

Введение

Целью прохождения производственной практики являлось закрепление теоретических и практических знаний, полученных при изучении дисциплин специальности и специализации, и приобретение навыков работы по избранной специальности.

Основными задачами являлись ознакомление с организацией студенческого конструкторского бюро, реализация курса лабораторных работ дисциплины “Введение в специальность”, автором которого является к.т.н. Капитонов А. А., на конструкторе Lego EV3, используя язык программирования Python.

Необходимость этого курса заключается в том, что конструктор Lego NXT во-первых, снят с производства и, во-вторых, новая модель конструктора обладает большими возможностями как для обучения, так и для научных исследований. На контроллере Lego EV3 установлена полноценная операционная система Linux Debian. Язык программирования python был выбран так как он достаточно широко распространен в научном сообществе по причине простоты, удобства и очень широкого круга задач, решаемых при помощи стандартных библиотек.

В процессе выполнения задания необходимо было закрепить теоретические знания в таких областях, как описание динамики робототехнических систем, идентификация параметров объекта управления и теория автоматического управления.

Основная часть

Производственную практику я проходила на кафедре СУиИ в студенческом конструкторском бюро (СКБ). СКБ является научно-исследовательским объединением студентов кафедры СУиИ, которое было создано для содействия в проведении научно-исследовательских и опытно-конструкторских работ и возможного внедрения результатов научно-технической деятельности в производство, с использованием образовательного и научного потенциала кафедры.

Основными целями, которые ставит перед собой СКБ, являются: интеграция студентов в сферу научных разработок, участие в соревнованиях по робототехнике и реализация студенческих проектов на базе оборудования кафедры.

В целом, нашу команду интересуют самые различные направления робототехники. Судя по выполненным за прошедший год задачам, к таковым можно отнести:

На сегодняшний день в СКБ реализуются широкий круг задач самых различных направлений робототехники, к ним относятся:

- Техническое зрение роботов;
- Картирование и локализация роботов (SLAM) (с помощью ИК датчика-дальномера и камер с глубинным изображением);
- Управление манипулятором;
- Футбол роботов.

В будущем в СКБ планируется участие в международных соревнованиях по робототехнике (в частности RoboCup и Eurobot).

Также СКБ является мощной базой для создания различной методической литературы, в том числе учебной. Моя задача заключалась в реализации курса лабораторных работ на новом конструкторе Lego EV3, с использованием широко распространенного в научной среде языка программирования Python.

Первым этапом моей работы было построение математической модели большого двигателя EV3. Для этого были измерены физические параметры, встроенного в мотор EV3, двигателя постоянного тока: масса ротора, радиус ротора, индуктивность обмотки ротора и сопротивление обмотки ротора, а также коэффициент редукции мотора.

Вторым этапом являлось получение конструктивных постоянных времени.

График зависимости угла поворота от времени представлен на рисунке 1. На графике сплошная линия представляет результаты реальной характеристики, а пунктирная линии – результаты моделирования.

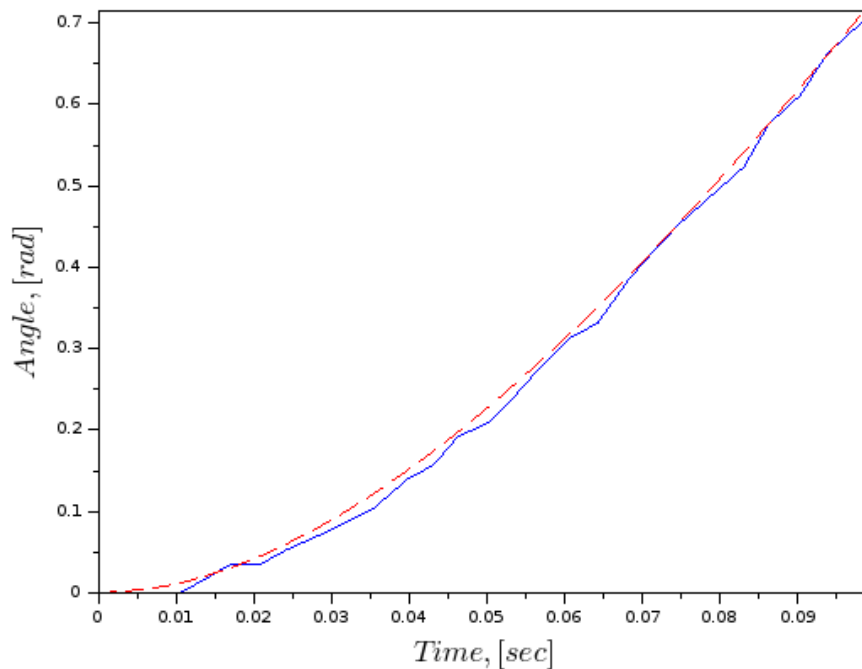


Рис. 1: Характеристика двигателя EV3

Из полученного графика были найдены установившаяся скорость мотора 16.32 радиан в секунду и электромеханическая постоянная времени 0.064 сек.

По полученным данным были сделаны выводы, что двигатели EV3 и NXT имеют одинаковые электромеханические параметры.

Третьим этапом стало написание программы управления двигателем на python. Был реализован пропорциональный регулятор и построены графики

стабилизации в заданном положении. На графике сплошной линией показана реальная характеристика, а пунктирной – результаты моделирования.

```
1#!/usr/bin/env python
2from ev3dev.ev3 import LargeMotor, \
3    OUTPUT_A, Sound, PowerSupply
4from time import time, sleep
5from math import pi, copysign
6
7N = 1000
8file_name = "data_pc.txt"
9
10Kp = 20
11pwr = 0
12
13motor = LargeMotor(OUTPUT_A)
14battery = PowerSupply()
15Sound().beep()
16fout = open(file_name, "w")
17sleep(0.05)
18motor.reset()
19print(motor.position)
20start_time = time()
21for i in range(0, N):
22    t = time() - start_time
23    rotation = motor.position
24    e = (360 - rotation) * pi / 180
25    pwr = Kp * e
26    pwr = pwr / battery.measured_volts * 100
27    if abs(pwr) > 100:
28        pwr = copysign(1, pwr) * 100
29    motor.run_direct(duty_cycle_sp=pwr)
30    line = "%f\t%d\n" % (t, rotation)
31    fout.write(line)
32fout.close()
```

Рис. 2: Пропорциональный регулятор на python

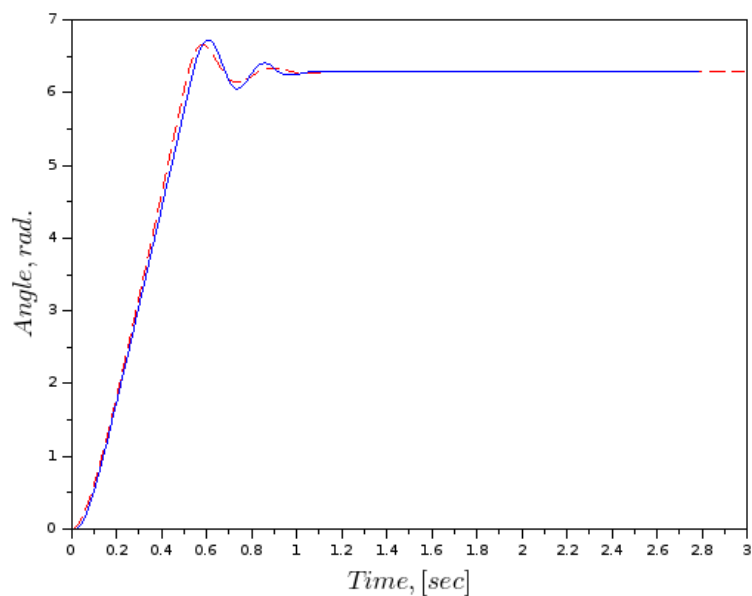


Рис. 3: Результаты работы пропорционального регулятора

На четвертом этапе, была построена математическая модель перевернутого маятника и рассчитан регулятор.

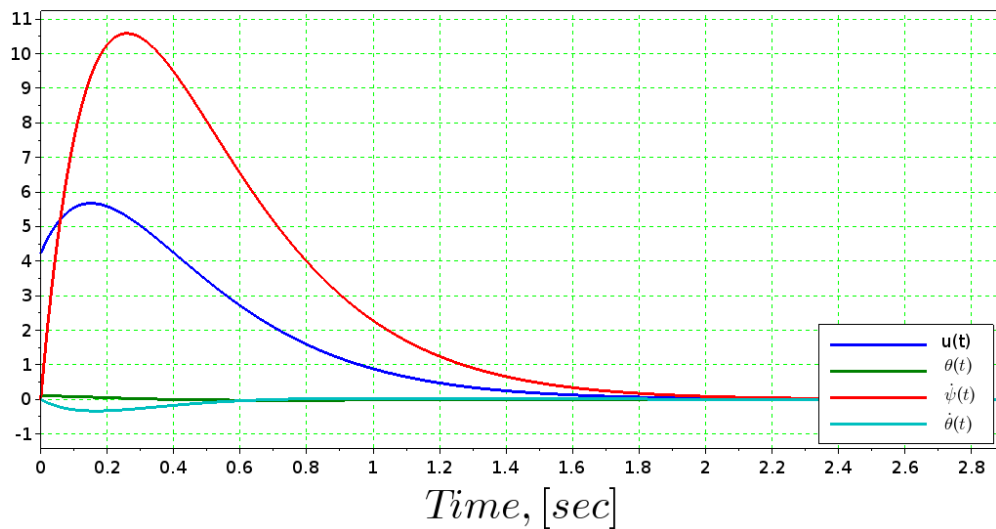


Рис. 4: Результаты моделирования перевернутого маятника на тележке

На пятом этапе была собрана реальная модель, написана программа управления на python и настроена.

Из особенностей, следует заметить, что регулятор был рассчитан в результате решения матричного уравнения Сильвестра.

Также, в существующих библиотеках python для Lego EV3 не была реализована возможность работы с датчиком угла поворота Lego HTAngleSensor, что было исправлено путем ее реализации.

```

1 #!/usr/bin/env python3
2 from ev3dev.auto import *
3 import time
4 import math
5 import AngleSensor
6
7 WAIT_TIME = 0.01
8 K_PSI = -29.6
9 K_DPSI = -0.55
10 K_DTHETA = -5.8
11
12 motorA = LargeMotor(OUTPUT_A)
13 motorB = LargeMotor(OUTPUT_B)
14 halt = Button()
15 angle = AngleSensor.Angle.get_obj()
16 offset_psi = 0
17 psi = 0
18 motorA.reset()
19 motorB.reset()
20 theta = motorA.position
21 Sound().beep()
22 time.sleep(0.5)
23 start_time = time.time()
24 while (True):
25     max_voltage = PowerSupply().measured_volts
26     if halt.enter:
27         motorA.duty_cycle_sp = 0
28         motorB.duty_cycle_sp = 0
29         raise SystemExit
30     last_psi = psi
31     last_theta = theta
32     psi = angle.get_angle() + 1
33     if psi > 180:
34         psi = psi - 360
35     theta = motorA.position
36     dtheta = (theta - last_theta) / WAIT_TIME
37     dps_i = (psi - last_psi) / WAIT_TIME
38     u = K_PSI * psi + K_DPSI * dps_i + K_DTHETA * dtheta
39     u = u * 100 / max_voltage
40     if abs(u) > 100:
41         u = math.copysign(1, u) * 100
42     motorA.run_direct(duty_cycle_sp=-u)
43     motorB.run_direct(duty_cycle_sp=u)
44     #time.sleep(WAIT_TIME)
45     t = time.time() - start_time

```

Рис. 5: Программа стабилизации перевернутого маятника на тележке для Lego EV3

Заключение

В результате проделанной работы, был полностью повторен курс лабораторных работ для более нового набора Lego EV3. Было отмечено, что разработка программы на python под привычную операционную может быть не только полезным, а также может приносить массу удовольствия от самого процесса. На контроллере был настроен репозиторий и система онлайн отладки программы. Подключение контроллера к компьютеру производилось по беспроводной сети WiFi, что существенно упрощает отладку, снятие показаний работы робота, просмотр логов и статистики работы программы.

В последствии стоит, переработать написанные программы в соответствии сегодняшними тенденциями развития языков программирования на python3 с применением парадигмы программирования ООП.

Также необходимо будет настроить операционную систему ROS на контроллере LegoEV3, произвести тесты производительности, в случае позитивных показаний которых, возможно будет реализовать техническое зрение для этого контроллера.