**Лекция 2. Способы создания моделей в MATLAB/Simulink**

Рассматриваются различные способы моделирования двигателя постоянного тока независимого возбуждения (ДПТ НВ): MATLAB-Simulink, MATLAB-Script, Си, Simulink/Simscape. В каждом из этих способов есть свои плюсы и минусы. Одинаковыми, как будет показано, будут характеристики, соответствующие одним и тем же параметрам двигателя в системе относительных единиц.

В качестве объекта управления рассмотрим двигатель постоянного тока независимого возбуждения (ДПТ НВ).

**M**

**ω**

**Φ**

**R**

**L**

**I**

**U**

**E**

**+**

**-**

*Рис.1.* Схема двигателя постоянного тока независимого возбуждения

Математическая модель ДПТ НВ двигателя (рис. 1) состоит из электрической и механической частей, описываемых системой уравнений (1):

** (1)

*где*

*U, В – напряжение источника питания, подаваемое на якорную обмотку двигателя;*

*Rя, Ом – полное активное сопротивление якорной цепи;*

*Iя, А – электрический ток в якорной обмотке двигателя;*

*Lя, мГн – индуктивность якорной цепи;*

*E, В – противо-ЭДС двигателя;*

*J, кг·м2 – суммарный момент инерции якоря и нагрузки;*

*M, Mс, Н·м – электромагнитный и статический моменты двигателя соответственно;*

*ω, рад/с – скорость вращения вала двигателя;*

*− поток полюсов (примем его за 1), тогда*

*– коэффициент связи между скоростью и противо-ЭДС;*

* – коэффициент связи между током якоря и электромагнитным моментом.*

*Т*я = *L*я / *R*я – постоянная времени якорной цепи;

*Т*м = (*R*я*J*) / (*kek*м) – электромеханическая постоянная времени двигателя;

Первое уравнение системы (1) − баланс напряжений в электрической части двигателя. Второе − уравнение движения для механической части. Третье и четвертое − уравнения связи электрической и механической частей двигателя.

В системе уравнений (1) произведем замену операторов d/dt операторами s (d/dt = s, где s − формально оператор Лапласа). В этом случае перейдем к алгебраическим уравнениям:

*U=RЯIя+LЯsIЯ +E;*

*M-MС =Jsω;*

*M=kмIЯ;*

*E=keω.*

В первом уравнении слагаемое с токами перенесем в левую часть, а ЭДС и напряжение − в правую:

*RЯIЯ+LЯsIЯ=U-E.*

Определим ток *IЯ*:

*IЯ (RЯ+LЯs) = U-E, откуда*

*IЯ  = (U-E) · = (U-E) · , где Тя = Lя / Rя* . (2)

Аналогично, из второго уравнения

*ω = (M-MС) .* (3)

**2.1. Создание модели в среде Simulink**

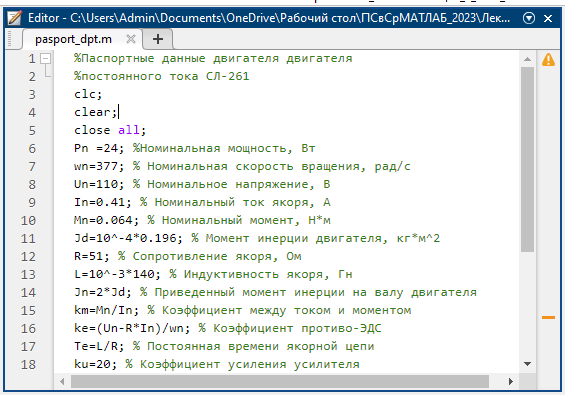
Считая входом значение напряжения U, подаваемое на якорь двигателя, а выходом скорость вращения двигателя ω, используя уравнения (2), (3) и уравнение связи между током и моментом, составим в Simulink структурную схему, показанную на рис. 3. Предварительно следует составить файл скрипта MATLAB с исходными данными двигателя, показанный на рис. 2. Данные конкретного двигателя возьмем из табл. 1.

**Паспортные данные двигателя**

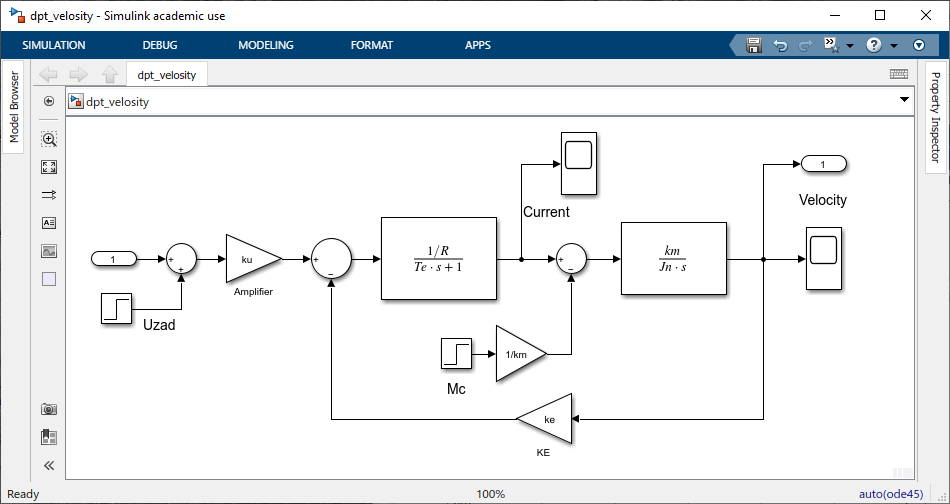
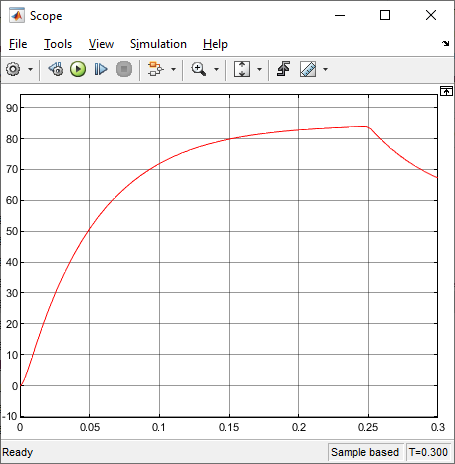
*Таблица 1*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Марка двигателя | *Р*н,  Вт | ɷн,  рад/с | *U*н,  В | *I*н,  А | *М*н,  Н·м | *J*дв*·*10−4,кг·м2 | *R*я,  Ом | *L*я,  мГн |
| СЛ-261 | 24 | 377 | 110 | 0,41 | 0,064 | 0,196 | 51 | 140 |

*Здесь нижние индексы «н» соответствуют номинальным значениям параметров.*



*Рис. 2.* Файл скрипта с паспортными данными двигателя СЛ-261

*Рис. 3.* Модель ДПТ НВ в MATLAB/Simulink и результат моделирования

Выполним симуляцию созданной модели. В качестве сигнала задания выберем такой, чтобы на обмотку якоря двигателя подавалось напряжение U = 1\*Ku В = 1\*20 В. Модель и результат симуляции приведены на рис. 3. Действие момента сопротивления (при Mc=0.02, T=0.25) показано на рис. 3.

Отметим достоинства и недостатки данного метода.

*Достоинства:*

* наглядность, высокая скорость и простота сборки структурной схемы из готовых блоков библиотеки с многочисленными элементами;
* простота подключений различных видов осциллографов к любой точки схемы и вывод на экран множества переменных от времени.

*Недостатки:*

* сложно проследить за преобразованиями, производимыми внутри передаточной функции с оператором *s*;
* при увеличении числа элементов структурной схемы возможны сбои, вызванные либо делением на нуль, либо таким явлением как «алгебраические контуры».

**2.2. Создание математической модели в среде MATLAB**

Другая форма описания ДПТ НВ может быть представлена непосредственно в среде MATLAB в виде его передаточной функции.

Передаточная функция скоростной части ДПТ по управляющему воздействию может быть представлена выражением



где

*ku −* коэффициент усилителя, 

Для получения передаточной функции в файл скрипта с паспортными данными двигателя введем дополнительно следующие команды:

k1=1/R;

k2=km/Jn;

numdv=ku\*1/ke;

dendv=[Te/(k1\*k2\*ke) 1/(k1\*k2\*ke) 1];

Wdpt=tf(numdv,dendv)% Передаточная функция ДПТ по скорости

step(Wdpt); %График переходного процесса ДПТ по скорости

После этого запустим этот файл на выполнение. Результаты представлены на рис. 4.

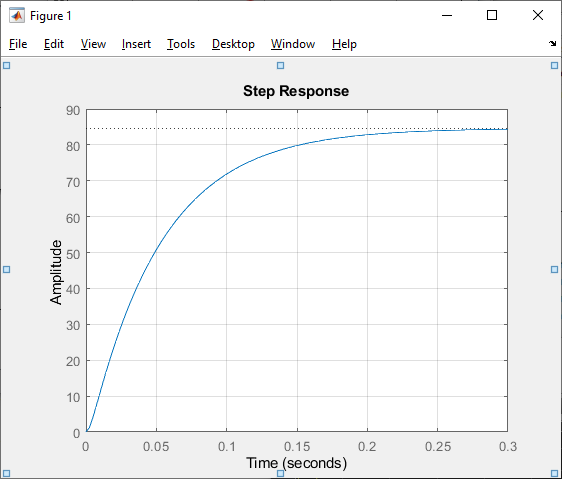
Сравнение полученных результатов (см. рис. 3 и 4) показывает, что они полностью совпадают (без учета *М*с).

*Достоинства:*

* простота получения передаточной функции, временных и частотных характеристик;
* быстрое преобразование к любой форме модели (нуль-полюсной, дискретной, в форме описания в пространстве состояний;

*Недостатки:*

* нет той наглядности, которая присутствует при сборке структурной схемы в Simulink;



84.63

Wdpt = -----------------------------------

0.0001488 s^2 + 0.0542 s + 1

*Рис.4.* Результаты моделирования ДПТ в среде MATLAB

* требуется много времени для прописывания команд по выводу графиков изменения во времени нескольких переменных и тем более во многих точках системы.

**2.3. Запись математической модели на языке программирования Си в MATLAB**

Обратимся снова к системе уравнений (1). Выразим производные *dIЯ/dt* и *dω/dt* через конечные разности:

*dIя/dt = (U-E-IяRя)*

*= (U-E(i)-Iя(i)Rя)*

где i − номер шага, dt − фиксированный шаг интегрирования.

Тогда ток в якоре в момент времени (i+1) определится как:

*Iя(i+1) = Iя(i) + (U-E(i)-Iя(i)Rя)* . (4)

Аналогично, для скорости *ω*:

*dω/dt = (M-Mс) ;*

*= (M-Mс) ;*

*ω(i+1)= ω(i)+(M-Mс) .* (5)

Дополнительно следует воспользоваться соотношениями

*M(i) = km\*Iя (i);* (6)

*E(i) = ke\*ω(i).* (7)

Тогда математическую модель ДПТ, в соответствии с уравнениями (4) − (7), на языке программирования Си для среды MATLAB можно представить в виде функции mexFunction, написанной в соответствии с правилами синтаксиса mex-функций, листинг которой представлен ниже.

Правила синтаксиса (см. справку по запросу **mexfunction (C)**):

**C Syntax**

**#include "mex.h"**

**void mexFunction(int nlhs, mxArray \*plhs[], int nrhs,**

**const mxArray \*prhs[])**

Примечание: Если при создании файла в редакторе вы записали его с расширением **.m**, измените это расширение на **.c** вручную.

Листинг функции «Program\_dpt.c».

#include "mex.h"

#define FILENAME "dpt\_velocity.txt"

void mexFunction(int nlhs, mxArray \*plhs[], int nrhs, const mxArray \*prhs[])

{

//Параметры двигателя постоянного тока

const double R=51, // Сопротивление якоря, Ом

L=0.140, // Индуктивность якоря, Гн

Te=2.7451e-3, // Постоянная времени якорной цепи L/R

J=0.392e-4, // Приведенный момент инерции на

// валу двигателя J=2\*Jd

km=0.156, // Коэффициент между током и моментом

ke=0.236, // Коэффициент противо-ЭДС

ku=20; // Коэффициент усиления усилителя

// Переменные математической модели двигателя постоянного тока

double U = 0, // управляющее воздействие

E = 0, // эдс двигателя

M = 0, // электромагнитный момент двигателя

// Mc = 0, // момент статического сопротивления

// в данном случае полагаем равным нулю

Ia = 0, // ток якоря

w = 0; // скорость вращения вала ДПТ

double dt = 1e-3; // шаг интегрирования

double t = 0; // текущее значение времени

double t1 = 0.3; // конечное значение времени расчета

unsigned int cnt = t1/dt; // количество точек

// Создаем временный файл, в который будем записывать

// текущие значения

FILE \*fp = fopen(FILENAME, "w");

// цикл расчета

for (unsigned int i = 0; i < cnt; i++) {

// подача управляющего воздействия

if (t >= 0.0)

U = ku ;

else

U = 0.0;

// расчет мат модели двигателя пост тока

Ia = Ia + dt \* (U - E - R \* Ia)/(R\*Te);

M = Ia \* km;

E = w \* ke;

w = w + dt \* M/J;

// записываем точки во временный файл

fprintf(fp,"%f\t%f\t%f\n", t, Ia, w);

// увеличиваем переменную время

t = t + dt;

}

// Закрываем текстовый файл с текущими значениями

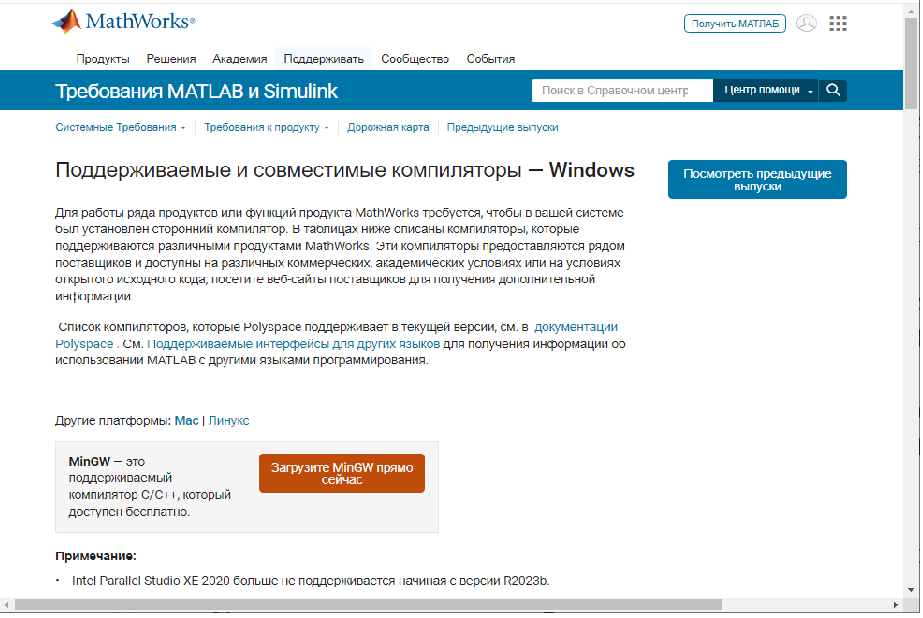
fclose(fp);

}

Чтобы выполнить этот файл, сначала необходимо произвести настройку компилятора (если он есть и ранее не был проинициализирован) с помощью команды

>> mex –setup

Если компилятора нет, то его нужно установить, как это было указано в материалах предыдущей лекции. Файл компилятора имеется в лаборатории 7310. Его также можно скачать с сайта mathworks.com (см. рис., но для этого нужно пройти регистрацию на сайте).



Команда mex –setup выводит следующее сообщение:

MEX configured to use 'MinGW64 Compiler (C)' for C language compilation.

To choose a different language, select one from the following:

mex -setup C++

mex -setup FORTRAN

В ответ следует выбрать первый пункт сообщения: mex -setup C++. Если все в порядке, будет выведено:

MEX configured to use 'MinGW64 Compiler (C++)' for C++ language compilation.

После этого следует запустить файл на компиляцию командой:

>> mex Program\_dpt.c

Building with 'MinGW64 Compiler (C)'.

MEX completed successfully.

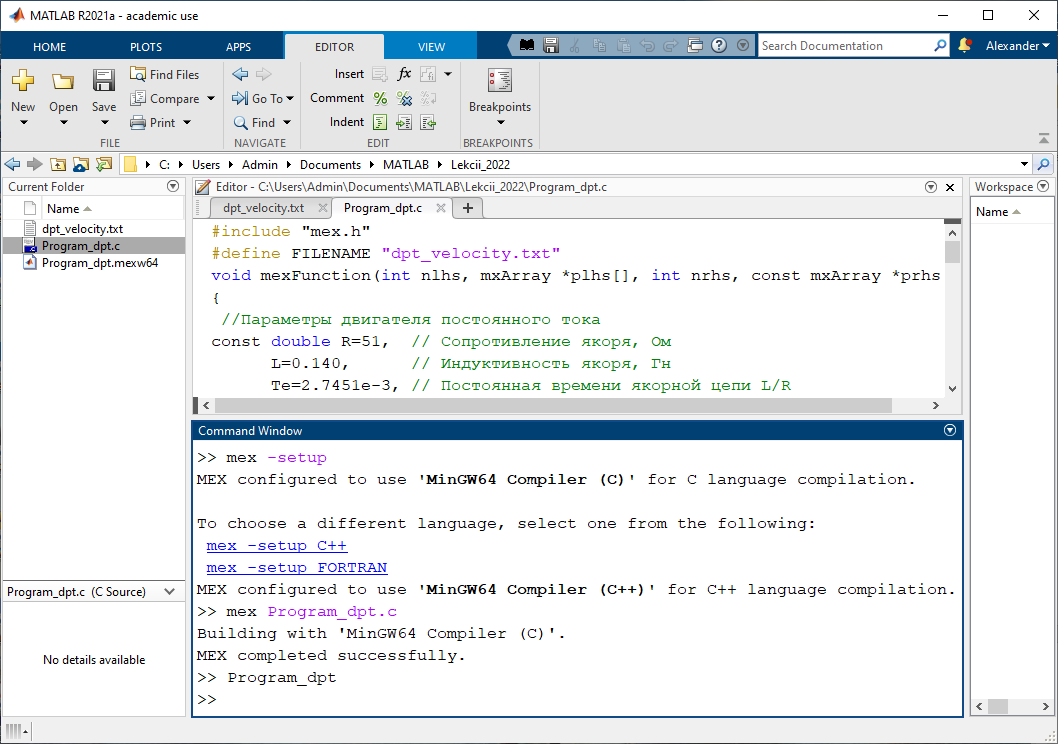
*Замечание: если вами уже был предварительно проинициализирован компилятор, то повторной инициализации не требуется. Она автоматически будет выполнена по расширению файла .c при запуске файла на компиляцию.*

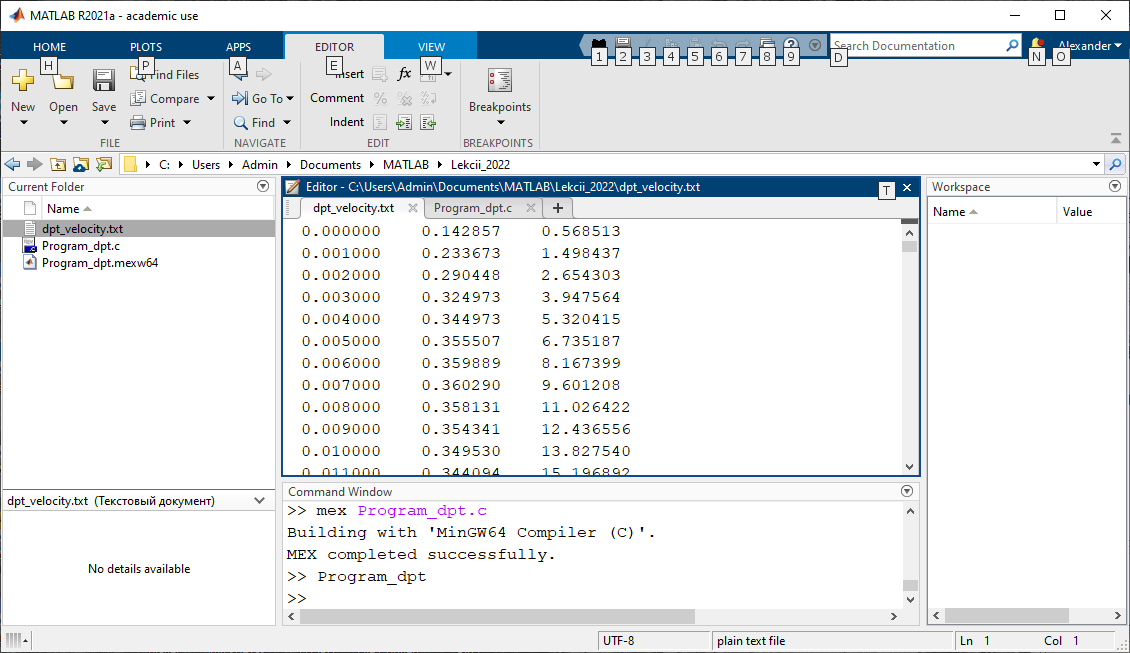
При удачной компиляции MATLAB выводит соответствующее сообщение (MEX completed successfully). При этом будет создан исполняемый файл Program\_dpt.mexw64.

После компиляции можно этот файл загрузить на исполнение командой:

>> Program\_dpt

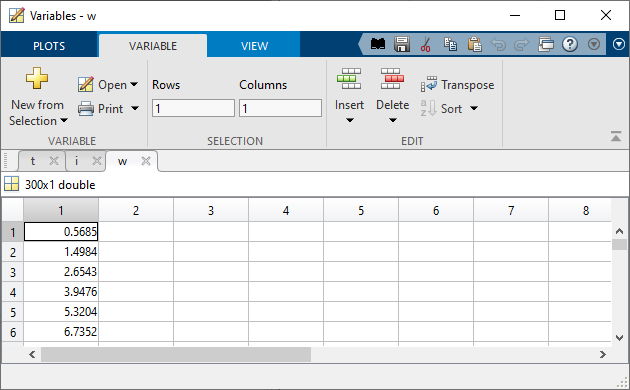
По завершении программы будет создан текстовый файл dpt\_velocity.txt с результатами. В этом файле содержаться три столбца со значениями времени − t, тока − i и скорости − w вала двигателя соответственно. Описанные действия иллюстрируются рис. 5.





*Рис. 5.* Запуск и результаты работы программы на языке Си

Создадим в рабочей области MATLAB три переменные: *t, i* и *w*. Из полученной ранее таблицы скопируем соответствующие значения в рабочую область MATLAB (см. рис 6). Копировать таблицу можно целиком, а затем для соответствующей переменной удалить лишние столбцы.

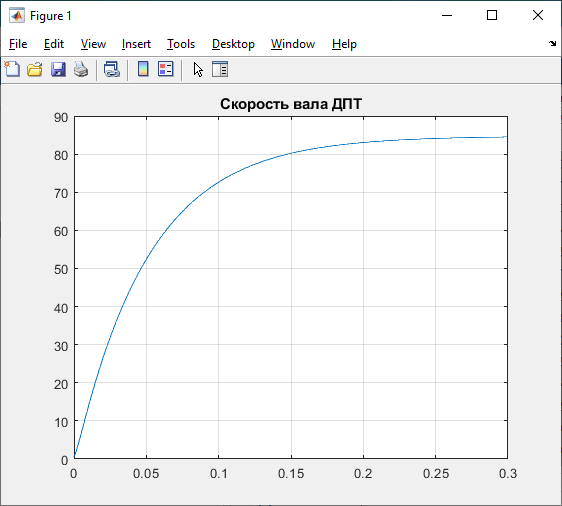


*Рис. 6.* Рабочая область с переменными *t, i* и *w*

Построим график изменения скорости командой

>> plot(t,w); grid;

представленный на рис. 7.



*Рис. 7.* График скорости вала двигателя,

полученный программой Program\_dpt.c

Видно, что результат и в этом случае совпадает с результатами предыдущих моделей.

*Преимущества:*

− программу, написанную на языке Си, можно загрузить в микроконтроллер, но для этого необходимо прописать адреса цифрового ввода.

*Недостатки*:

− более высокая трудоемкость;

− с целью увеличения быстродействия необходимо в программе прописать только необходимые файлы из конкретных библиотек.

*Замечание:* Подобным способом можно создать и математическое описание ДПТ в виде файла-скрипта на языке MATLAB. Результат будет тот же, но скорость расчета ниже.

**2.4. Создание Simulink-модели в виде модели в пространстве состояний**

Система уравнений (1) может быть записана в следующем виде:



 (8)





Здесь:

Входные переменные:

 ,

Выходные переменные:

 ,

Переменные состояния:

 − ток и скорость, соответственно.

В векторно-матричной форме стандартная система уравнений записывается в виде:

(9)

(10)

где в соответствии с (8)

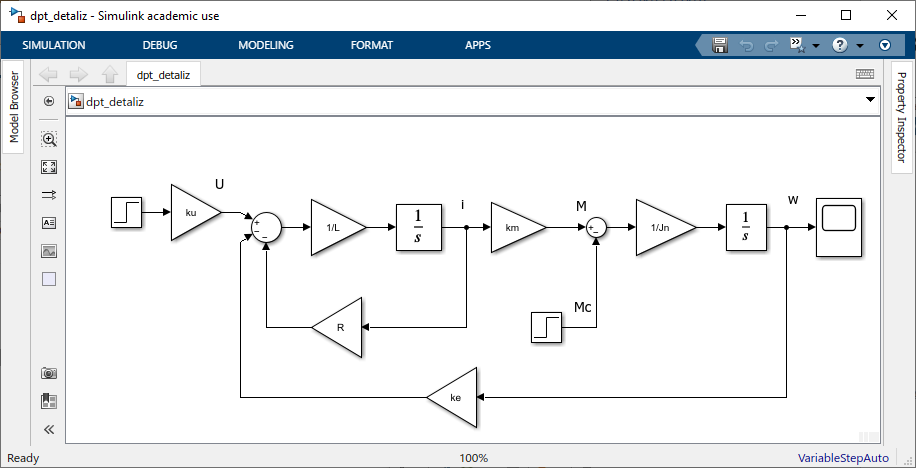
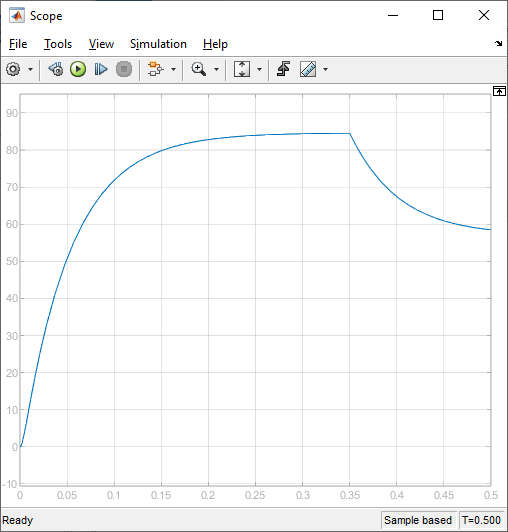
|  |  |
| --- | --- |
| , , C= | (11) |

Так как в данном случае входные переменные не участвуют в формировании выходных переменных, матрицу обхода D полагаем равной нулевой матрице (D = zeros(2,2)), поэтому в (11) она отсутствует.

В данном примере показано, что описание системы в матричной форме легко получить на основании дифференциальных и алгебраических уравнений. При этом дифференциальные уравнения высокого порядка с помощью замены переменных необходимо записать в стандартной форме Коши (в виде системы дифференциальных уравнений, разрешенных относительно первых производных переменных состояния).

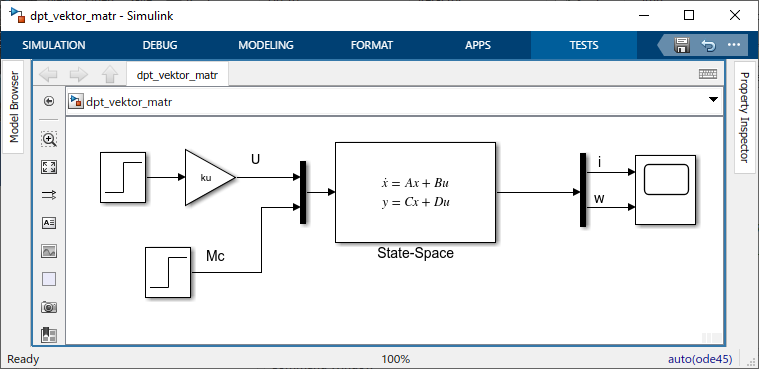
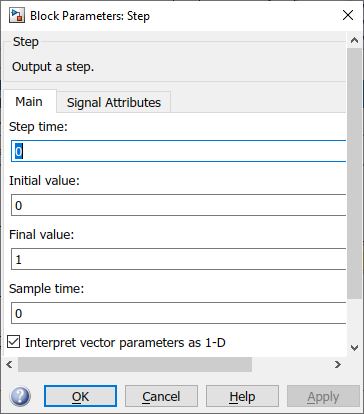
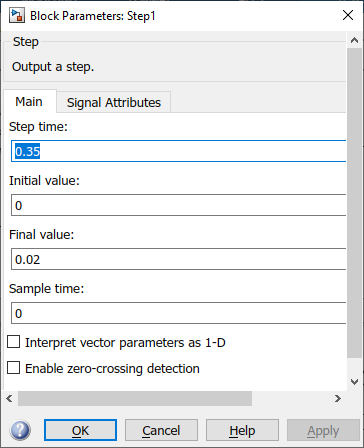
Детализированная структурная схема электромеханической системы с двигателем постоянного тока, соответствующая матричным уравнениям (9) − (11) приведена на рис 8. Структурная схема называется детализированной, если она содержит только звенья интеграторов и коэффициентов.

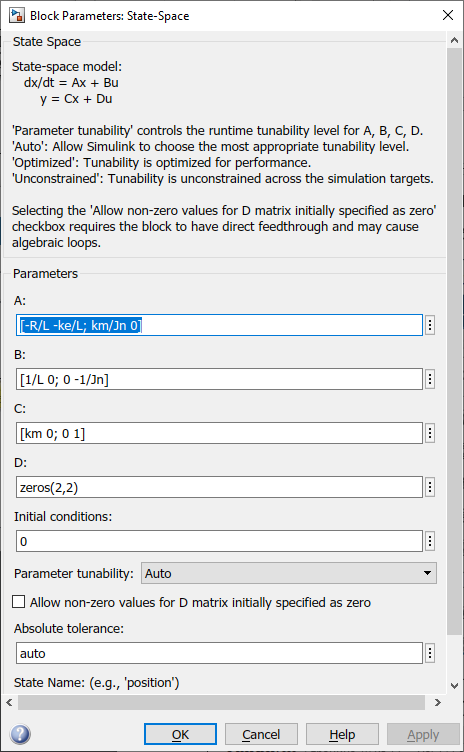
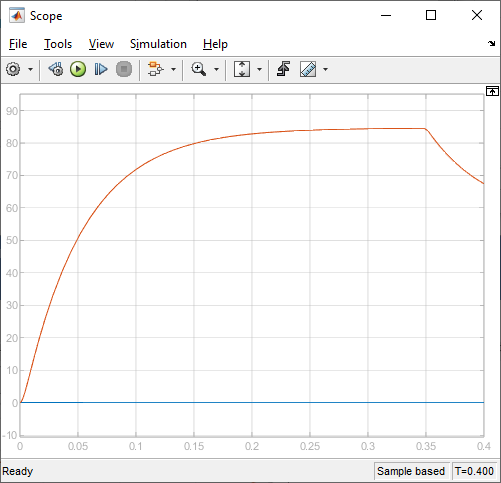
Как видно из графика переходного процесса по скорости вращения вала двигателя, он в точности совпадает с результатами, полученными ранее.

*Рис. 8.* Детализированная структурная схема ДПТ и результат моделирования

Можно также использовать векторно-матричную форму описания модели объекта управления с помощью блока State-Space. Пример такой модели, окон настроек ее блоков и результат симуляции представлены на рис. 9.

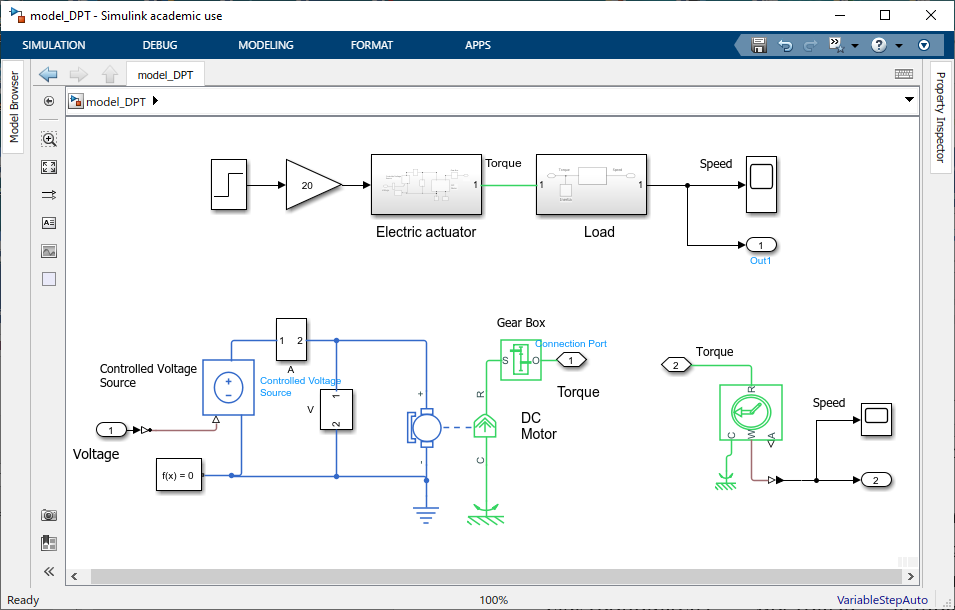
*Рис. 9.* Модель ОУ в виде блока State-Space, окна настроек ее блоков и результат симуляции

**2.5. Создание физической модели с помощью библиотеки компонентов Simulink/Simscape**

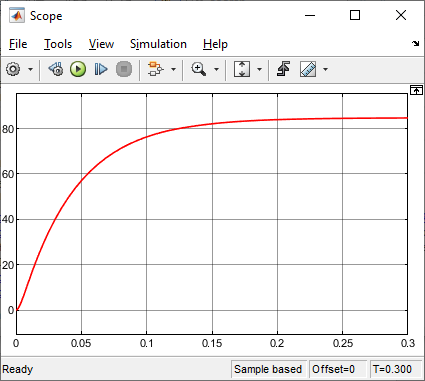
Simscape предназначен для создания моделей физических систем в среде Simulink. Моделируются такие узлы систем системы, как электродвигатели и электроприводы, мостовые выпрямители, гидроприводы и др. Дополнительные продукты Simscape предоставляют более сложные компоненты и инструменты анализа.

В качестве примера кратко рассмотрим создание физической модели ДПТ (см. файл model\_DPT.slx). Более подробному рассмотрению данной модели будет посвящена отдельная лекция.

Модель ДПТ и ее составные части (электрическая, механическая и измерительная) представлены на рис. 10. Эта модель создана с использованием стандартных блоков библиотеки Simscape. На рис. 11 изображена переходная характеристика по скорости вращения вала двигателя рассматриваемой модели. Как можно заметить, эта характеристика практически совпадает с полученными ранее.



*Рис. 10.* Simscape-модель ДПТ



*Рис. 11.* Переходная характеристика ДПТ

Итак, мы получили реализацию модели объекта управления (ДПТ) различными способами. Что дальше?

После получения математического описания объекта управления и проверки его адекватности реальному объекту с помощью моделирования следует решить основную задачу проектирования − задачу синтеза устройства управления (или регулятора), обеспечивающего требуемое качество управления в соответствии с техническим заданием.

Синтез регулятора предполагает получение его модели в том или ином виде. Конечной же целью проектирования для нас в рамках нашего курса будет реализация регулятора на языках Си и ST для микроконтроллеров и ПЛК соответственно и проверка их работы в составе системы управления.

**Замечание**

Все модели ОУ, рассмотренные в данной лекции имеют непрерывный характер. Это связано с тем, что как правило подавляющее большинство реальных объектов описывается в непрерывном времени. Если же при решении задач синтеза систем управления требуется представить ОУ или систему в целом в дискретной форме, то эта задача может быть решена с помощью стандартных процедур, которые рассматривались в курсе «Цифровые системы управления». В нашем курсе этим вопросам будет посвящена отдельная лекция.

В дальнейшем как в лекционном материале, так и на практических занятиях мы рассмотрим различные способы синтеза регуляторов для данных моделей. Например, для Simulink-модели рассмотрим автоматизированную настройку ПИД-регулятора с помощью стандартного блока PID Controller, а также с помощью алгоритма нелинейной оптимизации параметров регулятора. Рассмотрим также алгоритмы модального управления.

В среде MATLAB имеются различные подходы к синтезу регуляторов. Достаточно удобным является инструментарий для синтеза систем с одним входом и с одним выходом, так называемых SISO-систем. Работа с этим инструментом рассматривается в одной из лабораторных работ. Кроме того, в среде MATLAB имеются специальные встроенные функции синтеза и настройки ПИД-регуляторов, в частности, функция pidtune.

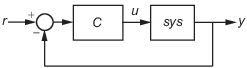
Функция pidtune настраивает параметры контроллера PID на баланс производительности (время отклика, полоса пропускания) и устойчивости (запас по фазе).

Синтаксис этой фукции:

C = pidtune(sys,type)

C = pidtune(sys,type,wc)

…………………………



*Рис. 12.* Схема системы с ПИД-регулятором

Схема системы с ПИД-регулятором для функции pidtune приведена на рис. 12. Работа этой функции осуществляется по тому же алгоритму, что и настройка блоков PID Controller в среде Simulink. Контроллер (PID, PI, PD и др.) может быть, как непрерывным, так и дискретным.

Ознакомьтесь с работой функции pidtune самостоятельно.

Пример:

>> sys = tf(1,[1 3 3 1]); %ПФ исходной системы

% Design PID with 45 degrees of phase margin

>> Options = pidtuneOptions('PhaseMargin',45);

>> [C info] = pidtune(sys, 'pid', Options)

C =

1

Kp + Ki \* --- + Kd \* s

s

with **Kp = 2.18, Ki = 0.933, Kd = 1.27**

Continuous-time PID controller in parallel form.

info =

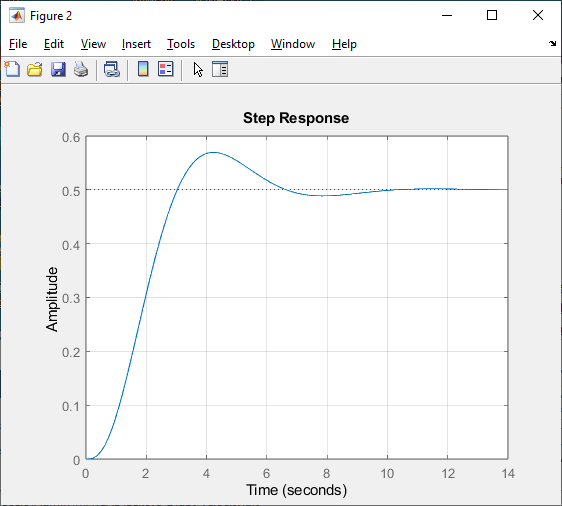
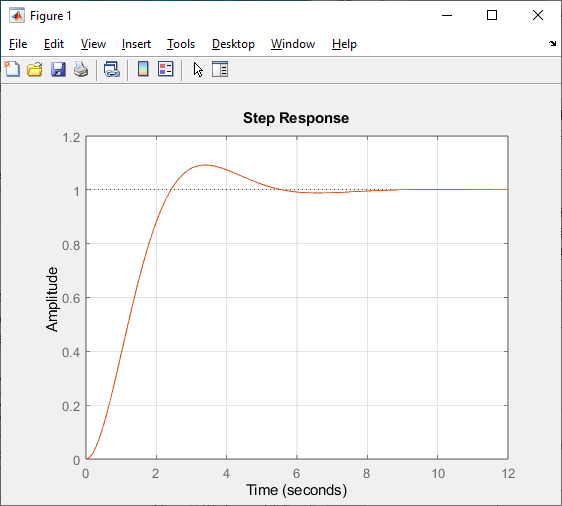
struct with fields:

Stable: 1

CrossoverFrequency: 0.8250

PhaseMargin: 59.2403

Результат настройки представлен на рис. 13.

а) б)

*Рис. 13*. Переходные процессы: a) − в исходной системе, б) − в системе

с PID-регулятором

**Задание к Практической работе 2**

1. Выбрать в соответствии с вариантом паспортные данные двигателя постоянного тока, представленные в лекции 2, табл. 1.
2. Составить математическую модель ДПТ всеми рассмотренными методами.
3. Представить разработанные модели и результаты моделирования в отчете по практической работе.
4. Сделать выводы.

*Таблица 1*

**Паспортные данные двигателей постоянного тока**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер  п/п | Марка  двигателя | *Р*н ,  Вт | ɷн ,,  рад/с | *U*н,  В | *I* н ,  А | *М*н ,  Н·м | *J*дв∙10−4, кг·м2 | *R*я,  Ом | *L*я,  мГн |
| 1 | СЛ-121 | 7,5 | 470 | 110 | 0,16 | 0,014 | 0,038 | 240 | 130 |
| 2 | СЛ-161 | 8,6 | 420 | 110 | 0,21 | 0,02 | 0,052 | 170 | 125 |
| 3 | СЛ-221 | 13 | 377 | 110 | 0,25 | 0,034 | 0,137 | 117 | 230 |
| 4 | СЛ-240 | 18 | 377 | 24 | 1,9 | 0,049 | 0,172 | 1,32 | 0,7 |
| 5 | СЛ-281 | 26 | 545 | 24 | 2,16 | 0,049 | 0,196 | 1,15 | 0,5 |
| 6 | СЛ-321 | 8 | 315 | 110 | 0,58 | 0,123 | 0,59 | 25,8 | 130 |
| 7 | СЛ-361 | 50 | 315 | 110 | 0,75 | 0,156 | 0,687 | 20,5 | 115 |
| 8 | СЛ-369 | 55 | 377 | 110 | 0,8 | 0,147 | 0,687 | 15,2 | 90 |
| 9 | СЛ-521 | 77 | 315 | 110 | 1,07 | 0,245 | 1,67 | 8,5 | 58 |