# Лабораторная работа №1
# Построение математической модели двигателя EV3

## 1 Методические рекомендации

До начала работы студент должен установить на компьютере необходимый для работы с набором LEGO Mindstorms EV3 софт и настроить подключение к интернету. Подробно процесс подключения описан в приложении (п. 6).

## 2 Теоретические сведения

В данном пособии изложены основные принципы функционирования неотъемлемой части многих робототехнических устройств — электродвигателя постоянного тока.

### Математическая модель двигателя постоянного тока

Основными частями любого двигателя постоянного тока являются *статор* (или индуктор) и *ротор* (или якорь). Статор представляет собой неподвижную часть двигателя, проще говоря, его корпус, который является постоянным магнитом. Ротор же — ту составляющую, которая приводится во вращение.

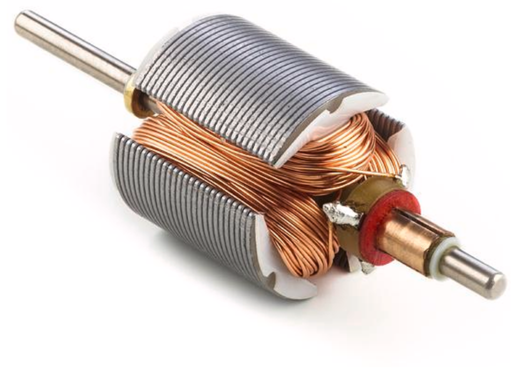

Рис. 1. Внешний вид ротора

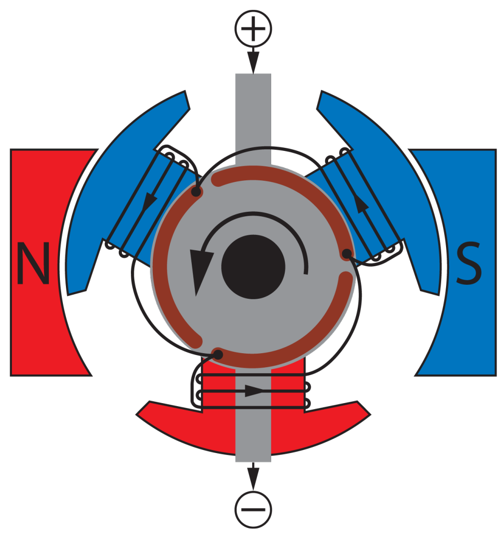


Рис. 2. Схематичное изображение двигателя

Ротор (рис. 1) состоит из нескольких катушек, намотанных на магнитомягкий материал. При подаче напряжения на катушки у ротора появляется собственное магнитное поле, которое при взаимодействии с полем статора приводит ротор в движение (рис. 2). При повороте ротора щетки, изображенные на рисунке 2 коричневым цветом, поворачиваются вместе с ним, а контакты, на которые подается входное напряжение, остаются на месте, из-за чего при повороте на определенный угол напряжение подается на следующие две щетки и ток в катушках меняет направление, что позволяет ротору продолжить вращение.

Для дальнейшего теоретического рассмотрения исследуемого объекта (двигателя), в первую очередь следует составить *математическую модель* его работы. Под ней понимается совокупность уравнений, описывающих изменения характеризующих объект величин и их взаимосвязь друг с другом.

Для составления математической модели рассмотрим несколько физических законов:

**Второй закон Ньютона** для вращательного движения имеет вид

$$M = J\dot{\omega}, \tag{1}$$

где $M, \textit{Н·м}$ — суммарный вращательный момент, действующий на ротор; $J, \textit{кг·м}^2$ — момент инерции ротора[1]; $\dot{\omega}, \textit{рад}/\textit{c}^2$ — угловое ускорение ротора.[2]

Величину $M$ можно представить суммой двух слагаемых

$$M = M_{el} + M_{oth}, \tag{2}$$

где $M_{el}$ — момент силы, возникающий в двигателе из-за протекающих в нем электродинамических процессов и раскручивающий его ротор; $M_{oth}$ — сумма всех остальных моментов сил, действующих на ротор. К последним можно отнести моменты сил трения и моменты, создаваемые весами каких-либо конструктивных элементов, прикрепленных к валу двигателя. Поскольку мы рассматриваем ситуацию, когда ротор двигателя ничем не нагружен, и полагаем силы трения отсутствующими, то

$$M_{oth} = 0 \quad \Rightarrow \quad M = M_{el}. \tag{3}$$

**Закон Ампера** устанавливает связь между силой тока в проводнике и силой, действующей на проводник со стороны внешнего магнитного поля. Однако при рассмотрении двигателя постоянного тока необходимо учитывать вращающий момент, действующий на весь ротор в целом, а не только силу, действующую на его обмотку. В связи с этим необходимо ввести понятие магнитного момента.

*Магнитный момент* — основная физическая величина, характеризующая магнитные свойства вещества, то есть способность создавать и воспринимать магнитное поле. В нашем случае магнитным моментом обладают катушки и сердечник (тело ротора), а общий магнитный момент ротора вычисляется как их сумма:

$$m = NIS + NI\left(\mu - 1\right)S = \mu NIS, \tag{4}$$

где $m, \textit{Вб} \cdot \textit{м}$ — магнитный момент ротора; $I, A$ — сила тока в катушках; $N$ — количество витков в катушках; $S, \textit{м}^2$ — площадь одного витка; $\mu$ — относительная магнитная проницаемость сердечника (тела ротора).

[1] При рассмотрении вращательного движения тела этот параметр играет ту же роль, что и масса тела при рассмотрении его поступательного движения.

[2] Точкой над какой-либо переменной обозначается её производная по времени. В данном случае $\omega, \textit{рад}/\textit{c}$ есть угловая скорость вращения ротора.

Момент силы, создаваемый внешним по отношению к ротору магнитный полем статора с индукцией $B_0$, будет равен:

$$M_{el} = mB_0 \sin\varphi,\text{[3]} \tag{5}$$

где $m$, *Вб·м* — магнитный момент ротора; $B_0$, *Тл* — величина магнитного поля статора снаружи ротора; $\varphi$, *рад* — угол между направлением намагниченности ротора и направлением магнитного поля статора.

Благодаря тому, что катушки при вращении ротора постоянно переключаются, величина $\sin\varphi$ остается приблизительно постоянной (см. рис. 4). Благодаря этому, можно усреднить значение синуса и принять момент силы пропорциональным магнитному моменту ротора (так как магнитная индукция $B_0$ постоянна). Из уравнения (4) видно, что магнитный момент ротора пропорционален силе тока в катушках. Тогда можно сказать, что и момент силы, действующий на ротор со стороны поля статора, будет пропорционален силе тока в катушках, то есть

$$M_{el} = k_m I, \tag{6}$$

где $M$, *Н·м* — вращательный момент, действующий на ротор; $I$, $A$ — сила тока в катушках; $k_m$ — некоторый коэффициент пропорциональности.

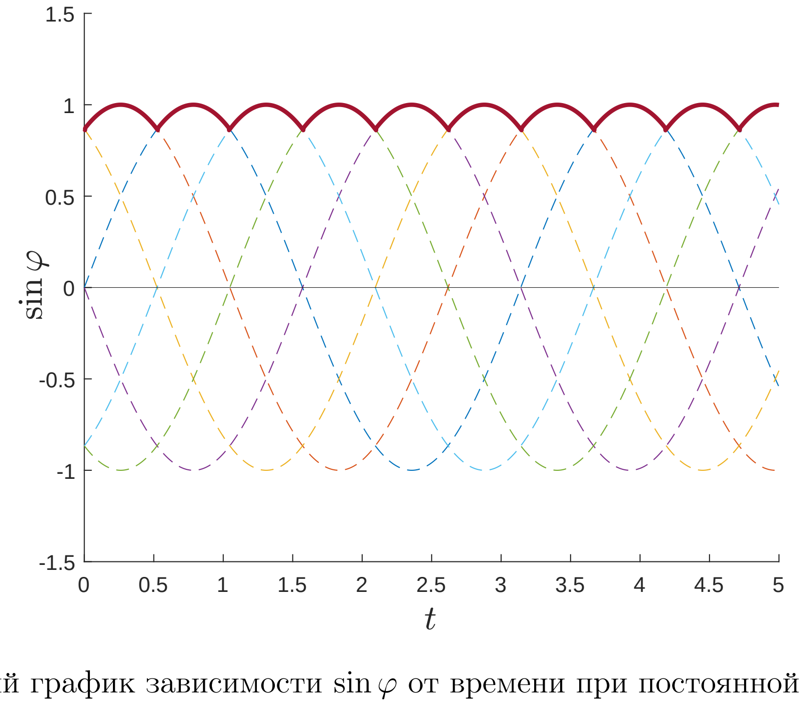


Рис. 3. Примерный график зависимости $\sin\varphi$ от времени при постоянной угловой скорости

**Электромагнитная индукция** является одним из самых важных явлений, возникающих в двигателе постоянного тока. Суть его заключается в том, что вращение двигателя создает в катушках ротора дополнительную ЭДС, которая приводит к возникновению дополнительной составляющей силы тока, призванной остановить вращение двигателя. Одним из проявлений

[3] Не вдаваясь в детали, заметим, что эта формула связана с *векторным произведением* векторов. Если $\vec{c} = \vec{a} \times \vec{b}$, то $|\vec{c}| = |\vec{a}||\vec{b}| \sin\alpha$, где $\alpha$ — угол между векторами $\vec{a}$ и $\vec{b}$.

электромагнитной индукции является тот факт, что при отключении напряжения двигатель постоянного тока затормозит и полностью остановится *даже при полном отсутствии трения*.

Рассмотрим отдельный виток одной из катушек как замкнутый контур. ЭДС индукции для него описывается выражением

$$\varepsilon_{\text{виток}} = \pm\frac{d\varPhi}{dt}, \tag{7}$$

где $\varepsilon_{\text{виток}}, B$ — ЭДС индукции, наводимая в одном витке катушки; $\varPhi, \textit{Вб}$ — магнитный поток поля статора, проходящий через контур. Знак в (7) определяется по правилу Ленца (ЭДС индукции должна действовать против причины, ее вызвавшей), мы определим его в конце рассуждения. Магнитный поток определяется выражением

$$\varPhi = BS\cos\varphi = \mu B_0 S\cos\varphi,\text{[4]} \tag{8}$$

где $B, \textit{Тл}$ — величина магнитного поля статора внутри ротора; $S, \textit{м}^2$ — площадь контура; $\varphi, \textit{рад}$ — угол между направлением магнитного потока и нормалью к контуру (он совпадает с углом $\varphi$, введенным ранее); $\mu$ — относительная магнитная проницаемость сердечника (тела ротора); $B_0$, *Тл* — величина магнитного поля статора снаружи ротора;.

Подставив выражение (8) в выражение (7), учитывая, что магнитный поток поля статора и площадь контура остаются постоянными в то время как виток поворачивается, получим

$$\varepsilon_{\text{виток}} = \pm BS\frac{d\cos\varphi}{dt} = \mp BS\dot{\varphi}\sin\varphi. \tag{9}$$

Таким образом, в отдельном витке наводится синусоидальная ЭДС, амплитуда которой пропорциональна величине $\dot{\varphi}$, которая в свою очередь равна угловой скорости вращения ротора. Суммарная ЭДС в катушке равна сумме ЭДС в отдельных витках, то есть:

$$\varepsilon_i = \mp NBS\omega\sin\varphi, \tag{10}$$

где $\varepsilon_i, B$ — ЭДС индукции, наводимая в активной катушке ротора; $N$ — количество витков; $\omega, \textit{рад}/\textit{с}$ — угловая скорость ротора.

Как уже было отмечено, при вращении двигателя катушки постоянно переключаются (см. рис. 2) и величина $\sin\varphi$ остается приблизительно постоянной (см рис. 4). Благодаря этому можно сказать, что ЭДС индукции пропорциональна угловой скорости ротора. Воспользовавшись правилом Ленца (ЭДС индукции должна стремиться затормозить двигатель, чтобы устранить причину, ее вызвавшую), выбираем знак «–» и получаем окончательную версию закона электромагнитной индукции для двигателя постоянного тока:

$$\varepsilon_i = -k_e\omega, \tag{11}$$

где $\varepsilon_i, B$ — ЭДС индукции в обмотках ротора, вызываемая полем статора; $\omega, \textit{рад}/\textit{с}$ — угловая скорость ротора; $k_e$ — некоторый коэффициент пропорциональности.

[4] В свою очередь, эта формула связана со *скалярным произведением* векторов. Если $c = \vec{a}\cdot\vec{b}$, то $c = |\vec{a}||\vec{b}|\cos\alpha$, где $\alpha$ — угол между векторами $\vec{a}$ и $\vec{b}$.

Наконец, рассмотрим **обобщенный закон Ома** для участка цепи

$$I = \frac{\mathcal{E}}{R}, \tag{12}$$

где $R$, *Ом* — полное сопротивление обмоток ротора; $\mathcal{E}$, *В* — суммарная ЭДС в обмотках.

Суммарная ЭДС состоит из подаваемого на двигатель напряжения (разницы потенциалов на клеммах ротора) и каких-либо других причин, вызывающих ток. В нашем случае ток вызывает не только разница потенциалов, но и ЭДС индукции со стороны поля статора, так что закон Ома приобретает вид

$$I = \frac{\varepsilon_i + U}{R}, \tag{13}$$

где $R$, *Ом* — полное сопротивление обмоток ротора; $U$, *В* — напряжение, которое мы подаем на двигатель; $\varepsilon_i$, *В* — ЭДС индукции.

На основе описанных выше четырех физических законов мы можем составить математическую модель двигателя постоянного тока. Для начала подставим выражение для ЭДС индукции (11) в выражение для силы тока (13), и выражение для момента (6) во второй закон Ньютона (1). Получим

$$I = \frac{-k_e\omega + U}{R}, \qquad J\dot{\omega} = k_m I. \tag{14}$$

Давайте по этим уравнениям представим, как будет происходить движение ротора. В начальный момент времени двигатель покоится, угловая скорость равна нулю. Мы подаем на двигатель напряжение, и в нем появляется ток, согласно первому уравнению равный $\frac{U}{R}$[5]. Из второго уравнения мы можем понять, что появление тока приведет к появлению ускорения. Таким образом, скорость начнет увеличиваться, следовательно начнет увеличиваться (в сторону отрицательных значений) слагаемое $-k_e\omega$, то есть ЭДС индукции, что приведет к уменьшению тока и уменьшению ускорения. Скорость продолжит расти, но медленнее. ЭДС индукции продолжит расти вместе со скоростью, пока не придет к значению, по модулю равному поданному напряжению $U$. В этот момент ток станет равен нулю, следовательно, ускорение тоже превратится в ноль, а скорость примет постоянное значение – это будет означать, что процесс разгона завершен, двигатель достиг установившейся угловой скорости.

Объединим получившиеся формулы (14) в одно выражение

$$J\dot{\omega} = k_m \left( \frac{-k_e\omega + U}{R} \right). \tag{15}$$

Раскроем скобки, поделим на $J$ и перенесем слагаемое с $\omega$ в левую часть равенства. Получим выражение

$$\dot{\omega} + \frac{k_m k_e}{JR}\omega = \frac{k_m}{JR}U, \tag{16}$$

которое и принято считать математической моделью двигателя. Оно устанавливает зависимость между величинами, описывающими движение ротора: используя это выражения, мы можем по

[5] В данной лабораторной работе мы считаем, что ток появляется мгновенно, однако на самом деле это не так. Более подробно данное явление будет рассмотрено во второй лабораторной работе.

известной функции подаваемого на двигатель напряжения $U(t)$ вычислить угловую скорость двигателя $\omega(t)$ в любой момент времени.

Полученное выражение является *дифференциальным уравнением*, так как в нем присутствует производная по времени от одной из величин (угловой скорости). Дифференциальными уравнениями описываются очень многие явления. Одним из самых известных примеров является второй закон Ньютона, который содержит ускорение, то есть производную от скорости.

Предположим, что на двигатель подается постоянное напряжение $U$. Найдем решение уравнения (16) относительно угловой скорости, чтобы узнать, как она будет зависеть от времени (то есть, как будет проходить разгон двигателя). Выразим $\dot{\omega}$ в левой части как

$$\dot{\omega} = \frac{k_m}{JR}(U - k_e\omega) \tag{17}$$

и представим уравнение в виде

$$\frac{d\omega}{dt} = \frac{k_m}{JR}(U - k_e\omega). \tag{18}$$

Умножив обе части на $dt$ и потом разделив на выражение в скобках, получим

$$\frac{d\omega}{(U - k_e\omega)} = \frac{k_m}{JR}dt. \tag{19}$$

Проинтегрируем обе части выражения

$$\int \frac{d\omega}{(U - k_e\omega)} = \int \frac{k_m}{JR}dt \tag{20}$$

и получим[6]:

$$-\frac{1}{k_e}\ln|U - k_e\omega| + c = \frac{k_m}{JR}t. \tag{21}$$

Знак модуля можно опустить, так как значение ЭДС индукции $k_e\omega$ не превышает напряжения. Домножим выражение на $-k_e$ и представим $k_e c$ как новую константу $-\ln C$[7]:

$$\ln(U - k_e\omega) + \ln C = -\frac{k_m k_e}{JR}t \tag{22}$$

Возьмем от обеих частей выражения показательную функцию:

$$e^{\ln(U - k_e\omega) + \ln C} = e^{-\frac{k_m k_e}{JR}t} \quad \Rightarrow \quad e^{\ln(U - k_e\omega)}e^{\ln C} = e^{-\frac{k_m k_e}{JR}t}. \tag{23}$$

Получим

$$(U - k_e\omega)C = e^{-\frac{k_m k_e}{JR}t}. \tag{24}$$

Из начальных условий — в начальный момент времени ($t = 0$) двигатель покоится ($\omega = 0$) — определим $C$:

$$(U - k_e \cdot 0)C = e^{-\frac{k_m k_e}{JR}\cdot 0} \qquad \Rightarrow \qquad C = \frac{1}{U}. \tag{25}$$

[6]Чтобы вычислить интеграл в левой части, заменим $U - k_e\omega = x$, тогда $\omega = \frac{U}{k_e} - \frac{x}{k_e}$, возьмем дифференциал от $\omega$: $d\omega = -\frac{1}{k_e}dx$. Далее подставим это в интеграл: $\int \frac{d\omega}{(U-k_e\omega)} = \int -\frac{1}{k_e}\frac{dx}{x} = -\frac{1}{k_e}\ln|x| + c$.

[7]Мы можем это сделать так как областью значений логарифма являются все действительные числа, то есть он может принимать такие же значения, как и $k_e c$. Форма $-\ln C$ удобнее для нас, так как $e^{\ln C} = C$

Таким образом, выражение для угловой скорости принимает вид

$$\omega(t) = \frac{U}{k_e}\left(1 - e^{-\frac{k_m k_e}{JR}t}\right). \tag{26}$$

Из выражения (26) мы можем найти установившуюся скорость двигателя $\omega_{уст}$, для этого возьмем от него предел при $t$ стремящемся к бесконечности:

$$\omega_{уст} = \lim_{t\to\infty} \omega(t) = \lim_{t\to\infty} \frac{U}{k_e}\left(1 - e^{-\frac{k_m k_e}{JR}t}\right). \tag{27}$$

Так как $\frac{U}{k_e}$ не зависит от времени, мы можем вынести его за знак предела. Предел от единицы равен единице, а предел от экспоненты равен нулю. Тогда получаем

$$\omega_{уст} = \frac{U}{k_e}. \tag{28}$$

Заметим, что установившуюся скорость можно было вычислить еще из дифференциального уравнения (16). Для этого нужно посмотреть, что будет с выражением, когда ускорение станет равным нулю (примем $\omega = \omega_{уст}$; $\dot{\omega} = 0$):

$$\frac{k_m k_e}{JR}\omega_{уст} = \frac{k_m}{JR}U \qquad \Rightarrow \qquad \omega_{уст} = \frac{U}{k_e}. \tag{29}$$

Введем еще одну величину, характеризующую двигатель постоянного тока — **электромеханическую постоянную времени** $T_m$, определяемую выражением

$$T_m = \frac{JR}{k_m k_e}. \tag{30}$$

С использование введенных обозначений выражение (26) примет вид

$$\omega(t) = \frac{U}{k_e}\left(1 - e^{-\frac{t}{T_m}}\right) = \omega_{уст}\big(1 - e^{-\frac{t}{T_m}}\big). \tag{31}$$

Забегая вперед, стоит заметить, что при выполнении работы в программе для управления двигателем на него будет подаваться не напряжение в вольтах, а проценты от максимально возможного напряжения, так что для удобства $U$ можно представить как

$$U = \frac{U_{\%}}{100}U_{max}, \tag{32}$$

где $U_{\%}$ — процент от максимального напряжения; $U_{max}$, $B$ — максимальное напряжение. Считая величину $\frac{U_{max}}{100}$ постоянной, мы можем объединить ее с $k_e$ в новую константу $k$:

$$\frac{U}{k_e} = \frac{U_{\%}}{100k_e}U_{max} = kU_{\%}. \tag{33}$$

Тогда выражение для угловой скорости приобретает вид

$$\omega(t) = kU_{\%}\big(1 - e^{-\frac{t}{T_m}}\big). \tag{34}$$

Как видно из выражения (31), величина $T_m$ показывает время, за которое скорость вращения первоначально покоящегося ротора возрастает до

$$\omega(T_m) = \omega_{уст} - \frac{\omega_{уст}}{e}. \tag{35}$$

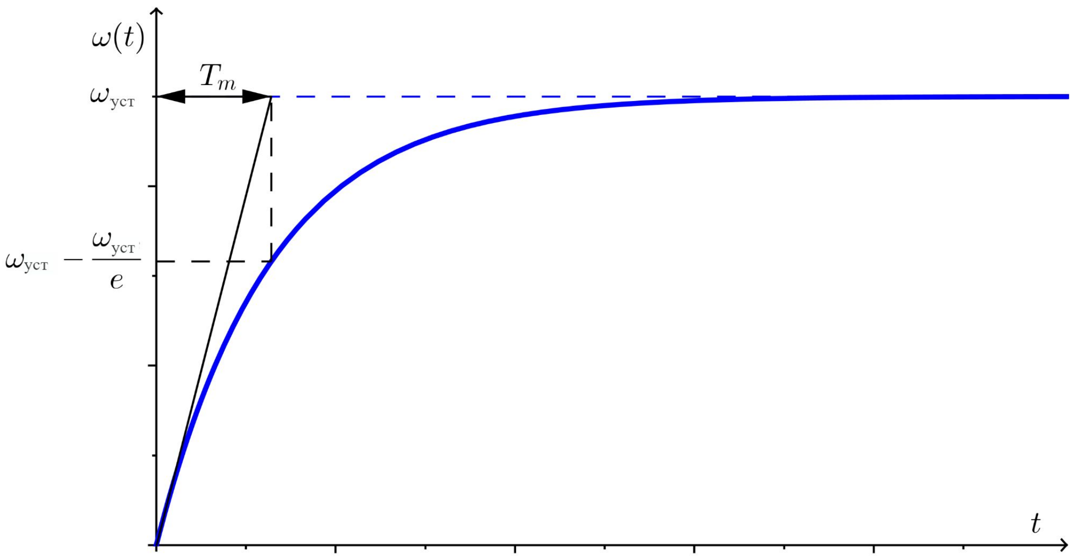


Рис. 4. График зависимости угловой скорости вращения ротора от времени.

С учетом введенных обозначений уравнение (16) можно записать в виде

$$T_m\dot{\omega} + \omega = \frac{1}{k_e}U, \tag{36}$$

или (считая, что напряжение подается в процентах от максимального)

$$T_m\dot{\omega} + \omega = kU_{\%}. \tag{37}$$

Из найденной зависимости получим выражения для углового ускорения ротора двигателя ($\alpha$) и его угловой координаты ($\theta$):

$$\alpha(t) = \dot{\omega} = \frac{\omega_{уст}}{T_m}e^{-\frac{t}{T_m}}, \tag{38}$$

$$\theta(t) = \int \omega dt = \omega_{уст}t + \omega_{уст}T_m e^{-\frac{t}{T_m}} + C, \tag{39}$$

где $C$ — некоторая постоянная, чье значение можно определить из начальных условий ($\theta(0) = 0$) в виде

$$\theta(0) = \omega_{уст} \cdot 0 + \omega_{уст}T_m e^{-\frac{0}{T_m}} + C \qquad \Rightarrow \qquad C = -\omega_{уст}T_m. \tag{40}$$

С учетом найденного значения постоянной $C$ окончательно имеем

$$\alpha(t) = \frac{\omega_{уст}}{T_m}e^{-\frac{t}{T_m}} = \frac{U}{k_e T_m}e^{-\frac{t}{T_m}}, \tag{41}$$

$$\theta(t) = \omega_{уст}\left(t - T_m\left(1 - e^{-\frac{t}{T_m}}\right)\right) = \frac{U}{k_e}\left(t - T_m\left(1 - e^{-\frac{t}{T_m}}\right)\right). \tag{42}$$

Если учесть подачу напряжения в процентах, то выражения примут вид

$$\alpha(t) = \frac{kU_{\%}}{T_m}e^{-\frac{t}{T_m}}, \tag{43}$$

$$\theta(t) = kU_{\%}\left(t - T_m\left(1 - e^{-\frac{t}{T_m}}\right)\right). \tag{44}$$

Приблизительные графики соответствующих функций представлены на рис. 5.

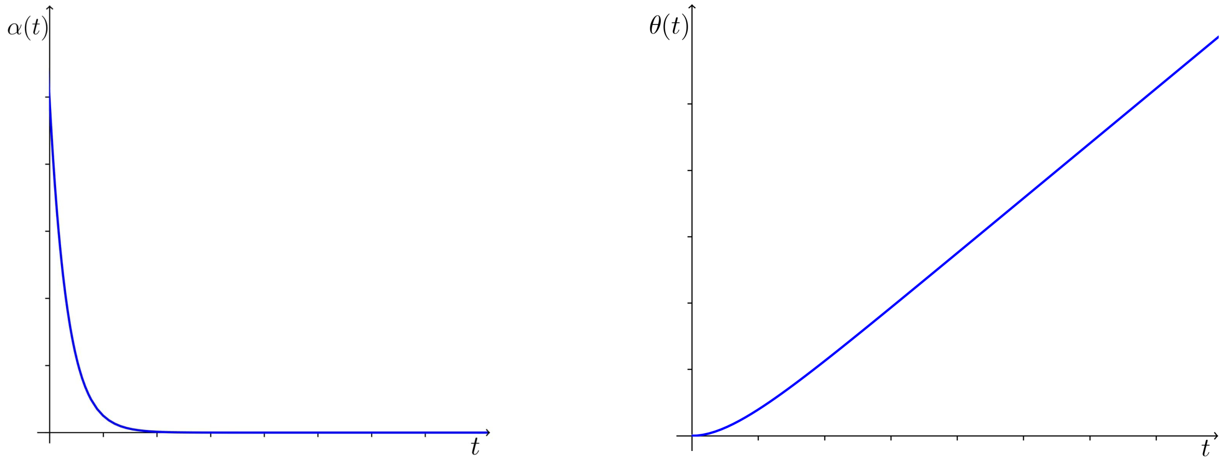


Рис. 5. Графики зависимостей $\alpha(t)$ и $\theta(t)$.

Целью практической части настоящей работы будет опытная проверка результатов, полученных в этом разделе.

**Моделирование работы двигателя**

Для получения из математической модели количественной информации о рассматриваемом процессе, необязательно прибегать к решению дифференциальных уравнений. С указанной задачей успешно справляются ряд вычислительных программ, подобных Matlab Simulink, которую мы будем использовать в данном курсе. Их можно использовать следующим образом.

Во-первых, на основании составленной математической модели исследуемого процесса в упомянутых программах строится блок-схема, описывающая единичный такт циклически повторяющихся математических операций, соответствующих изменению характеризующих процесс величин. Созданная в Simulink на основании уравнения (37) для ранее рассмотренного нами разгона ненагруженного двигателя эта схема будет выглядеть, как показано на рис. 6[8].

Во-вторых, по полученной схеме производится моделирование процесса и непосредственно сбор всех интересующих исследователя данных.

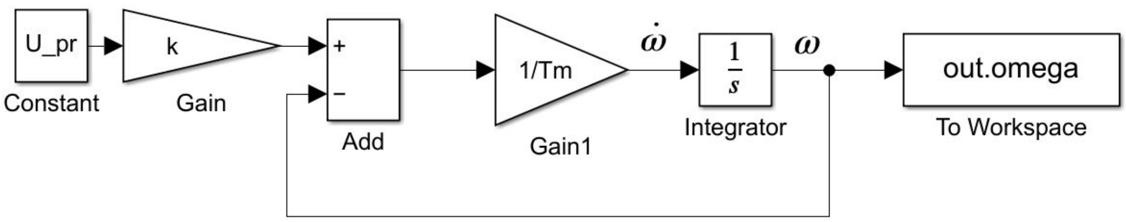


Рис. 6. Схема моделирования исследуемого процесса.

Для раскрытия принципов работы данного подхода скажем следующее.

Каждый блок, составляющий схему, имеет в своем составе *входы* и *выходы*, обозначенные стрелочками, направленными, соответственно, в тело блока и от него. Общение блоков

[8] Некоторые несущественные детали схемы, такие, как, например, форма некоторых блоков, могут различаться в зависимости от используемого программного обеспечения и его версий.

осуществляется с помощью сигналов — определенных числовых значений. Последние, перемещаясь по каналам связи (обозначены черными линиями), приходят на вход блока, в нем изменяются и уже в измененной форме поступают на его выход для дальнейшего путешествия по каналам связи.

Начинающий эту схему блок с буквенным обозначением «U_pr» и с одним выходом означает, что на последний подается значение, хранящееся в переменной U_pr (это значение подаваемого на двигатель напряжения в процентах). Блоки в форме треугольника умножают входящий сигнал на значение написанного на нем выражения и подают полученный результат на выход. Блок «Add» вычитает из своего сигнала, идущего на вход, помеченный знаком «+», другой сигнал (идущий на вход, помеченный знаком «−») и подает на свой выход значение этой разности. Блок с надписью «1/s» интегрирует входной сигнал и прибавляет к числовому значению, хранящемуся в его памяти.[9] Эта сумма затем поступает на его выход. Блок «To workspace» необходим сохранения результатов моделирования, отнесенных к заданным промежуткам времени, в используемой среде разработки (данные будут доступны в командной строке Matlab из переменной out.omega.Data и out.omega.Time после запуска симуляции).

Теперь покажем, как данная схема расшифровывается (читается). На блок-сумматор (под названием «Add») приходят два сигнала: один поступает из блока под названием Gain, в котором значение поданного напряжния в процентах умножается на $k$, второй — с конца схемы (выхода интегратора) и заключает в себе хранящееся в его памяти число. Эти сигналы вычитаются друг из друга, и полученная разность приходит на следующий блок, где умножается на значение $1/T_m$. В таком виде она идет дальше. Таким образом, на блок с надписью «1/s» поступает значение, равное $\omega_{уст}/T_m$, которое после интегрирования, согласно (37), даст не что иное, как некоторое значение угловой скорости вращения ротора $\omega_1$ в «первый» момент времени.[10] Этот сигнал, в свою очередь, пойдет в память среды Simulink и на один из входов сумматора. С учетом этого имеем, что при втором и последующих обходах сигналами схемы на выходе последнего будет формироваться значение $\omega_{уст}/T_m - \omega_i$,

[9]Имеется в виду следующее. Как известно, определенный интеграл дает лишь приращение первообразной $F(x)$ своей функции-«аргумента» $f(x)$, то есть:

$$\int_a^b f(x)\,dx = F(b) - F(a). \tag{45}$$

При таком условии для того чтобы иметь возможность вычислять именно значение первообразной в интересующей точке $x = b$, надо знать ее значение в начальной точке $x = a$:

$$F(b) = \int_a^b f(x)\,dx + F(a). \tag{46}$$

Возможность помещать в память интегратора постоянное числовое значение как раз и рассчитана на то, что в качестве такого будет взято $F(a)$, ведь в этом случае, согласно сказанному, данный блок будет выдавать значение $F(b)$.

[10]Каждый обход схемы сигналами осуществляется через равные промежутки времени, длительность которых можно задавать вручную. По указанной причине первое прохождение схемы и полученные в результате его числовые результаты будут отнесены к моменту времени, равному $t = \tau$, где $\tau$ — упомянутая временная задержка. Следующий обход — к $t = 2\tau$ и т.д. Моменту времени $t = 0$ будет соответствовать сигнал находящийся в конце схемы при ее запуске.

где $\omega_i$ — скорость, вычисленная в предыдущем цикле. Это значение, поделенное на $T_m$ и затем проинтегрированное, согласно (37), вновь даст значение скорости $\omega_{i+1}$, но уже в другой момент времени — следующий за тем, в течение которого скорость была равна $\omega_i$.

Так, повторяясь много раз, данный процесс сформирует два массива данных, в первом из которых будут находиться значения скорости — значения сигналов, пришедших на вход блока с надписью «To workspace» со стороны схемы, а во втором — соответствующие моменты времени. Построенный по их данным график в случае, если схема была построена правильно, совпадет с графиком функции (34) — см. рис. 4.

В заключение сделаем два замечания.

Во-первых, значения переменных «U», «ke» и «Tm», которые прописываются в блоках, нужно задать заранее либо в скрипте, запустив его перед началом моделирования, либо в командной строке.

Во-вторых, если добавить в конце схемы процесса еще один интегратор (рис. 7), то на его выходе будут формироваться значения угловой координаты поворота якоря ($\theta$).

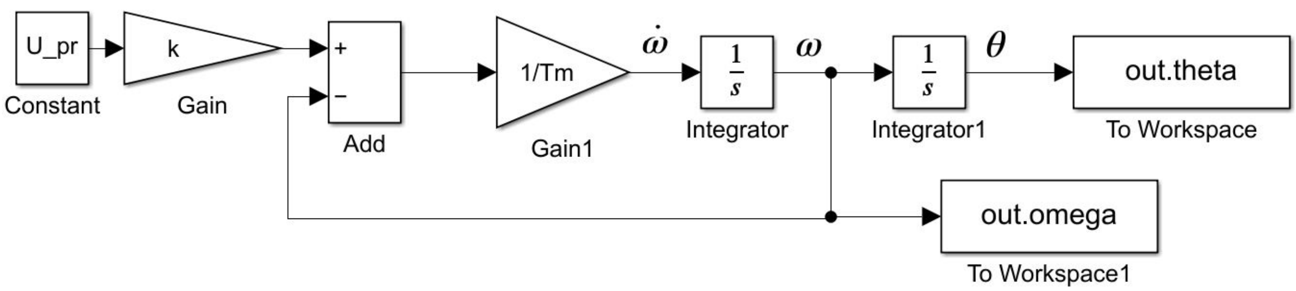


Рис. 7. Схема процесса, возвращающая значения зависимости $\theta(t)$.

## 3 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 $T_m$.

## 4 Порядок выполнения работы

1 Снятие показаний с двигателя EV3

1.1 Соберите конструкцию, показанную на рис. 8.

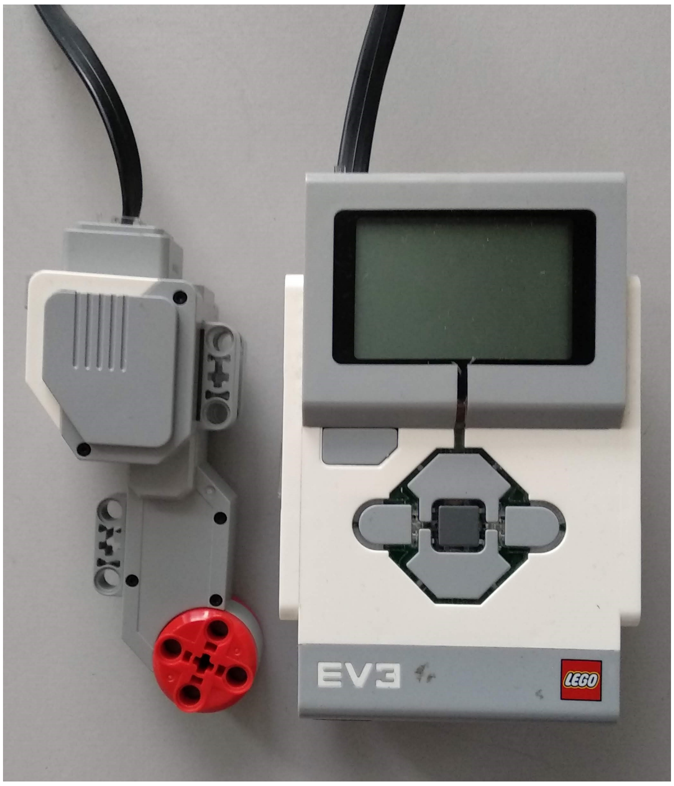


Рис. 8. Общий вид экспериментальной конструкции.

1.2 Напишите программу, которая подает на двигатель EV3 максимально возможное постоянное напряжение (данному требованию удовлетворяет значение аргумента `duty_cycle_sp` метода `run_direct`, равное 100). При этом она также должна периодически снимать показания текущего угла поворота ротора и его угловой скорости и записывать их с соответствующими значениями времени, прошедшего с начала работы программы, в два столбца в текстовый (.txt) файл на EV3.

Пример содержания файла (Первый столбец — значения времени в секундах, второй столбец — значения угла поворота в градусах, третий столбец — значение угловой скорости в градусах в секунду):

| | | |
|---|---|---|
| 0.0202 | 0 | 3 |
| 0.036 | 0 | 7 |
| 0.050 | 1 | 46 |
| 0.063 | 2 | 78 |
| ... | ... | ... |
| 0.963 | 156 | 174 |
| 0.987 | 161 | 174 |
| 1.008 | 165 | 178 |

1.3 Загрузите программу в блок EV3.

1.4 Запустите программу на выполнение.

1.5 Переместите созданный текстовый файл, содержащий экспериментальные данные, из памяти EV3 на компьютер.

1.6 Повторите действия, описанные в пп.1.3-1.5, при 9 различных значениях аргумента `duty_cycle_sp` метода `run_direct`, равных элементам следующего множества $\{80, 60, 40, 20, -20, -40, -60, -80, -100\}$.

2 Обработка экспериментальных данных.
Обработку снятых с двигателя показаний следует провести в среде Matlab.

2.1 Запустите Matlab.

2.2 В открывшемся окне (рис. 9) выберите New Script для запуска редактора кода (рис. 10). Все дальнейшие действия выполняйте в данном редакторе.

2.3 Для чтения данных из файла необходимо использовать функцию `data = readmatrix("filename");`, где filename — полное имя файла, содержащего показания, снятые с двигателя во время выполнения пункта 15, и абсолютный путь до него, например C:\Documents\exp_data.txt. Функция readmatrix создает массив, считывая столбцы данных из файла.

Результат, возвращаемый этой функцией, присваивается переменной data (переменные в Matlab отдельно объявлять не надо).

2.4 Скопируйте первый столбец матрицы data в отдельную одностолбцовую матрицу angle, используя команду `angle=data(:,1)`.

Обращаясь в Matlab к отдельным элементам матриц, после имени последних в круглых скобках записывают два числа — номер строки и столбца соответственно, на пересечении которых находится интересующий элемент. Данный случай не исключение. Указанное на месте номера строки двоеточие означает, что обращение

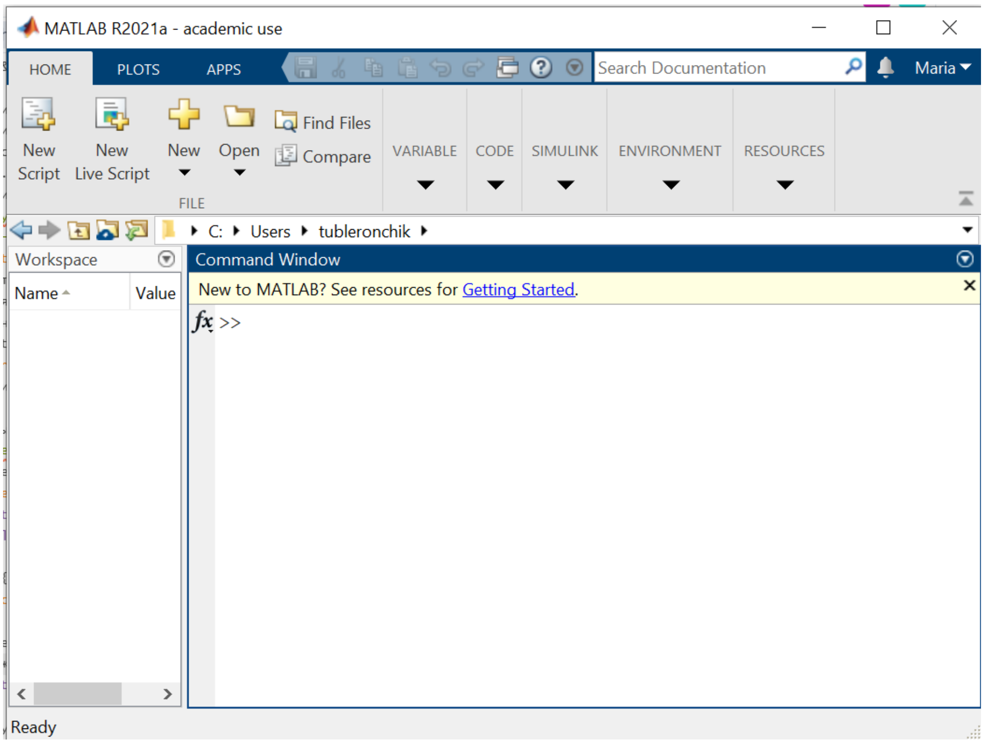


Рис. 9. Главное окно Matlab.

идет ко всем строкам сразу, поэтому в матрицу angle копируется сразу весь первый столбец.

2.5 Теперь переведите все значения матрицы angle из градусов в радианы: `angle=angle*pi/180`. Несложно догадаться, что конструкцией `pi` в Matlab обозначается число $\pi$.

2.6 Второй столбец матрицы data также скопируйте в отдельную одностолбцовую матрицу time и переведите из миллисекунд в секунды: `time=data(:,2)/1000`. Одним действием можно было обойтись и в случае матрицы angle.

2.7 Третий столбец матрицы data также скопируйте в отдельную одностолбцовую матрицу omega и сразу же переведите из градусов в радианы: `omega=data(:,3)*pi/180`.

2.8 Далее необходимо построить графики зависимости угла и угловой скорости от времени. Для построения графика необходимо создать окно графика командой `figure(1)`, где 1 — номер окна. Рисование графика происходит командой plot(x, y). Для зависимости угла поворота ротора от времени данная команда будет выглядеть так: `plot(time, angle)`. Подпись осей X и У происходит с помощью команд `xlabel("")` и `ylabel("")` соответственно, где в скобках указываются желаемые подписи осей. Для построения зависимости угловой скорости от времени необходимо создать второе графическое окно командой `figure(2)`, после чего нарисовать графики с помощью команды plot.

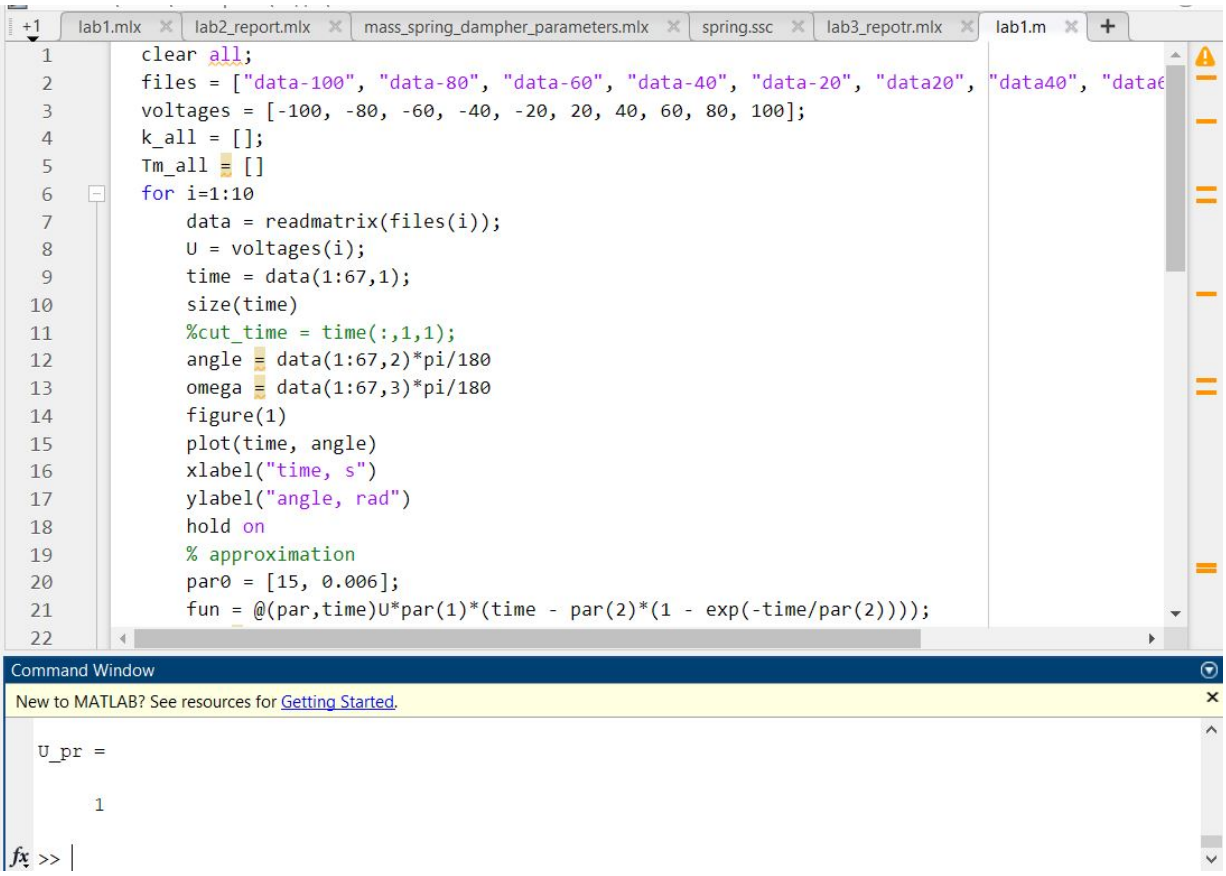


Рис. 10. Редактор кода Matlab.

Если теперь запустить написанный код на выполнение, кликнув, на изображение треугольника Run на панели инструментов, то в появившемся графическом окне отобразятся графики, похожие на представленные на рис. 11, 12.

2.9 Далее нужно выполнить *аппроксимацию* полученной зависимости — подобрать некоторую функцию и найти такое ее уравнение, которое наилучшим образом усреднит полученный результат. В качестве последней следует взять непосредственно функцию $\theta(t)$ — выражение (44). Чтобы найти *методом наименьших квадратов* (прочтите о нем в дополнительной литературе) те значения параметров $k$ и $T_m$, которые обеспечат графику функции (44) наибольшее сходство с графиком, построенным на основании экспериментальных данных, в Matlab нужно сделать следующее.

2.9.1 Задайте одностолбцовую (нельзя однострочную) матрицу par0 из двух элементов, где надо разместить те значения параметров $k$ и $T_m$, которые, по вашему мнению, получатся. Например `par0=[0.1;0.06];`. Также задайте переменную $U$, в которую записывается поданное в процентах напряжение, например, `U_pr = 100;`.

2.9.2 Задайте вид функции, коэффициенты которой будут определяться при аппроксимации. Для уравнения зависимости $\theta(t)$ эта операция в Matalb запишется в виде

```
fun = @(par,time)U_pr*par(1)*(time - par(2)*(1 - exp(-time/par(2))));
```

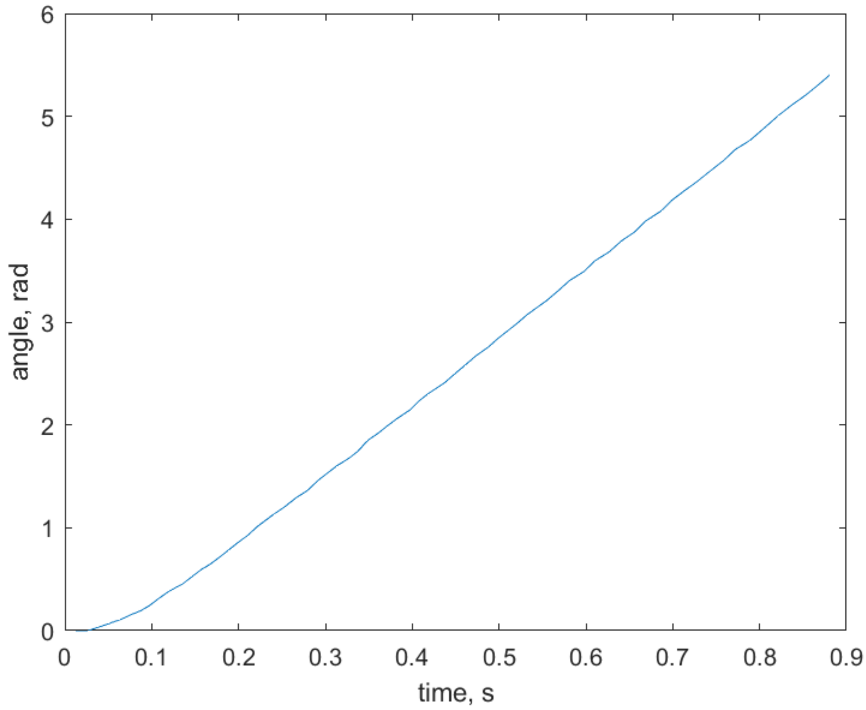


Рис. 11. Первый график угла от времени.

В качестве параметров $k$ и $T_m$ функции $\theta(t)$ берутся значения одностолбцовой (или однострококовой) матрицы par, состоящей из двух элементов.

Смысл именно такого определения интересующей функции вытекает исключительно из алгоритмов, которые использует Matlab для выполнения этой операции, но о них в данном пособии говориться не будет.

2.9.3 Запустите процесс аппроксимации командой `par = lsqcurvefit(fun,par0,time,angle)`. В результате ее работы будет создана матрица par, в которую будут сохранены найденные параметры аппроксимирующей функции.

Процесс аппроксимации может занять некоторое время.

2.9.4 Введите две новые переменные k и Tm, в которые следует сохранить найденные значения параметров $k$ и $T_m$ аппроксимирующей функции и вычислите их средние значения:

```
k = par(1);
Tm = par(2);
```

2.9.5 Для вычисления значения угла поворота ротора $\theta(t)$ задайте новую матрицу time с меньшим шагом `time_apr = 0:0.01:1;` и найдите значение угла $\theta(t)$: `theta = U_pr*k*(time_apr - Tm*(1 - exp(-time_apr/Tm)));`.

2.10 Постройте полученый аппроксимирующий полином, введя команду `plot2d(time_apr,theta)`. Для того, чтобы добавить полученный график на график figure1, необходимо в пункте 2.8 добавить команду `hold on`, которая сохраняет текущий график при добавлении новых.

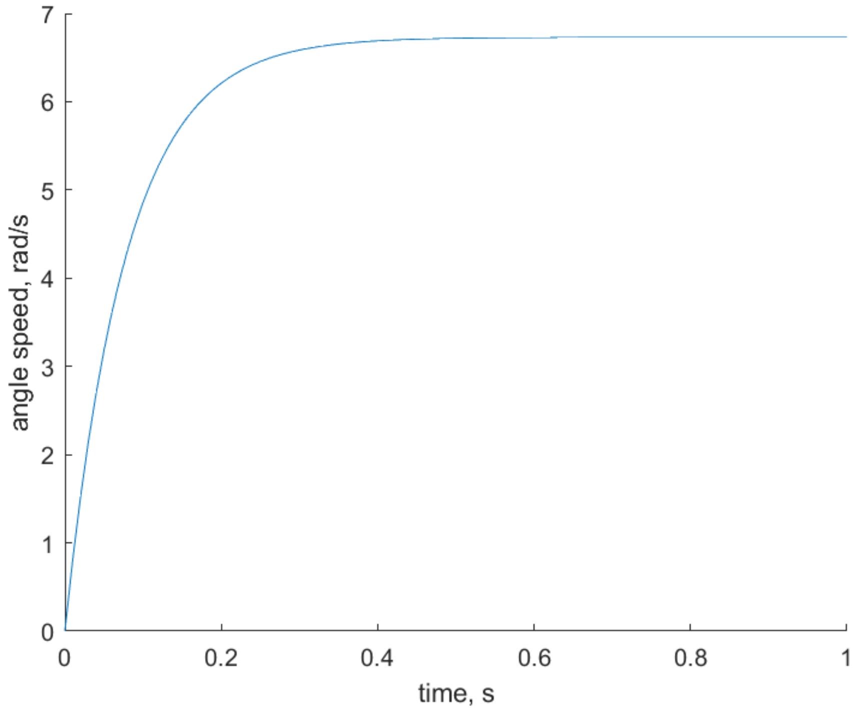


Рис. 12. Первый график угловой скорости от времени.

2.11 Мы выполнили аппроксимацию для зависимости угла поворота ротора. Затем необходимо сделать аппроксимацию для зависимости угловой скорости ротора от времени. Необходимо повторить пункты 2.9.2 - 2.9.5 для функции $\omega(t)$. Теперь весь написанный код следует запустить на выполнение. В результате всех проделанных действий в графическом окне появится еще одна кривая, и общий его вид станет похожим на представленный на рис. 13, 14 пример.

3 Моделирование схемы в Simulink.

Удостовериться в том, что моделирование представленной на рис. 7 схемы исследуемого процесса дает те же результаты, что и решение дифференциальных уравнений, можно, сравнив график аппроксимирующей функции, найденной в п.2, с графиком, построенным на основании результатов моделирования схемы. Для создания последней следует использовать пакет прикладных математических программ для моделирования динамических систем Simulink (предустановлен в Matlab). При моделировании используются средние значения параметров $k$ и $T_m$, вычисленные во время аппроксимации по углу поворота для разных значений мощности двигателя.

3.1 Вызовите рабочие окна Simulink, введя в командную строку Matlab команду `simulink`. Создайте новую модель, открыв Blank Model. (рис. 15). Для открытия библиотеки блоков откройте раздел Library Browse.

3.2 Соберите ту схему двигателя, которая снабжена дополнительным интегратором и, следовательно, выдает значения угла поворота ротора (рис. 7). Значения блоков (надписи на них) выставляются в окне «Gain», появляющемся при двойном клике на

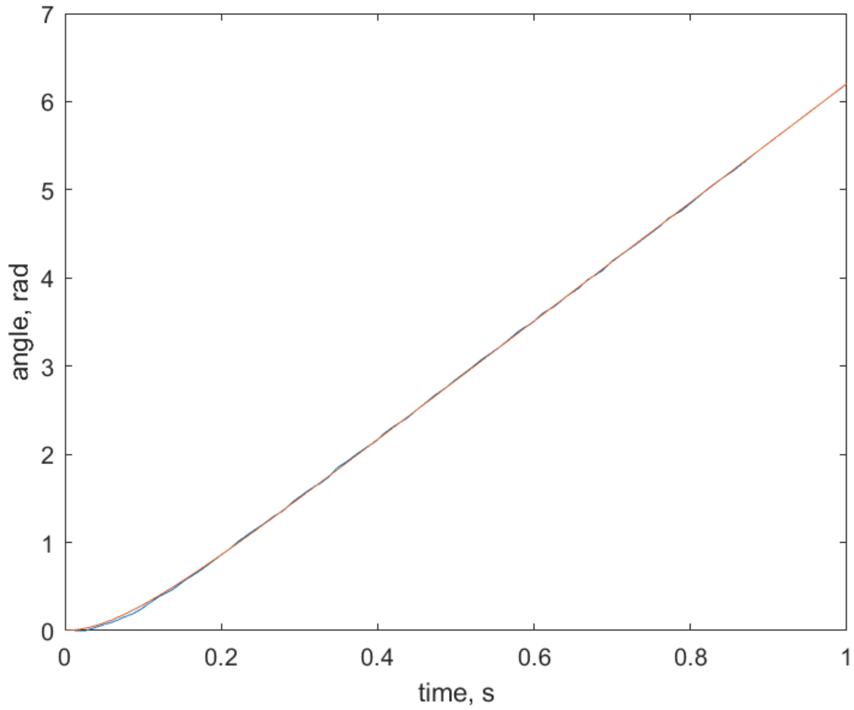


Рис. 13. График зависимости угла от времени с аппроксимирующей кривой.

блок. Обратите внимание, что если вы хотите использовать в качестве значения блока некоторую переменную, предварительно ее нужно определить, введя в командном окне Matlab конструкцию типа «имя_переменной = ее_значение» (например, `J = 0.0023`).

Значение момента инерции ротора примите равным 0.0023 (оно будет получено на следующем занятии).

3.3 В окне «Parameters» блока с надписью «To workplace» есть поле «Variable name», есть общая часть имени массивов результатов моделирования — массива значений скорости (out.имя.Data) и массива соответствующих моментов времени (out.имя.Time). Все остальные параметры оставьте без изменений.

3.4 Время моделирования задается в окне Stop Time в верхней части окна. Присвойте ему значение, приблизительно равное наибольшему значению времени из графиков, построенных в прошлом пункте («Обработка данных»). Например, на основании графиков, показанных на рис. 13 и 14, это значение можно принять равным 1 с.

3.5 Осуществите моделирование системы, выбрав команду «Run» в верхней части окна.

3.6 Постройте получившийся график, дав в главном окне Matlab команду `plot2d(out.имя.Time, out.имя.Data)` (смысл «имени» см. пункт 33).

Пример конечного результата см. рис. 16. Имеющаяся на нем легенда строится командами `xlabel()` `ylabel()`, в качестве аргументов которых следует указать названия графиков в порядке их построения.

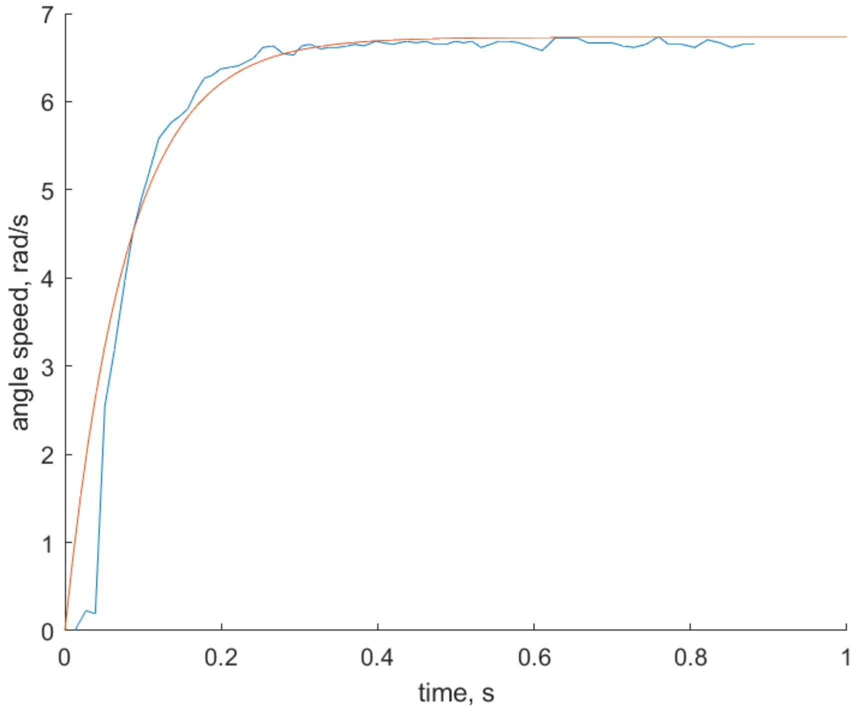


Рис. 14. График зависимости угловой скорости от времени с аппроксимирующей кривой.

4 Повторите все описанные действия для оставшихся девяти файлов с данными, полученными в п. 16 – получите еще 9 пар значений для величин $k$ и $T_m$ и 9 графиков, подобных показанному на рис. 16. Для каждого значения $k$ рассчитайте значение $\omega_{ycm}$ как `w_nls = U_pr*k;`

5 Постройте графики для полученных зависимостей $\omega_{ycm}$(voltage) и $T_m$(voltage), подобные показанным на рис. 17 и 18.

Любые отступления от приведенного порядка действий в рамках тех же программ, дающие необходимый конечный результат, разрешены.


Рис. 15. Создание новой модели в Simulink

# 5 Содержание отчета

В отчете обязательно должны присутствовать:

1 Графики и результаты расчетов величин $k$ и $T_m$, предусмотренные пунктами 2 и 3 раздела «Порядок выполнения работы».

2 Исходный код основной расчетной программы.

3 Построенная в Simulink схема моделирования работы двигателя EV3.

4 Исходный код написанной для EV3 программы.

5 Выводы о проделанной работе.

# 6 Приложение Описание метода работы с Lego EV3

## 6.1 Синхронизация блока EV3 и ПК по USB

### 6.1.1 Windows

1 После соединения компьютера и робота USB кабелем, пройти в Устройства и принтеры в настройках компьютера. EV3 должен быть указан как Remote NDIS Compatible Device.

2 Далее, зайти в «Центр управления сетями и общим доступом», перейти во вкладку «Изменение параметров адаптера». В списке подключений можно найти EV3, подписаный как NDIS Compatible Device, и ваше текущее подключение к интернету ( на рис. 19 текущее подключение называется «Беспроводная сеть»).

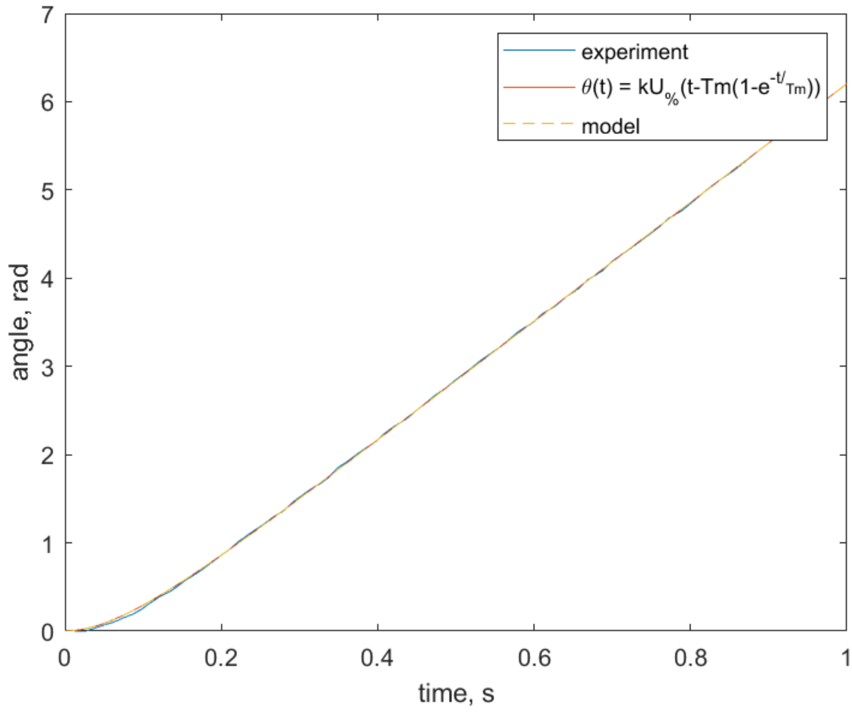


Рис. 16. Итог всех построений.

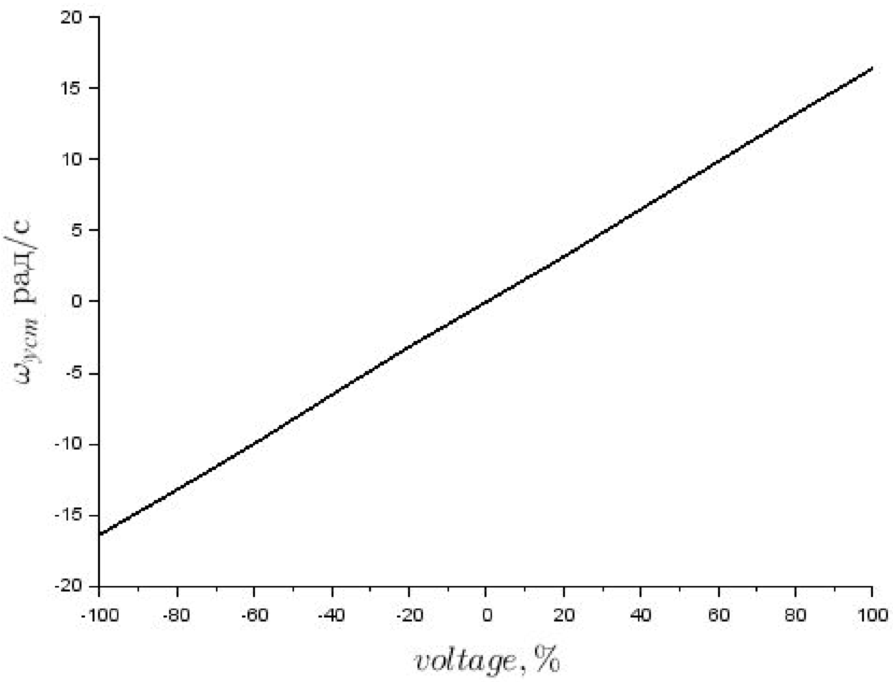


Рис. 17. График зависимости $\omega_{уст}$(voltage).

3 Нажать на текущее подключение правой кнопкой мыши, выбрать вкладку «Свойства» контекстного меню, затем перейти во вкладку «Доступ». Поставить галочку напротив «Разрешить другим пользователям сети использовать подключение к интернету данного компьютера» (рис. 20).

4 После этого уже в роботе необходимо зайти в Wireless and Network, далее в All Network Connections, там должно отобразиться подключение Wired, которое нужно подключить.

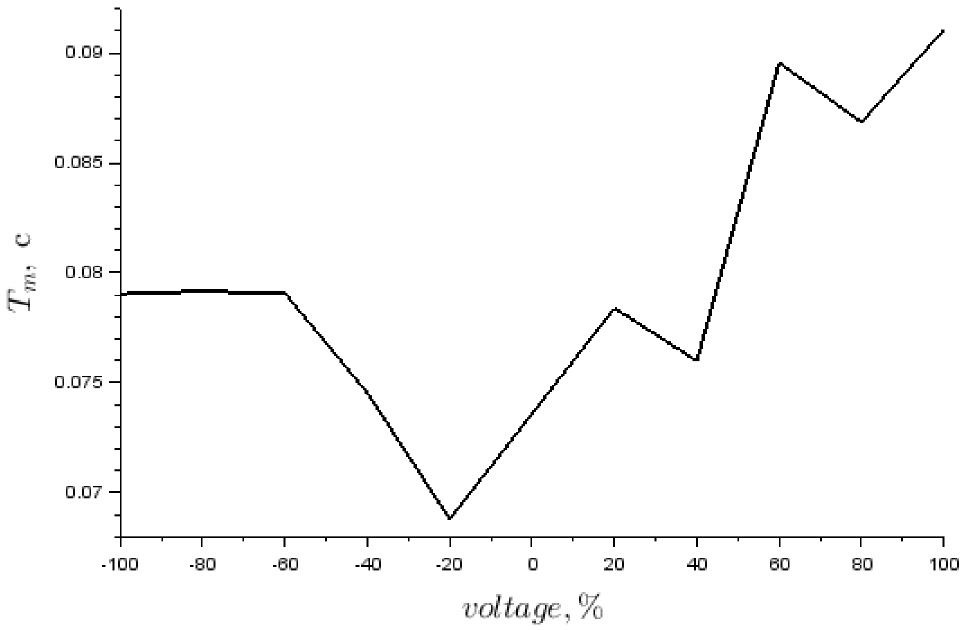


Рис. 18. График зависимости $T_m$(voltage).

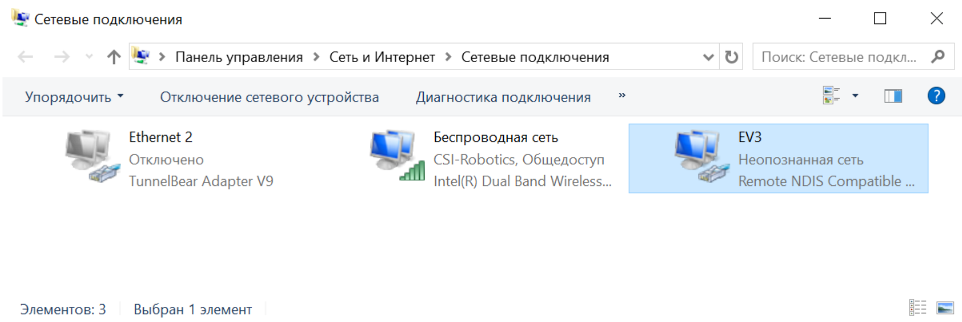


Рис. 19. Окно «Сетевые подключения».

### 6.1.2 Linux

1 После подключения робота по USB к компьютеру нажимаем на символ «Сетевые подключения» и выбираем пункт «Изменить соединения...» (рис. 22).

2 В диалоговом окне «Сетевые соединения» нажать кнопку «добавить», выбрать тип соединения Ethernet.

3 В открывшемся окне «Изменение» заполнить поле «названия соединения» (например, ev3dev). В списке «Устройство» выбрать предоставленный вам вариант и оставить только номер, как показанно на рис. 23.

4 Далее перейти во вкладку «Параметры IPv4» и выбрать способ настройки «Предоставить сеть другим компьютерам» (рис 24).

5 После этого уже в роботе нужно зайти в Wireless and Network, далее в All Network Connections, там должно отобразиться подключение Wired, которое необходимо подключить.

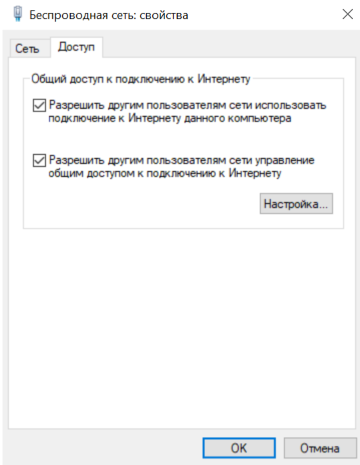


Рис. 20. Окно настройки доступа.

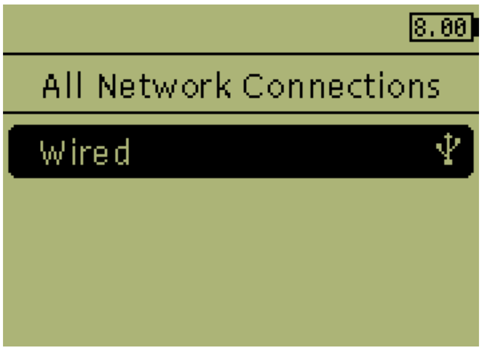


Рис. 21. Окно EV3 «All Network Connections».

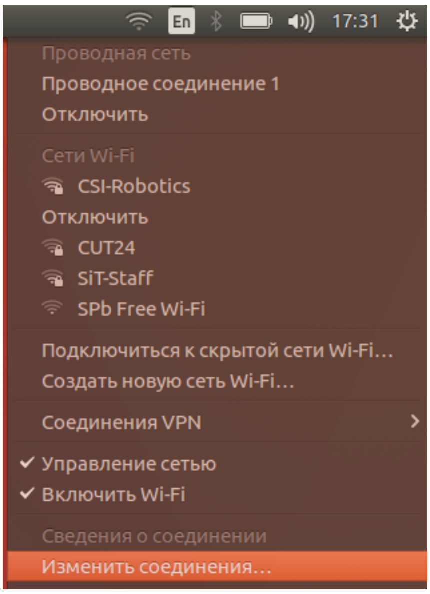


Рис. 22

## 6.2 Синхронизация блока EV3 и ПК по WiFi

Для того чтобы с роботом можно было работать по Wifi, необходим Wifi адаптер, вставляемый в свободное гнездо USB.

Робот и компьютер должны быть подключены к одной и той же сети Wifi. Иных настроек не потребуется.

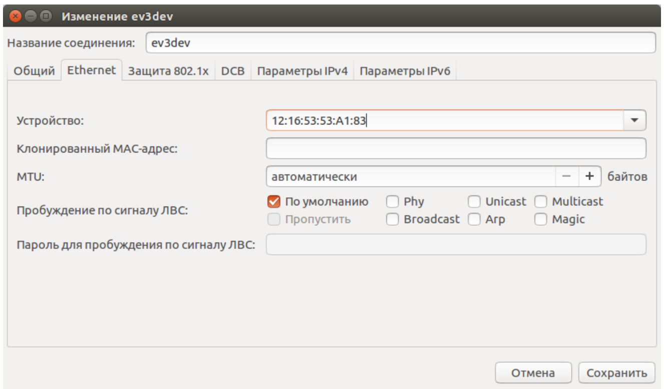


Рис. 23

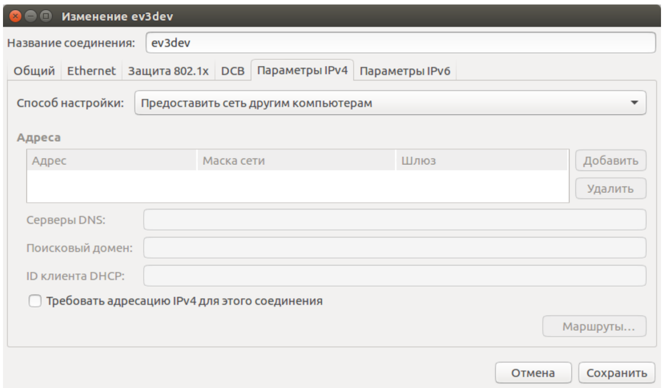


Рис. 24

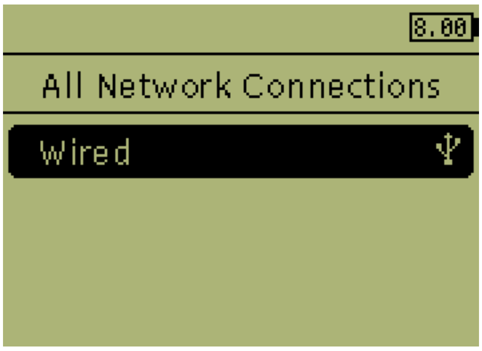


Рис. 25. Окно EV3 «All Network Connections».

## 6.3 Работа с файловой системой EV3

### 6.3.1 Windows

Для работы с EV3 потребуется установить программу WinSCP.

1 Открываем программу WinSCP. В открывшемся диалоговом окне (рис. 26) в качестве имени хоста используем — `ev3dev`, в качестве имени пользователя — `robot`, пароль — `maker`. Подключение можно сохранить.

2 Откроется окно, в котором можно удобно работать с файлами внутри робота. При необходимости можно открыть консоль EV3. Метод работы с консолью робота описан ниже в п 7.

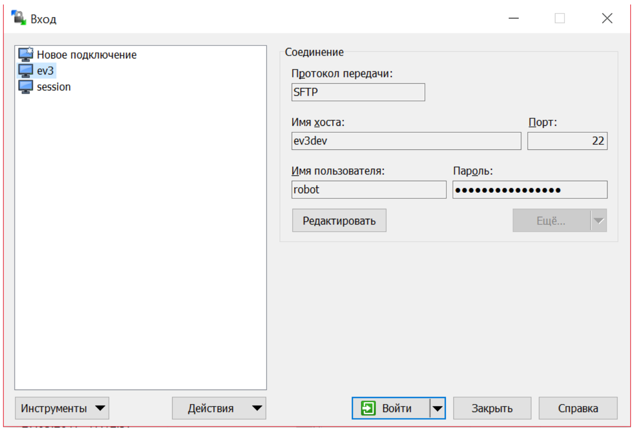


Рис. 26. Окно подключения WinSCP.

### 6.3.2 Linux

1 Для начала необходимо установить на компьютер sshfs, используя команду:

`sudo apt-get install sshfs`

2 Также необходимо создать папку, куда будет монтироваться робот:

`mkdir (путь до создаваемой директории)`

3 После чего можно начинать монтирование:

`sshfs robot@(robot IP):/home/robot/ /(созданная директория)`

4 Потребуется ввести пароль - «maker», IP робота можно посмотреть в углу на экране EV3. В примонтированной папке хранятся все файлы, находящиеся в памяти робота.

5 Далее подключается SSH:

`ssh robot@(robot ip)`

6 Потребуется еще раз ввести пароль. После чего откроется терминал EV3 рис. 27. Через

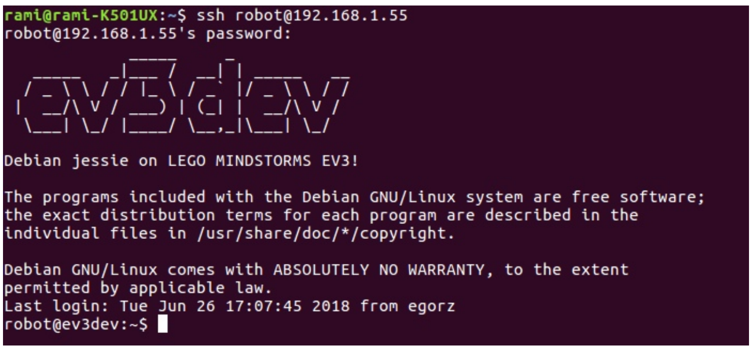


Рис. 27. Терминрал EV3

этот терминал осуществляется работа с программами.

7 Для того чтобы дать роботу права на выполнение программы:

`chmod +x file.py`

8 Чтобы запустить программу:

```
./file.py
```

9 Важно! Перед завершением работы и выключением робота необходимо выключить SSH и отмонтировать папку, подключённую по SSHFS.

10 Для отключения SSH достаточно ввести в терминале команду:

```
exit
```

11 Разрыв соединения между компьютером и роботом необходимо производить непосредственно из примонтированной папки. Для этого заходим в эту папку и, после нажатия правой кнопки мыши, выбираем пункт `Открыть в терминале`. После чего вводим:

```
sudo umount (имя папки)
```

Потребуется ввести пароль, после чего папка отмонтируется. EV3 можно выключать.

## 6.4 Введение в язык python

Для прохождения курса необходимо знать основы языка python3. Ниже будут приведены ресурсы для его изучения. Книги подойдут для людей, которые хотят изучать данный язык углубленно. Для этого курса достаточно просмотреть первые 20 уроков https://pythonworld.ru/samouchitel-python или онлайн уроки на английском https://pythonprogramming.net/python-fundamental-tutorials/. Также, можно пройти онлайн-курс на платформе степик https://stepik.org/course/512/.

- Книги
  - Марк Лутц, Изучаем Python
  - Зед Шоу, Learn Python the Hard Way
  - Эрик Мэттс, Python Crash Course

## 6.5 Используемые функции

При написании программы для EV3 потребуется использовать некоторые классы и функции библиотеки EV3 и встроенные элементы языка Python.

Для импорта библиотеки EV3 необходимо прописать в начале программы следующую строку

- `from ev3dev.ev3 import *`
  Символ * в данном случае означает, что мы импортируем все элементы, находящиеся в библиотеке EV3.
- `import time`
  Подключает модуль time, предназначенный для работы со временем в Python.

- `time.sleep(x)`

  Даёт возможность приостановить выполнение программы на определенное количество секунд. Количество времени указывается вместо x.

- `time.time()`

  Отображает Unix-время в секундах. Прим. Unix-время - время, моментом начала отсчёта которого считается полночь (по UTC) с 31 декабря 1969 года на 1 января 1970.

- `mA=LargeMotor('port')`

  Данный класс предназначен для работы с большим мотором Lego EV3.

  Выходные порты указываются соответствующими заглавными буквами с префиксной конструкцией out. Например, команда mA = LargeMotor('outA') позволит работать с двигателем, включенным в порт A.

- `f = open('text.txt', 'w')`

  Функция open открывает файл, в который будет производиться запись данных, где 'text.txt' - это название файла, а 'w' это режим, при котором функция позволит создать новый файл и записывать в него данные.

- `f.write('0'+' '+'0'+'\n')`

  Запись в файл происходит с помощью метода write. Например, f.write('Hello, world!') запишет в файл строку Hello, world!

- `mA.run_direct(duty_cycle_sp=-100)`

  Метод run direct запускает мотор с процентом напряжения, указанным в duty_cycle_sp. Знак отвечает за направление движения. Например, mA.run_direct(duty_cycle_sp=100) запустит двигатель на полную мощность в направлении "вперед".

- `mA.position`

  Позволяет узнать угол поворота двигателя в градусах.

- `mA.speed`

  Позволяет узнать угловую скорость двигателя в градусах в секунду.

Пример программы на python3, которая вращает мотор на протяжении 5 секунд, записывает угол поворота вала и время в файл `data`. В данном участке кода используется конструкция `try-finally`, функции которой будут объяснены позднее. Во время выполнения работы **вы можете написать свою программу**, в том числе более простую – использование конструкции `try-except-finally` и цикла `for` совершенно необязательно.

```
1 #!/usr/bin/env python3
2 from ev3dev.ev3 import LargeMotor
3 import time
4 
5 motorA = LargeMotor('outA')
6 voltages = [100, 80, 60, 40, 20, -20 , -40, -60, -80, -100]
7 try:
8     for vol in voltages:
9         timeStart = time.time()
10        startPos = motorA.position
11        name = "data" + str(vol)
12        file = open(name, "w")
13        while True:
14            timeNow = time.time() - timeStart
15            motorA.run_direct(duty_cycle_sp = vol)
16            pos = motorA.position - startPos
17            file.write(str(timeNow) + " " + str(pos) + " " + str(motorA.speed) + "\n"
    )
18            if timeNow > 1:
19                motorA.run_direct(duty_cycle_sp = 0)
20                break
21        file.close()
22 except Exception as e:
23     raise e
24 finally:
25     motorA.stop(stop_action = 'brake')
26     file.close()
```

Рис. 28. Пример программы

Строка №1 показывает, какой интерпретатор должен использоваться для запуска данной программы. Данная строка должна присутствовать во всех ваших программах.

В строках №2 и №3 происходит подключение необходимых модулей (так в Python называют библиотеки). Модуль `ev3dev.ev3` содержит классы и функции, которые необходимы для запуска программ на блоках `Lego EV3`, `time` включает в себя различные инструменты для работы со временем.

Строка №5 создает объект, соответствующий мотору EV3, подключенному в порт A.

В строке №6 задаются значения напряжения в процентах, которые мы будем подавать на двигатель. Далее с помощью цикла for программа переберет эти значения (на каждой итерации цикла переменная vol примет одно из значений массива voltages)

В строках №9, 10 задаются начальные значения времени и угла поворота для данного напряжения, они понадобятся, чтобы записывать время и угол поворота начиная с нуля.

В строке №12 создается файл с именем `data{vol}`, где vol — значение напряжения, которое сейчас подается. Например, data100 при подаче 100% напряжения, в который будут записываться данные. На запись в файл указывается флаг `w` (второй аргумент).

Далее в бесконечном цикле вычисляется время с начала подачи определенного напряжения (строка №14), на двигатель подается это напряжение (строка № 15) и вычисляет угол, на который повернулся двигатель с начала подачи данного напряжения (строка №16)

В строке № 17 в файл записываются значения времени, угла поворота и скорости двигателя (символ «\n» обозначает, что следующая запись будет с новой строки)

В строке №18 проверяется, прошло ли больше 1 секунды со смены напряжения.Еесли да, то останавливаем мотор (строка №19) и выходим из цикла while с помощью оператора break и закрываем файл в строке №21. После этого программа переходит на следующую итерацию цикла for, в которой подается следующее напряжение из списка и все действия повторяются.

Конструкция try-except-finally работает по следующему принципу: участок кода после инструкции try (строки с №7 по №21) выполняется обычным образом. В случае завершения программы, аварийного или нормального, участок кода после инструкции finally обязательно выполняется. Использование данного инструмента языка Python позволяет обеспечить обязательное закрытие всех открытых файлов (строка №26) и остановку двигателей после завершения программы (строка №25). В случае аварийного завершения вызывается исключение, которое будет отражено в стандартном потоке вывода ошибок.