютона и Баттерворта.

6. Сделать моделирование исследуемого процесса в среде *xcos* пакета *scilab*.

Результаты необходимых расчетов и построений

Таблица 1. Коэффициент трения K_f .

ω	π	2π	3π	итог
K_f	0.0004	0.0057	0.0011	0.0024

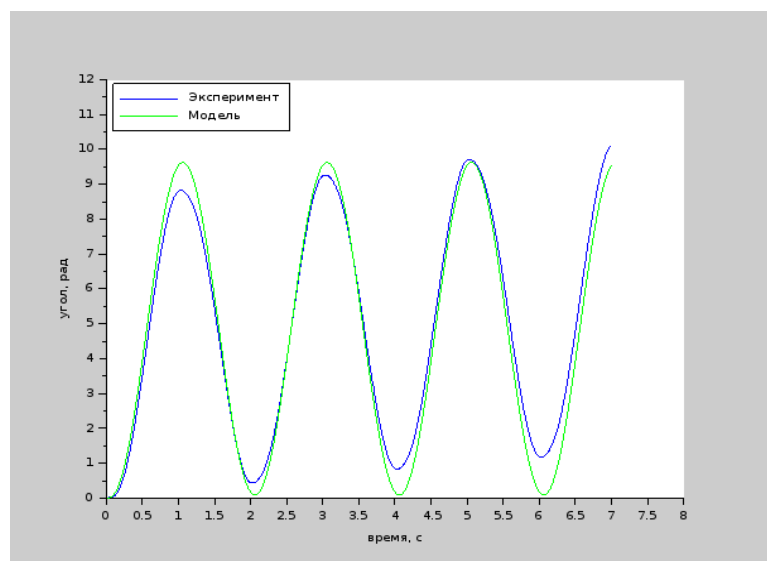


Рис. 1. График зависимости угла поворота вала двигателя от времени при $\omega = \pi$.

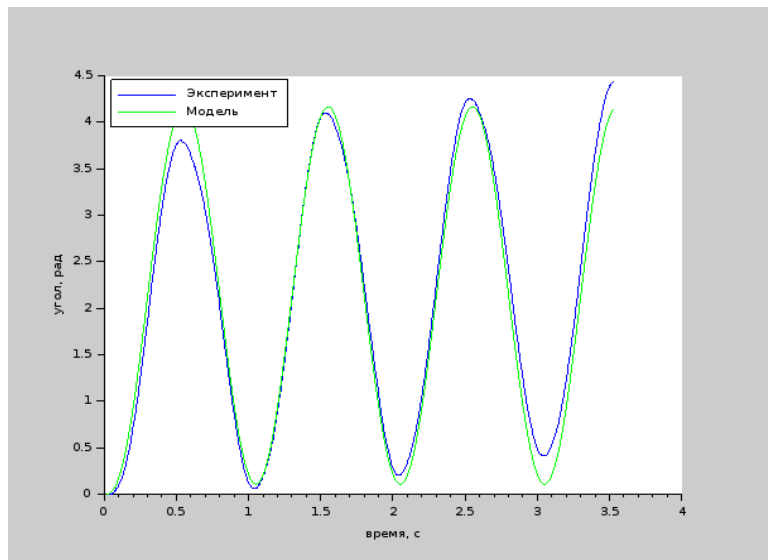


Рис. 2. График зависимости угла поворота вала двигателя от времени при $\omega = 2\pi$.

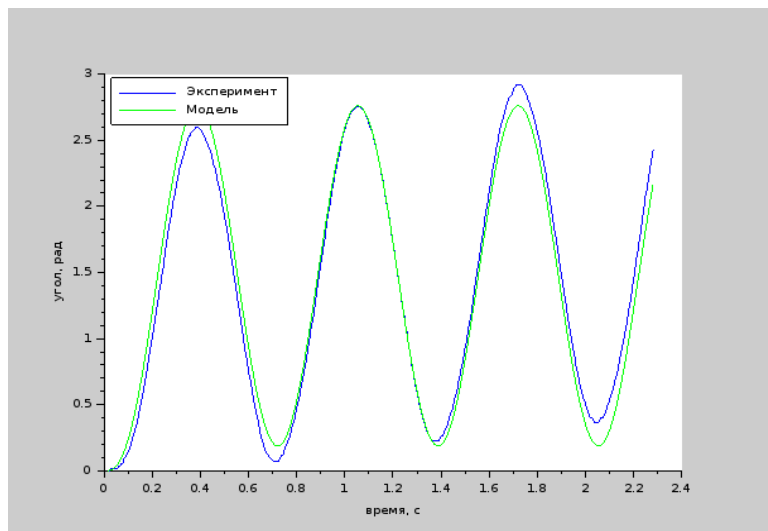


Рис. 3. График зависимости угла поворота вала двигателя от времени при $\omega = 3\pi$.

Таблица 2. Коэффициенты ПИД-регулятора угла поворота.

Подстановка	Ньютона	Баттерворта
t_p	0.6	0.3
k_p	0.8	2
k_i	2.9	20
k_d	0.035	0.057

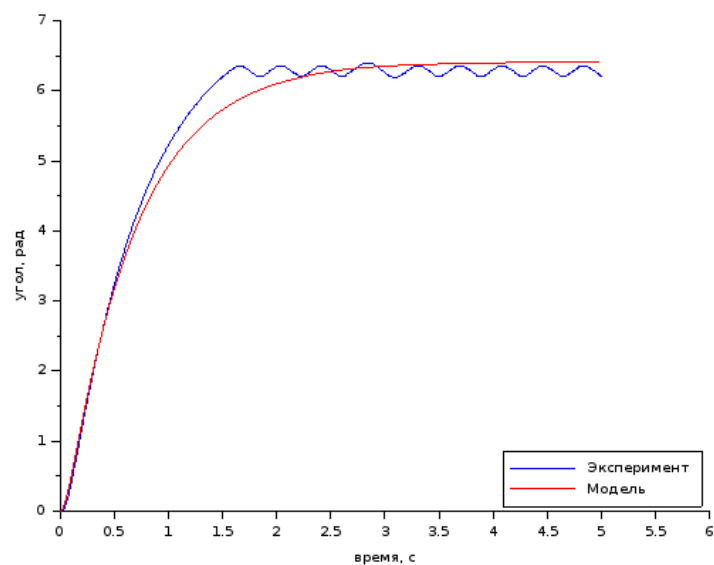


Рис. 4. График переходного процесса регулятора угла поворота сочленения с расчетом коэффициентов подстановкой полинома Ньютона.

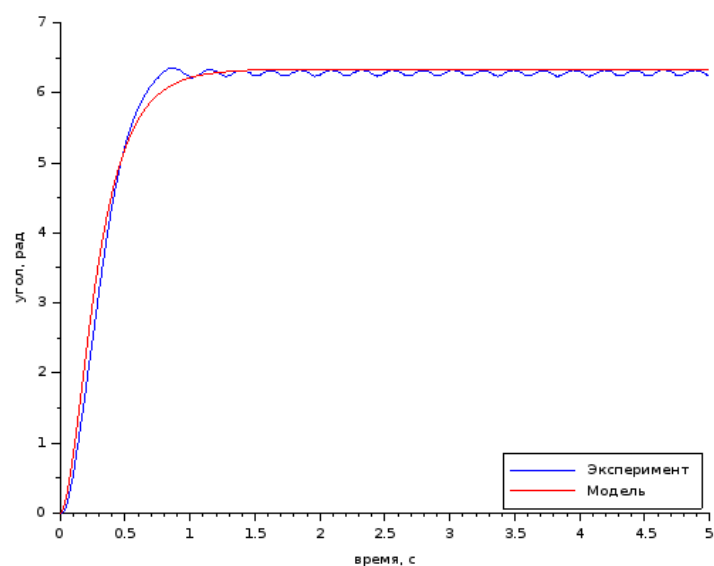


Рис. 5. График переходного процесса регулятора угла поворота сочленения с расчетом коэффициентов подстановкой полинома Баттерворта.

Таблица 3. Коэффициенты ПИ-регулятора угловой скорости.

Подстановка	Ньютона	Баттерворта
t_p	0.3	0.2
k_p	0.037	0.007
k_i	0.64	0.5

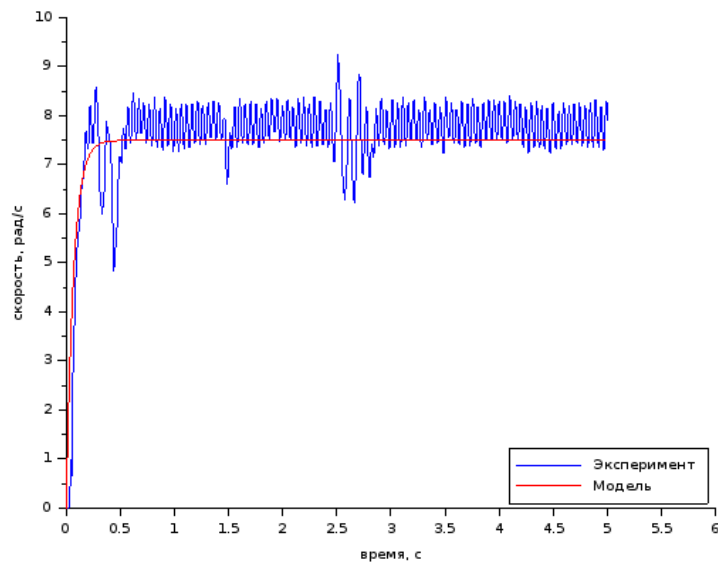


Рис. 6. График переходного процесса регулятора угловой скорости сочленения с расчетом коэффициентов подстановкой полинома Ньютона.

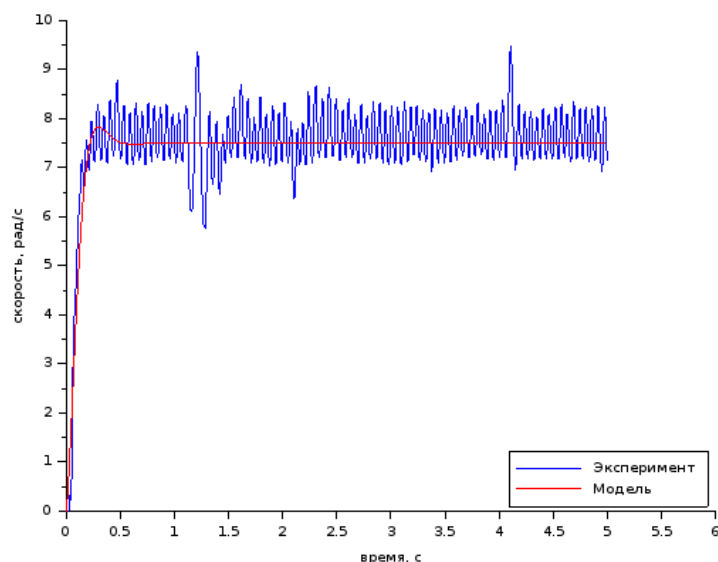


Рис. 7. График переходного процесса регулятора угловой скорости сочленения с расчетом коэффициентов подстановкой полинома Баттерворта.

Код программы для EV3

a) Генерация синусоидального напряжения

```
#!/usr/bin/env python3
from ev3dev.ev3 import *
import time
import math

mA = LargeMotor('outA')
mA.position = 0
start = time.time()
try:
    with open('sin_3pi.txt', 'w') as f_obj:
        while (time.time()-start) < 7:
```

```

        u = 100*math.sin(3*math.pi*(time.time()-start))
        mA.run_direct(duty_cycle_sp=u)
        f_obj.write(str(mA.position) + ' ' +
str(round((time.time()-start)*1000)) + '\n')
finally:
    mA.stop(stop_action='brake')

```

б) ПИД-регулятор угла поворота сочленения.

```

#!/usr/bin/env python3
from ev3dev.ev3 import *
import math

kp = 4.5
ki = 67.5
kd = 0.1
integral = 0
err_old = 0
km = 0.5
R = 6.1
mA = LargeMotor('outA')
try:
    mA.position = 0
    with open('bat_t02_p'+str(kp)+'-i'+str(ki)+'-d'+str(kd)+'.txt',
'w') as f_obj:
        start = time.time()
        while (time.time() - start) < 5:
            err = 2*math.pi - mA.position*math.pi/180
            integral += err*ki
            if abs(integral) > 1:
                integral = math.copysign(1, integral)
            derivative = (err - err_old)*kd
            err_old = err
            u = kp * err + integral + derivative
            u = (u*km/R)*100/8.4
            if abs(u)>100:
                u=math.copysign(100,u)
            mA.run_direct(duty_cycle_sp=u)
            f_obj.write(str(mA.position) + ' ' + str(time.time() -
start) + '\n \n')
finally:
    mA.stop(stop_action='brake')

```

в) ПИ-регулятор угловой скорости сочленения.

```

#!/usr/bin/env python3
from ev3dev.ev3 import *
import math

kp = 0.058
ki = 2.1
integral = 0
mA = LargeMotor('outA')
try:

```

```

mA.position = 0
with open('bat_t01_p'+str(kp)+'-i'+str(ki)+'.txt', 'w') as f_obj:
    start = time.time()
    while (time.time() - start) < 5:
        err = 10 - mA.speed*math.pi/180
        integral += err*ki
        if abs(integral) > 10:
            int = math.copysign(1, integral)
        u = kp * err + integral
        u = u*100/8.4
        if abs(u)>100:
            u=math.copysign(100,u)
        mA.run_direct(duty_cycle_sp=u)
        f_obj.write(str(mA.speed) + ' ' + str(time.time() -
start) + '\n \n')
    finally:
        mA.stop(stop_action='brake')

```

Схема моделирования

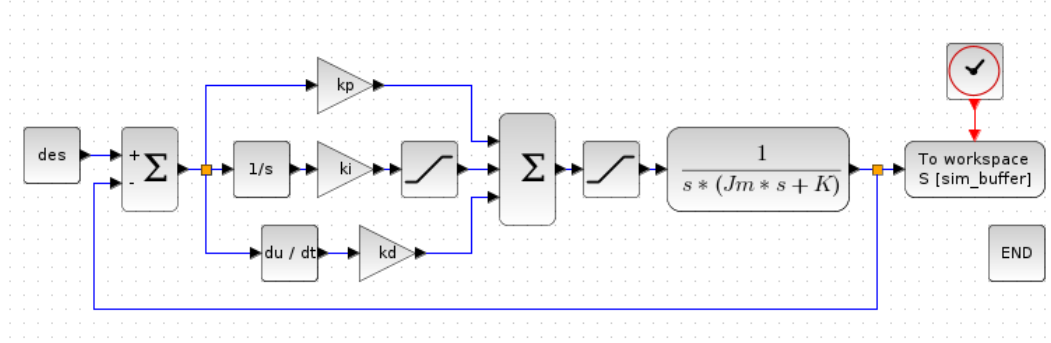


Рис.8. Схема моделирования ПИД-регулятора угла поворота сочленения.

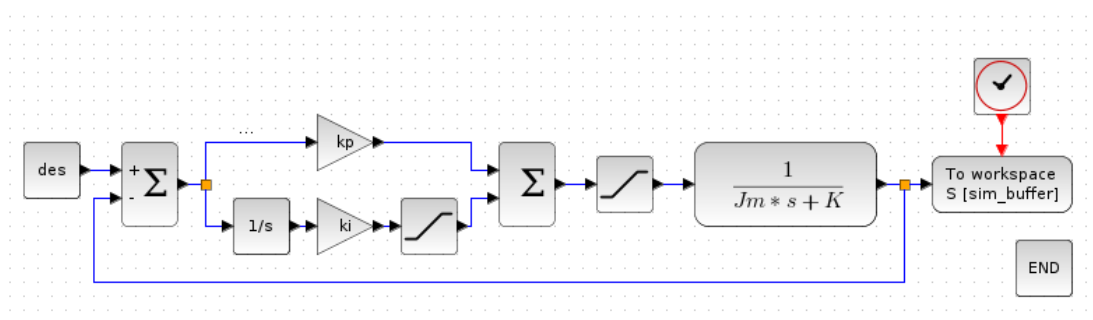


Рис.9. Схема моделирования ПИ-регулятора угловой скорости сочленения.

Код программы Scilab

а) Аппроксимация синусоиды и расчет коэффициента трения.

```

Ke = 0.5
Km = 0.5
Jm = 0.0025
R = 6
Umax = 8.4
Kf = 0
A = Kf/Jm + (Ke*Km)/(Jm*R)
B = (Km*Umax)/(Jm*R)
l = 1
w = l*pi

```

```

import = "/home/aleksandr/ITM0_lab2/ev3/lab2/data/sin_"+string(l)+"pi.txt"
export1 = "/home/aleksandr/ITM0_lab2/ev3/lab2/photos/graph"+string(l)
+".png"
export2 =
"/home/aleksandr/ITM0_lab2/ev3/lab2/photos/graph"+string(l+3)+".png"

results = read(import, -1, 2)
len = length(results)/2
angle = results(1:len/l, 1)*%pi/180
time = results(1:len/l, 2)/1000
angle2 = B*(A^2*(1-cos(w*time))+w^2*(1-exp(-A*time))-A*w*sin(w*time))/(
(A*w*(A^2+w^2))
f1 = figure(1)
a1 = newaxes()
plot2d(time, angle, 2)
plot2d(time, angle2, 5)
xtitle('', 'время, с', 'угол, рад')
legend('Эксперимент', 'Модель', 2)
xs2png(f1, export1)

function err=func(estimate, measure)
    time = measure(1)
    angle = measure(2)
    A = estimate(1)
    B = estimate(2)
    err = angle - B*(A^2*(1-cos(w*time))+w^2*(1-exp(-A*time))-
A*w*sin(w*time))/(A*w*(A^2+w^2))
endfunction

aim = [time, angle]'
init_guess = [40; 4]
[koeffs, errs] = datafit(func, aim, init_guess)
A = koeffs(1)
B = koeffs(2)
Kf = Jm*A - Ke*Km/R
angle3 = B*(A^2*(1-cos(w*time))+w^2*(1-exp(-A*time))-A*w*sin(w*time))/(
(A*w*(A^2+w^2))
f2 = figure(2)
a2 = newaxes()
plot2d(time, angle, 2)
plot2d(time, angle3, 3)
xtitle('', 'время, с', 'угол, рад')
legend('Эксперимент', 'Модель', 2)
xs2png(f2, export2)

```

б) Расчет коэффициентов регуляторов.

```

Jm = 0.0025
Ke = 0.5
Km = 0.5
R = 6.1
Umax = 8.4
Kf = 0.0024
A = Kf/Jm + (Ke*Km)/(Jm*R)
B = (Km*Umax)/(Jm*R)
K = Jm*A

tp = 0.5
//Nuton
t_n2 = 4.8

```



```

t_n3 = 6.3
w0_n2 = t_n2/tp
w0_n3 = t_n3/tp
//pid
Kp_n = 3*(w0_n3^2)*Jm
Ki_n = (w0_n3^3)*Jm
Kd_n = 3*w0_n3*Jm - K
//pi
kp_n1 = 2*w0_n2*Jm - K
ki_n1 = (w0_n2^2)*Jm
//Batterwort
t_b2 = 2.9
t_b3 = 6.0
w0_b2 = t_b2/tp
w0_b3 = t_b3/tp
//pid
kp_b = 2*(w0_b3^2)*Jm
ki_b = (w0_b3^3)*Jm
kd_b = 2*w0_b3*Jm - K
//pi
kp_b1 = 1.4*w0_b2*Jm - K
ki_b1 = (w0_b2^2)*Jm

```

в) Построение графиков.

```

sim_time = 5
sim_period = 0.01
sim_buffer = sim_time/sim_period
Jm = 0.0025
K = 0.043
km = 0.5
R = 6.1
voltage = 8.4
des = 2*%pi
kp = 0.037
ki = 0.64
kd = 0
anti_wind_up = 1
file_name = "Nut_sp_pi/t0.3_p0.037-i0.64.txt"
res = read("/home/aleksandr/ITMO_lab2/ev3/lab2/data/" + file_name, -1, 2)
y = res(:, 1)*%pi/180
x = res(:, 2)
importXcosDiagram("/home/aleksandr/ITMO_lab2/ev3/lab2/scilab/model1.zcos")
xcos_simulate(scs_m, 4)
plot2d(x, y, 2)
plot2d(S.time, S.values, 5)
xtitle('', 'время, с', 'скорость, рад/с')
legend('Эксперимент', 'Модель', 4)
xs2png(0, "/home/aleksandr/ITMO_lab2/ev3/lab2/photos/speed_nut_03")

```

Выводы

В результате проделанной работы была решена задача расчета коэффициентов регулятора, а также реализовано управление двигателем.

Кроме того, была построена схема моделирования исследуемого процесса в среде Xcos. Сравнив график экспериментальных данных с графиком, построенным на основании результатов моделирования схемы, мы удостоверились в том, что моделирование исследуемого процесса дает результаты схожие со значениями полученными в ходе прямых измерений.