ИТМО

Факультет Систем Управления и Робототехники

ПОСТРОЕНИЕ МАТЕМАТИЧЕСКОЙ МОДЕЛИ ДВИГАТЕЛЯ EV3

Аннтоация – В этой работе мы заглянули "под капот" блока лего EV3: исследовали работу электродвигателя, проверив физические формулы на практике. С помощью графиков наглядно увидели, как зависят от времени угол и угловая скорость. Не обошлось и без трудностей: нам пришлось с нуля научиться работать с роботом, писать для него команды, строить графики по полученным данным, разбираться с кучей ошибок, которые мы получали в процессе. Возможность использовать Python и Matlab значительно упростила нам работу, но и здесь скрывались подводные камни, о которых вы можете прочитать в нашем отчёте. Но всё это было безумно интересно!

Ключевые слова: Математическая модель; Электродвигатель; Python; Lego EV3

Выполнили

Чуфарова Полина Игоревна[1], Антонова Алёна Евгеньевна[2], Бовржди Тамим[3], and Воробьёв Владимир Дмитриевич[4]

[1] *R3138, @fakkmoimosk*
[2] *R3138, @tucnutaya*
[3] *R3138, @tameemmaher*
[4] *R3138, @vvoblla*

Проверил

Овчаров Алексей Олегович

20 февраля 2024 г.

1 Постановка задачи

Цель работы: Познакомиться с оборудованием и программным обеспечением, которые понадобятся при изучении материала данного курса. Экспериментально проверить истинность найденных функций, описывающих работу ненагруженного двигателя постоянного тока, и определить значения входящих в них параметров k и T_m.

Функции, описывающие работу ненагруженного двигателя постоянного тока:

$$\omega(t) = kU_{\%}(1 - e^{-t/T_m}) \tag{1}$$

$$\theta(t) = kU_{\%}(t - T_m(1 - e^{-t/T_m}) \tag{2}$$

Задачи:

1. Снятие показаний с двигателя EV3;
2. Обработка экспериментальных данных;
3. Моделирование схемы в Simulink;
4. Построение графиков для полученных зависимостей;

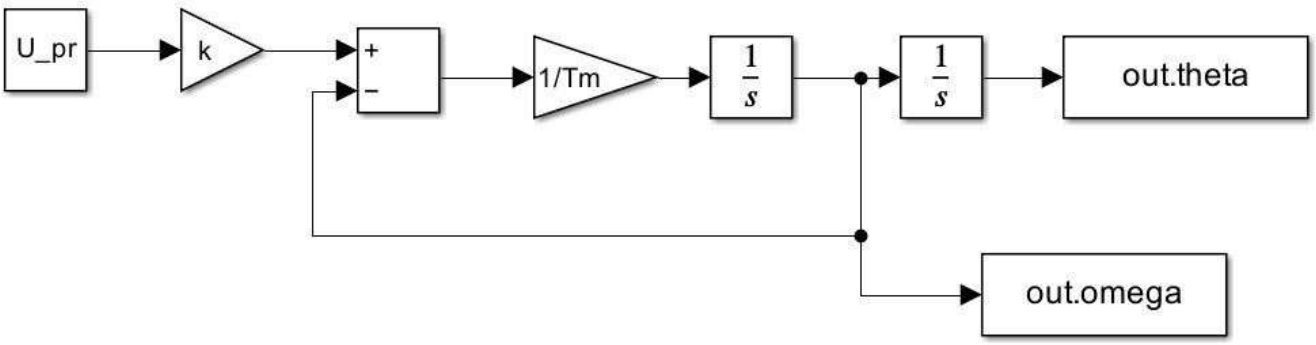

Рис. 1: Схема моделирования исследуемого процесса

2 Снятие показаний с двигателя EV3

Для того чтобы снять показания с двигателя, нам нужно было для начала подключиться к блоку EV3 с помощью компьютера. Для этого нам нужно, чтобы компьютер и робот были подключены к одной сети Wi-Fi, затем с помощью командной строки нам нужно выполнить подключение, используя следующие команды:

```
ping @[ip adress]
ssh robot@[ip adress]
```

Листинг 1: Подключение к блоку

Затем нужно ввести пароль (в нашем случае - maker), и подключение выполнено! Теперь мы можем отправлять нашему роботу файлы или же забирать файлы с робота. Это делается при помощи следующих команд:

```
scp file.py robot@[ip adress]:file.py
scp robot@[ip adress]:file.py file1.py
```

Листинг 2: Отправка и получение файла

Чтобы запустить файл на роботе, нужно выполнить следующую команду:

```
    python3 file.py
```

Листинг 3: Запуск программы

Для получения данных нам понадобились следующие программы:

```
#!/usr/bin/env

import math

f = open('file.csv', 'w')
for val in range(100):
    t = 0.1*val
    y = math.sin(2*t)
    f.write('{}, {}\n'.format(t, y))
f.close()
```

Листинг 4: Код file.py

```
#!/usr/bin/env python3
import ev3dev2.motor as motor
import time

voltages = [100, 80, 60, 40, 20, -20, -40, -60, -80, -100]
motor_a = motor.LargeMotor(motor.OUTPUT_B)
for vol in voltages:
    startTime = time.time()
    f = open('data{}.csv'.format(vol),'w')
    while (True):
        currentTime = time.time() - startTime
        motor_pose = motor_a.position
        motor_vel = motor_a.speed
        motor_a.run_direct(duty_cycle_sp = vol)
        f.write("{} {} {}\n".format(currentTime, motor_pose, motor_vel))

        if currentTime > 1:
            motor_a.run_direct(duty_cycle_sp = 0)
            time.sleep(1)
            break
    f.close()
```

Листинг 5: Код control.py

Программа control.py подает на двигатель EV3 максимально возможное постоянное напряжение. При этом она также периодически снимает показания текущего угла поворота ротора и его угловой скорости и записывает их с соответствующими значениями времени, прошедшего с начала работы программы, в два столбца в текстовый файл на EV3. В итоге мы получили 10 файлов, каждый файл соответствует определенному напряжению (100, 80, 60, 40, 20, -20, -40, -60, -80, -100). Это и есть нужные нам данные, с которым мы далее работаем.

Мы не приводим здесь таблицы с полученными данными, чтобы избежать загромождения отчёта. Полученные данные можно увидеть в нашем репозитории на GitHub.

3 Обработка экспериментальных данных

Для обработки данных мы использовали MatLab. Код для построения графиков также можно найти в репозитории на GitHub, здесь же мы приводим получившиеся графики зависимости угла и угловой скорости от времени.

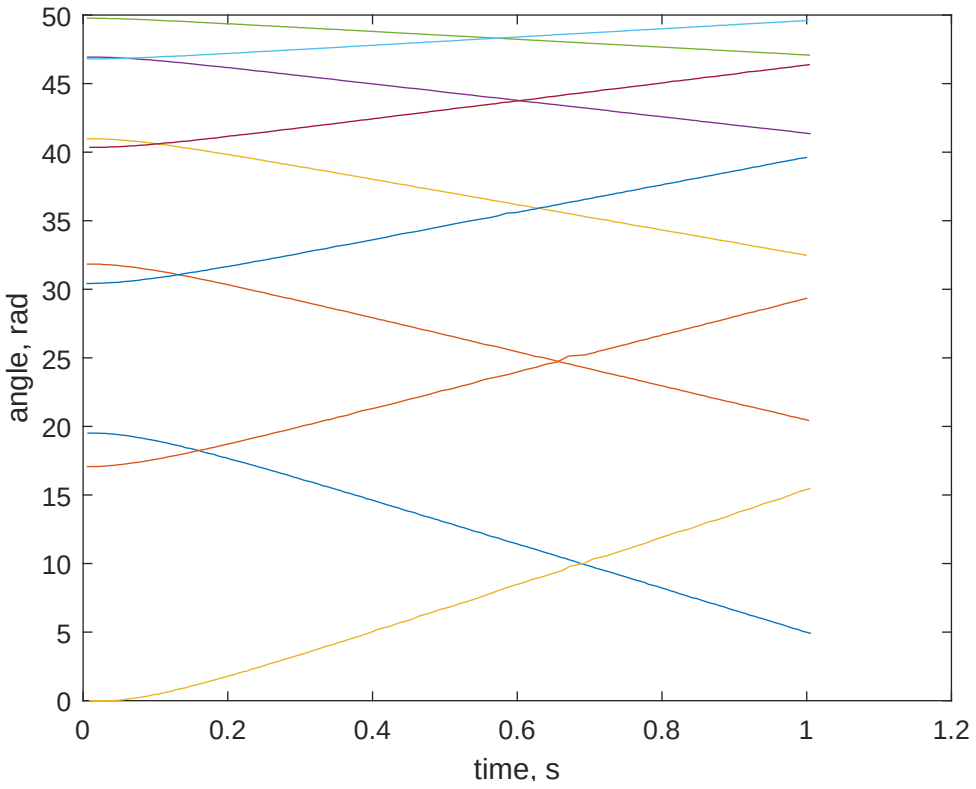

Рис. 2: График зависимости угла от времени при разных напряжениях

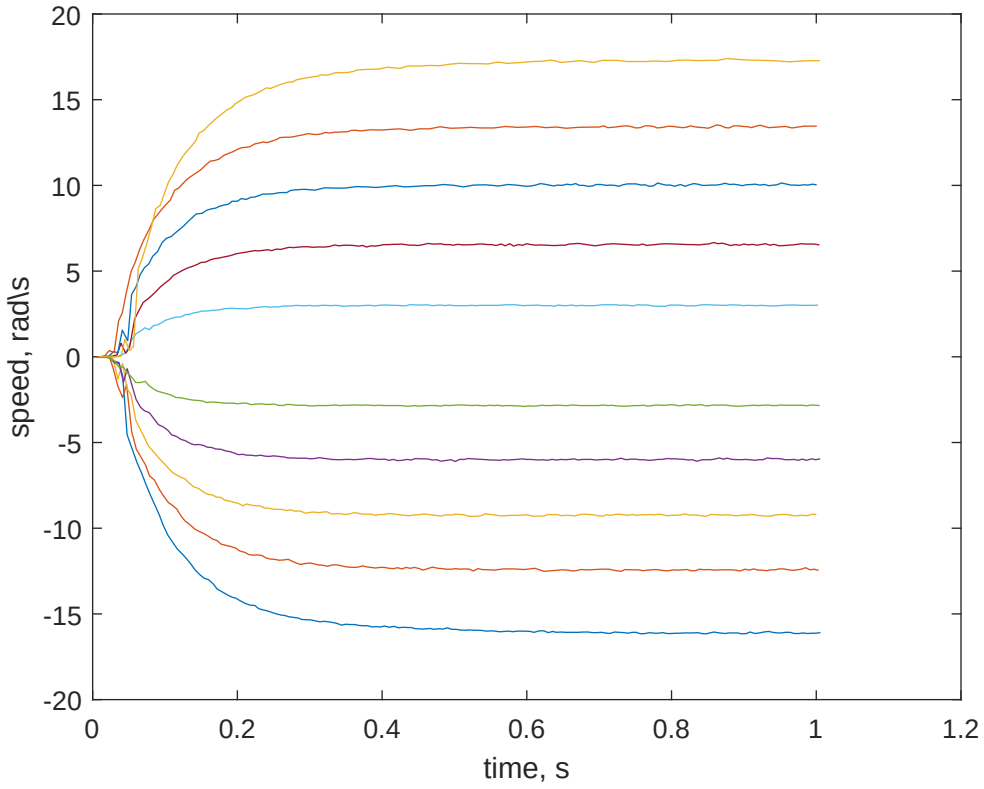

Рис. 3: График зависимости угловой скорости от времени при разных напряжениях

Далее нам нужно было выполнить аппроксимацию полученных данных. Для аппроксимации зависимости угла от времени была выбрана функция (2). На нашем GitHub можно найти графики аппроксимации для каждого напряжения по отдельности, здесь же для наглядности мы приводим графики для напряжения $U = 100\%$.

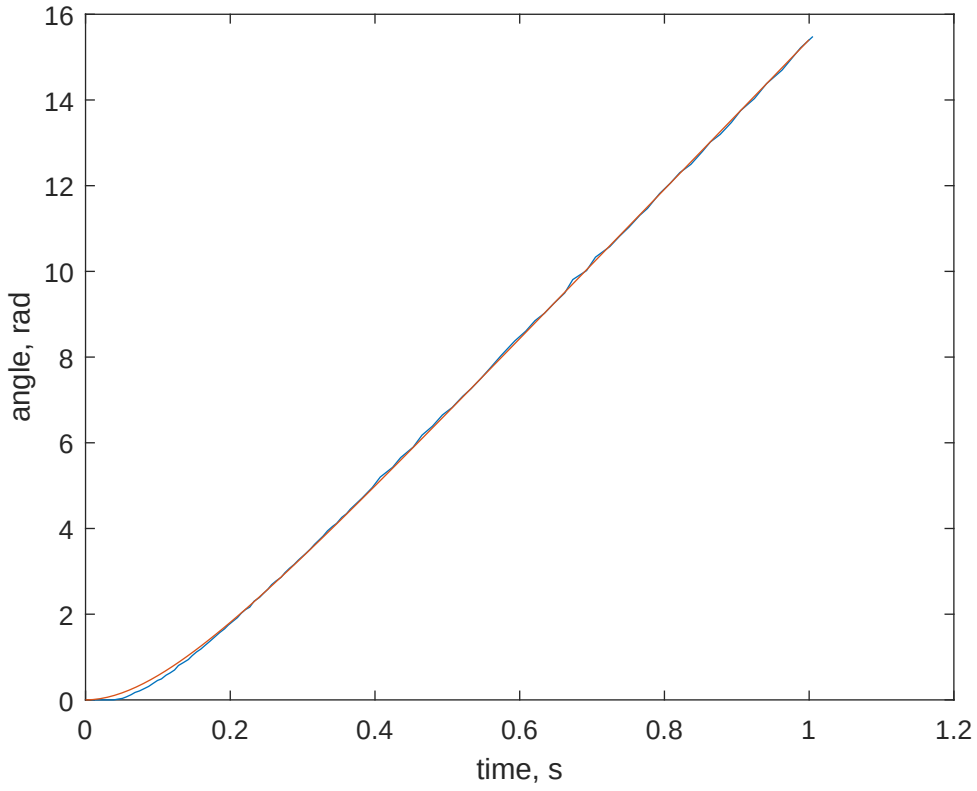

Рис. 4: График с аппроксимирующей кривой для зависимости угла от времени

Для аппроксимации зависимости угловой скорости от времени была выбрана функция (1). График вновь приведён для напряжения 100%.

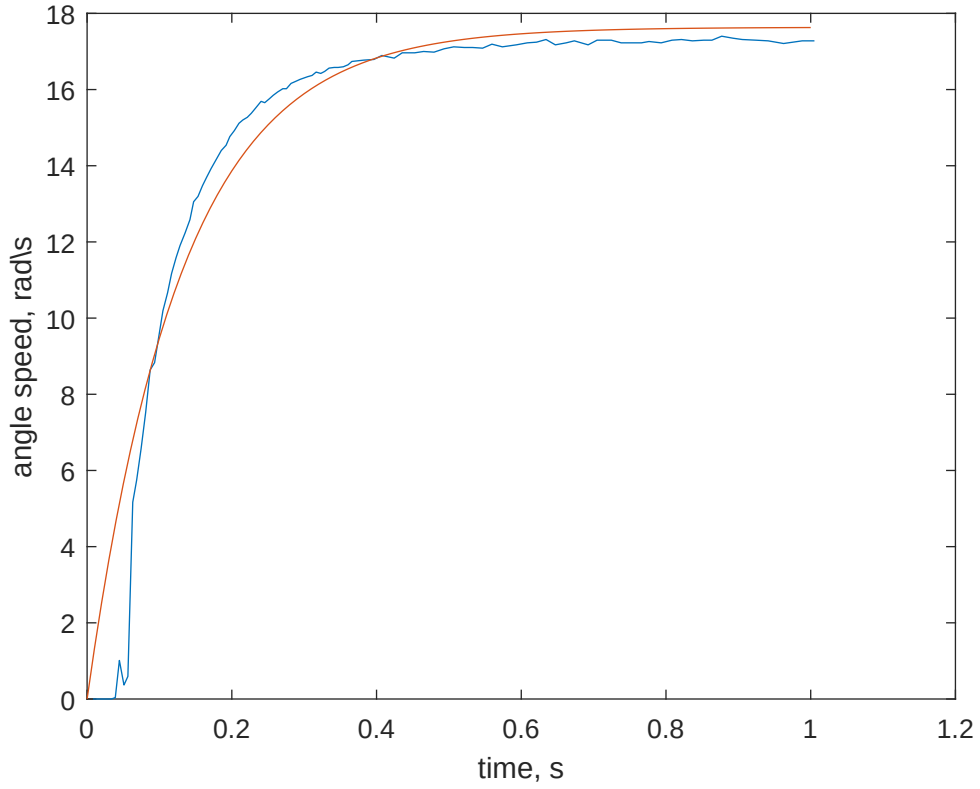

Рис. 5: График с аппроксимирующей кривой для зависимости угловой скорости от времени

4 Моделирование схемы в Simulink

Теперь нам предстояло смоделировать схему нашего процесса и убедиться в её корректности. Для этого нам нужно было сравнить график аппроксимирующей функции с графиком, построенным на основании результатов моделирования схемы 1.

С помощью Simulink мы получили графики, где показаны математическая модель и аппроксимирующая кривая. Опять же, в отчёте представлен график при напряжении в 100%.

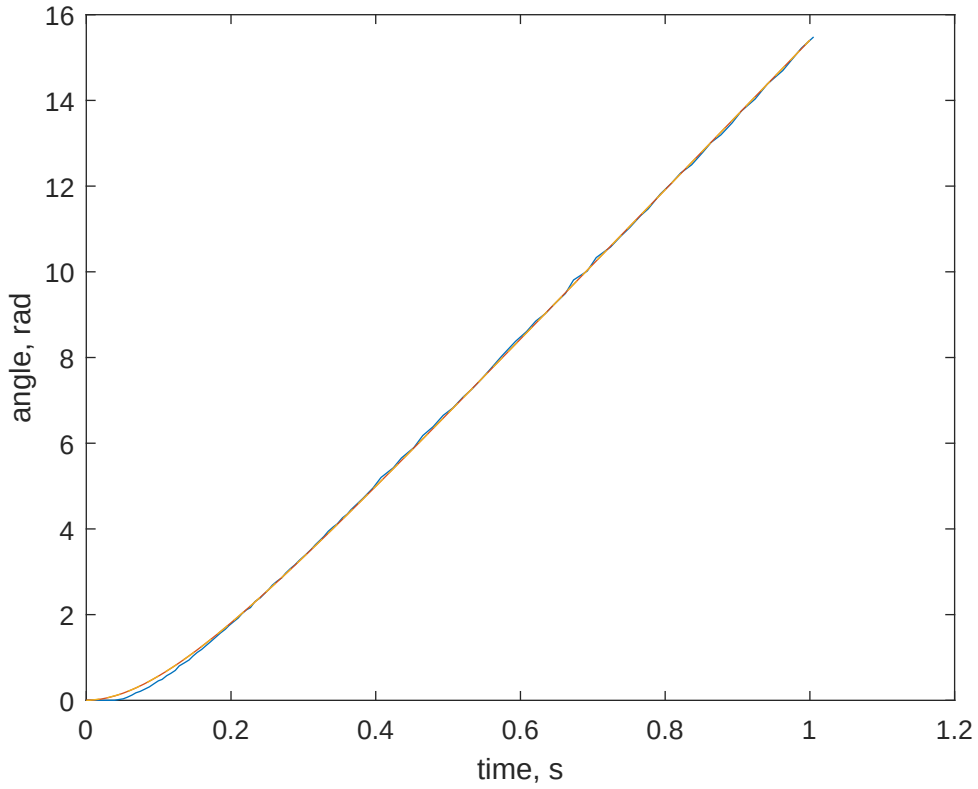

Рис. 6: Математическая модель и аппроксимирующая кривая для угла

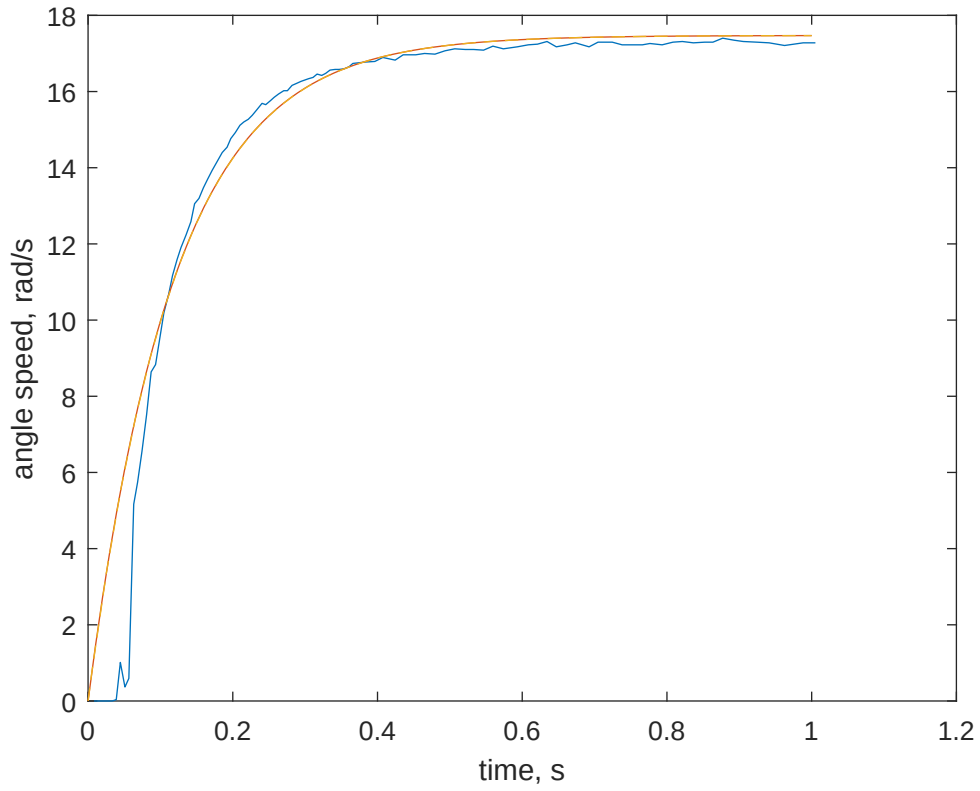

Рис. 7: Математическая модель и аппроксимирующая кривая для угловой скорости

Также удалось построить график зависимости коэффициента T_m от того, какое напряжение мы подаём:

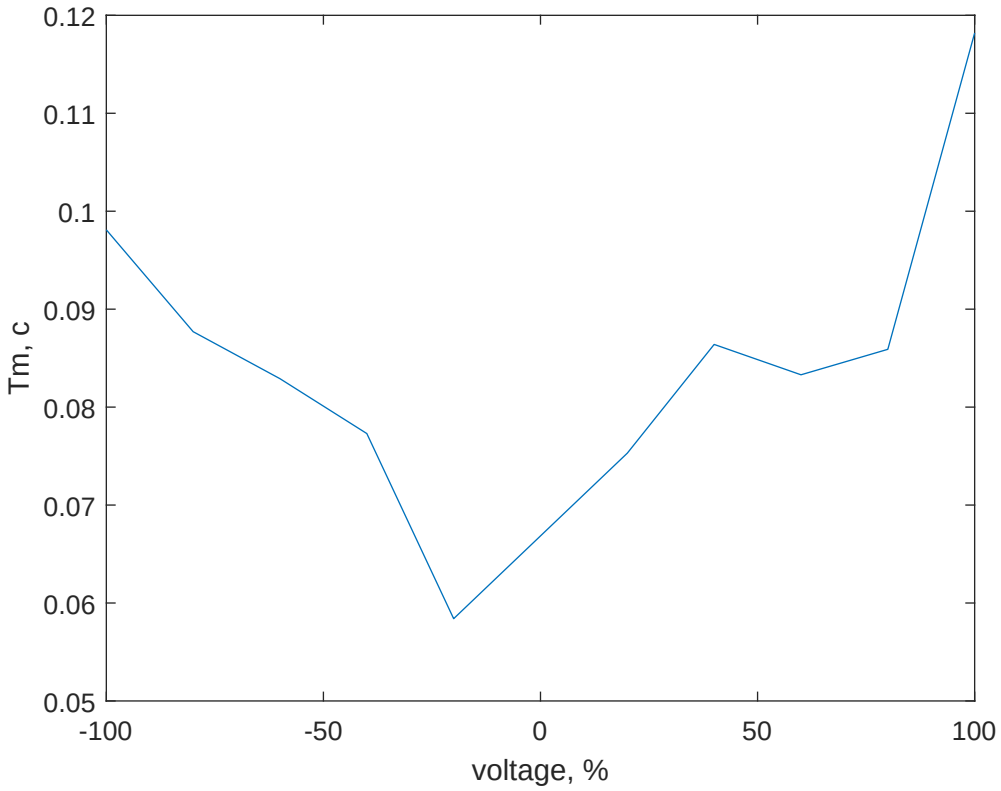

Рис. 8: Зависимость T_m от напряжения

Для построения графика зависимости скорости от напряжения необходимо было рассчитать $\omega_{ycm} = U_{pr} * k$. График полученной зависимости представлен ниже.

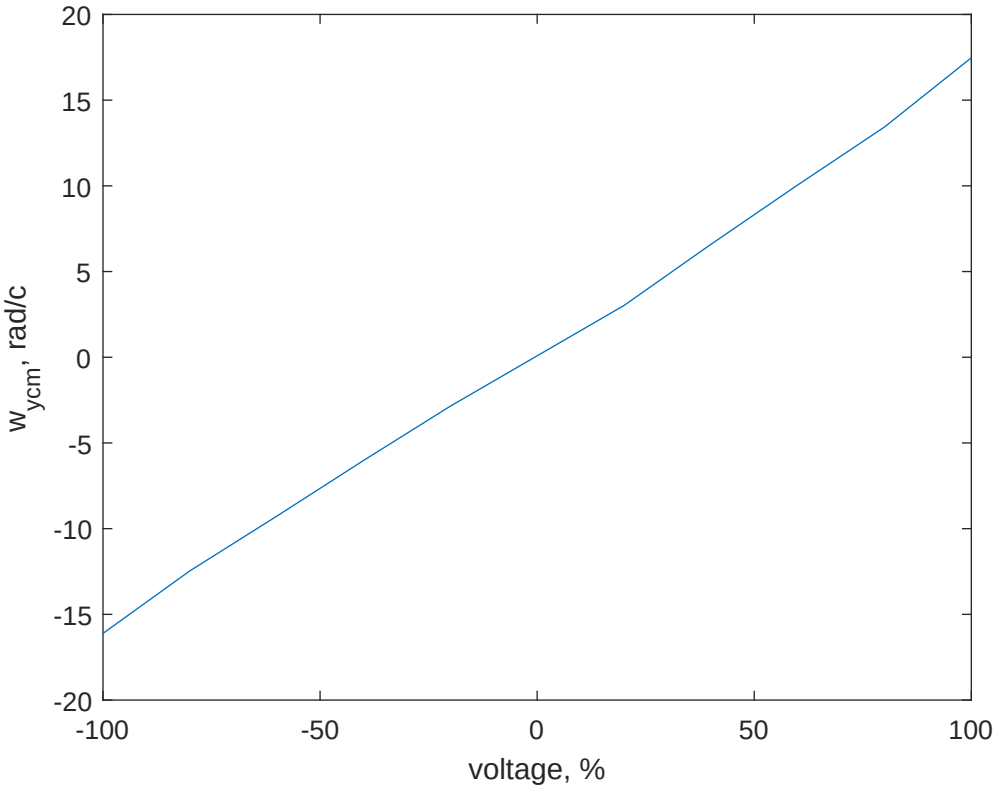

Рис. 9: Зависимость ω_{ycm} от напряжения

5 Выводы по работе

Анализируя проделанную работу, можно сказать, что у нас получилось выполнить цель и все поставленные задачи. Но мы столкнулись с некоторыми трудностями:

1. **Запуск программы на роботе.** В начале у нас возникла ошибка при попытке компиляции наших программ file.py и control.py на блоке EV3. Оказалось, что мы не написали в коде комментарий, который нужен для удачной компиляции. Исправить ошибку удалось, запустив программу на роботе напрямую через консоль.

2. **Получение данных.** Когда мы снимали показания, мы забыли дописать в коде для робота строчку, которая отвечает за то, чтобы мотор не работал непрерывно и между сменами напряжения была остановка. Из-за этого нами были получены неверные данные. Нам пришлось менять код, снова подключаться к роботу и снимать новые, уже правильные данные.

3. **Аппроксимация.** Отдельной проблемой было разобраться с аппроксимацией и как ее сделать в MatLab. Никто из нашей команды не умел работать в Матлабе, так что пришлось разбираться с нуля, используя методичку. У нас никак не получалось построить график аппроксимации для зависимости угла от времени, причём при напряжении в 100% всё получалось, а при других напряжениях - нет. Оказалось, что наши исходные графики должны исходить из нуля, а у нас они были смещены вверх по оси y, и это значение для каждого напряжения было разным. В итоге эту ошибку мы исправили, узнав значение смещения и вычтя его из расчёта угла (для каждого напряжения своё!). Тогда графики аппроксимации построились без проблем.

4. **Схема в Simulink.** У нас возникали сложности при построении математической модели, многие из них решались долгим разбором документации. Кроме того мы с толкнулись с похоже проблемой, что и при построении графиков аппроксимации: нам нужно было "сдвинуть"графики до нуля, чтобы построить корректную математическую модель.

Разобравшись со всеми проблемами, у нас получилось закончить работу, но самое главное, что мы получили очень ценный опыт, и все эти ошибки тоже помогли нам стать лучше.