

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

"Национальная научно-образовательная корпорация ИТМО"

Факультет Систем Управления и Робототехники

Лабораторная работа №3 "Использование регуляторов для
управления двигателем"

по дисциплине

«Введение в профессиональную деятельность»

Группа:

R3136

Выполнила команда:

Поляков Антон Александрович

Ибахаев Зубайр Руслан-Бекович

Иванов Виктор Олегович

Скавронский Александр Вадимович

Преподаватель:

Алексей Алексеевич Перегудин

Санкт-Петербург, 2023

Цель работы

Рассмотреть способы управления двигателем постоянного тока с использованием регуляторов, работающих на основе обратной связи.

Основная теория и формулы

Релейный регулятор

Принцип действия заключается в следующем: пока угол поворота двигателя меньше нужного, подается максимальное напряжение в одну сторону, когда угол поворота двигателя больше заданного, подается максимальное напряжение в другую сторону.

$$U = \begin{cases} U_{max}, & \theta < \theta^* \\ 0, & \theta = \theta^* \\ -U_{max}, & \theta > \theta^* \end{cases}$$

Уравнение 1. Релейный регулятор

Работу данного регулятора можно представить в виде формулы:

где U — напряжение, подаваемое на двигатель, U_{max} — максимально возможное напряжение, подаваемое на двигатель, θ — текущий угол поворота двигателя, θ^* — желаемый угол поворота двигателя.

Пропорциональный регулятор

Задаёт управление, пропорциональное разнице между текущим значением и желаемым. Формула для него выглядит следующим образом:

$$U = k_p(\theta^* - \theta)$$

Уравнение 2. Пропорциональный регулятор

где U — напряжение, подаваемое на двигатель, k_p — коэффициент пропорциональности, θ и θ^* — текущий и желаемый углы поворота двигателя соответственно.

Величину $e = (\theta^* - \theta)$ называют ошибкой управления.

Коэффициент k_p может высчитываться или подбираться эмпирически, в данной работе он будет подбираться в ходе экспериментов. Данный коэффициент показывает, насколько сильно ошибка управления влияет на управляющее воздействие.

ПИД-регулятор

Показатели качества — это оценка некоторых параметров переходного процесса. Существуют следующие показатели качества переходного процесса:

- **Перерегулирование**

Показывает, насколько сильны колебания, то есть насколько сильно переходный процесс отклоняется от заданного значения. Вычисляется перерегулирование как отношение разницы между максимальным значением и установившимся к установившемуся значению.

$$\sigma = \frac{f_{\text{макс}} - f_{\text{уст}}}{f_{\text{уст}}}$$

Уравнение 3. Перерегулирование

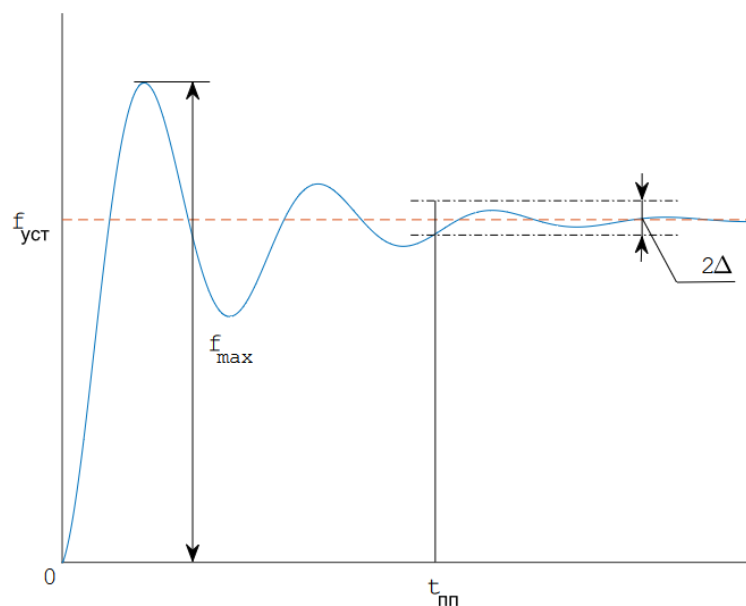


Рисунок 1. Перерегулирование и время переходного процесса

- **Время переходного процесса**

Время, отсчитываемое с момента приложения воздействия к системе, до момента, с которого отклонение регулируемой величины не превышает заранее заданного значения. Заранее заданное отклонение связано с тем, что система, как при моделировании, так и в реальной жизни, может так и не достигнуть требуемого значения. Обычно, отклонение принимают равным пяти процентам от установившегося значения. На рисунке 1 отклонение обозначено буквой Δ , а время переходного процесса — $t_{\text{пп}}$.

- **Установившаяся ошибка**

Оценивает ошибку управления в то время, когда система пришла в состояние равновесия.

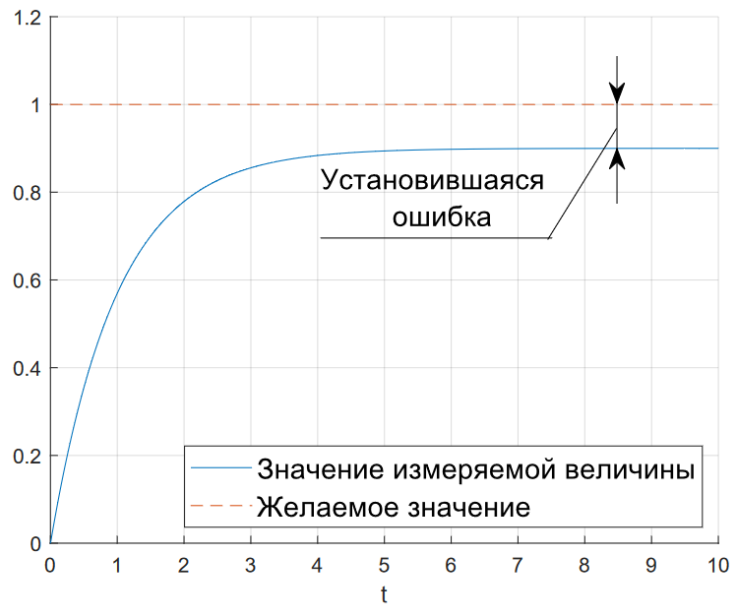


Рисунок 2. Установившаяся ошибка

ПИД-регулятор дополняет П-регулятор еще двумя составляющими: интегральной и дифференциальной.

- **Интегральная составляющая**

Пропорциональна интегралу по времени от отклонения регулируемой величины где $e(t) = \theta^* - \theta(t)$.

$$I = K_i \int_0^t e(\tau) d\tau$$

Уравнение 4. Интегральная составляющая

Она позволяет устранить основную ошибку пропорционального регулятора — статическую ошибку.

- **Дифференциальная составляющая**

Пропорциональна скорости изменения ошибки регулируемой величины и предназначена для противодействия отклонениям от целевого значения, которые прогнозируются в будущем.

$$D = K_d \frac{de}{dt}$$

Уравнение 5. Дифференциальная составляющая

Выходной сигнал регулятора определяется как сумма трех его составляющих.

$$U = P + I + D = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de}{dt}$$

Уравнение 6. Выходной сигнал регулятора

где K_p , K_i , K_d — коэффициенты усиления пропорциональной, интегральной и дифференциальной составляющих регулятора соответственно.

Выполнение

Релейный регулятор

```
1  #!/usr/bin/env python3
2  from ev3dev.ev3 import LargeMotor
3  import time
4
5  motorA = LargeMotor('outA')
6  angle_final = 180
7  voltage = 100
8  motorA.position = 0
9
10 try:
11     file = open("rele_reg15", "w")
12     timeStart = time.time()
13     while True:
14         if motorA.position > angle_final:
15             voltage = -100
16         elif motorA.position < angle_final:
17             voltage = 100
18         else:
19             voltage = 0
20
21         motorA.run_direct(duty_cycle_sp=voltage)
22         timeNow = time.time() - timeStart
23         file.write(str(timeNow) + " " + str(motorA.position) + " " + str(motorA.speed) + "\n")
24         if timeNow > 15:
25             break
26     file.close()
27
28 except Exception as e:
29     raise e
30 finally:
31     motorA.stop(stop_action='brake')
32     file.close()
```

Рисунок 3. Код релейного регулятора

Получить данные времени и угла поворота во время работы регулятора и построить по ним график.

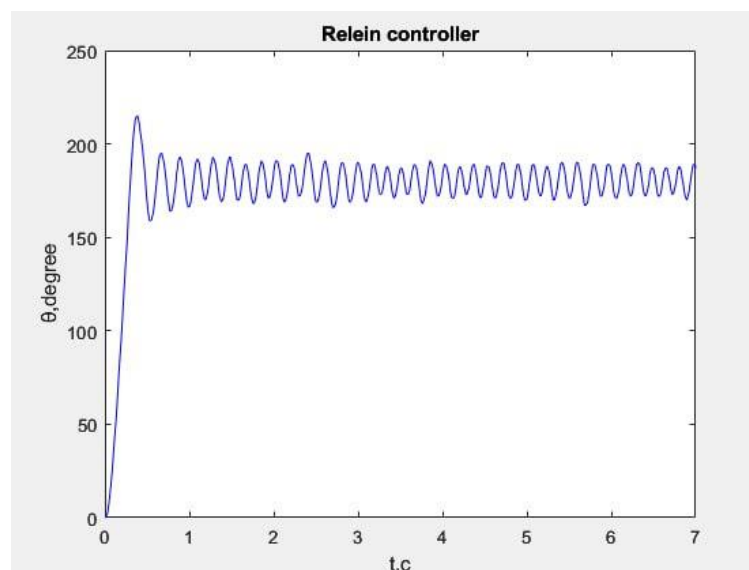


Рисунок 4. График для релейного регулятора

Пропорциональный регулятор

```
1  #!/usr/bin/env python3
2  from ev3dev.ev3 import LargeMotor
3  from ev3dev.ev3 import PowerSupply as ps
4  import time
5
6  motorA = LargeMotor('outA')
7  angle_final = 180
8  startPos = motorA.position
9
10 motorA.position = 0
11 timeStart = time.time()
12 max_voltage = ps().measured_volts
13 kp = 0.6
14 name = "prop_coef_" + str(kp)
15 file = open(name, "w")
16 while True:
17     angle_now = motorA.position
18     voltage = kp * (angle_now - angle_final)
19     if voltage > 100:
20         voltage = 100
21     elif voltage < -100:
22         voltage = -100
23     motorA.run_direct(duty_cycle_sp=voltage)
24     timeNow = time.time() - timeStart
25     file.write(str(timeNow) + " " + str(motorA.position) + " " + str(motorA.speed) + "\n")
26     if timeNow > 10:
27         break
28 timeStart = time.time()
29 file.close()
```

Рисунок 5. Код П-регулятора

Подобрать коэффициенты k_p , при которых:

1. Присутствует существенная установившаяся ошибка; $k_p = 0,6$

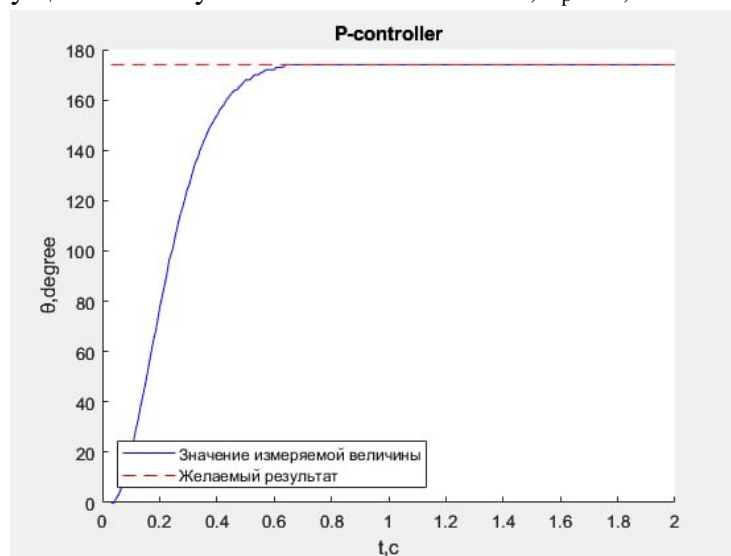


Рисунок 6. График для П-регулятора при $k_p=0,6$

2. Присутствует существенное перерегулирование, колебания в итоге затухают; $k_p = 1,5$

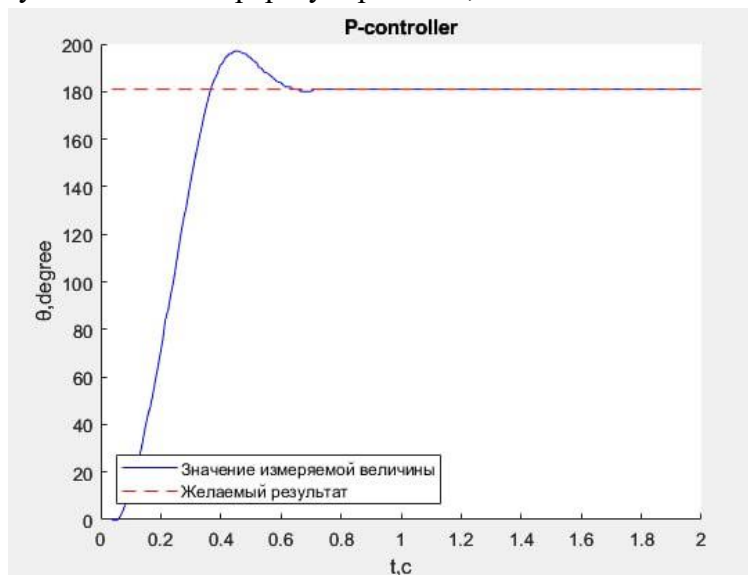


Рисунок 7. График для П-регулятора при $k_p=1,5$

3. Перерегулирование, время переходного процесса и установившаяся ошибка достаточно малы; $k_p = 0,8$

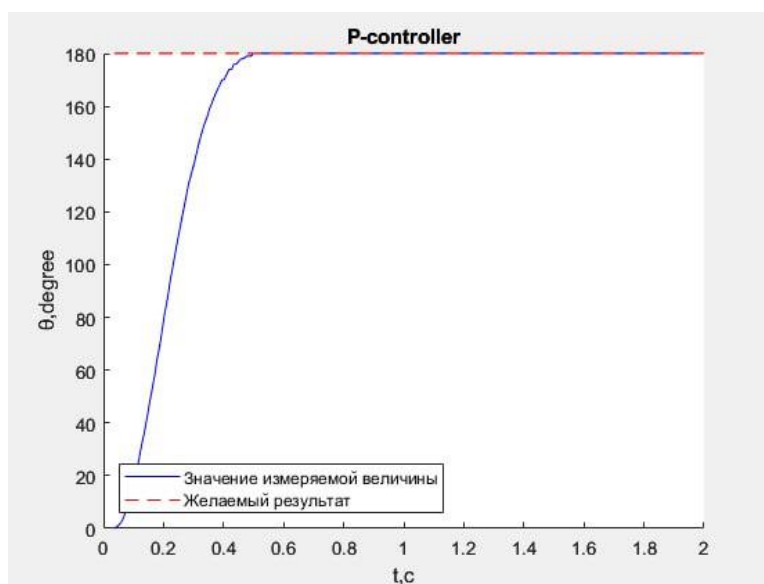


Рисунок 8. График для П-регулятора при $k_p=0,8$

ПИД-регулятор

```
1  #!/usr/bin/env python3
2  from ev3dev.ev3 import LargeMotor
3  from ev3dev.ev3 import PowerSupply as ps
4  import time
5
6  motorA = LargeMotor('outA')
7  angle_final = 180
8  startPos = motorA.position
9
10 motorA.position = 0
11 max_voltage = ps().measured_volts
12 kp = 0.5
13 ki = 0.06
14 kd = 0
15 name = "intg_coef_" + str(kp) + str(ki) + str(kd)
16 file = open(name, "w")
17 e_prev = 150
18 ei = 0
19 time_start = time.time()
20 timePrev = time_start
21 while True:
22     timeNow = time.time()-time_start
23     angle_now = motorA.position
24     e = angle_final - angle_now
25     de = (e - e_prev) / (timeNow - timePrev)
26     ei += e * (timeNow - timePrev)
27
28     voltage = kp * e + ki * ei + kd * de
29
30     if voltage > 100:
31         voltage = 100
32     elif voltage < -100:
33         voltage = -100
34     motorA.run_direct(duty_cycle_sp=voltage)
35     e_prev = e
36     timePrev = timeNow
37     file.write(str(timeNow) + " " + str(motorA.position) + " " + str(motorA.speed) + "\n")
38     if timeNow > 5:
39         break
40 file.close()
```

Рисунок 9. Код ПИД-регулятора