

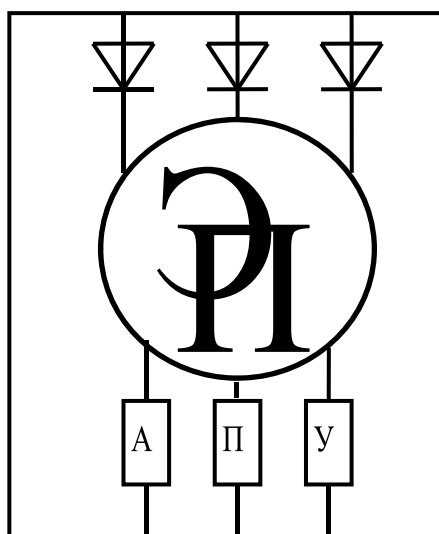
МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Электропривод и АПУ»

ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНОГО ЭКСПЕРИМЕНТА В ЭЛЕКТРОЭНЕРГЕТИКЕ

*Методические рекомендации к лабораторным работам
для магистров специальности 1-43 80 01
«Электроэнергетика и электротехника»
очной и заочной форм обучения*

Часть 2



Могилев 2021

УДК 621.3: 004.3
ББК 31.2: 32.81
И74

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой ЭПиАПУ «3» февраля 2021 г., протокол № 7

Составитель В. Н. Абабурко

Рецензент канд. техн. наук С. В. Болотов

Цель издания методических рекомендаций – регламентация лабораторного практикума по дисциплине «Информационное обеспечение вычислительного эксперимента в электроэнергетике» для магистрантов специальности 1-43 80 01 «Электроэнергетика и электротехника».

Методические рекомендации содержат состав второй части курса лабораторных работ по указанной дисциплине. Они включают: цель, описание задания, ход выполнения работ, требования к содержанию отчета и контрольные вопросы к лабораторным работам № 5–8.

Учебно-методическое издание

ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНОГО ЭКСПЕРИМЕНТА В ЭЛЕКТРОЭНЕРГЕТИКЕ

Часть 2

Ответственный за выпуск	Г. С. Леневский
Корректор	А. А. Подошевка
Компьютерная верстка	Е. В. Ковалевская

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 31 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский
университет, 2021

Содержание

5 Лабораторная работа № 5. Построение динамических моделей цифровых систем управления электроэнергетики в среде MATLAB/Simulink.....	4
6 Лабораторная работа № 6. Оптимизация параметров СУ электроэнергетики с помощью MATLAB	17
7 Лабораторная работа № 7. Оптимизация структуры электротехнической системы с помощью моделирования.....	26
8 Лабораторная работа № 8. Моделирование нейронных сетей в электроэнергетике	34
Список литературы	45

Часть 2

5 Лабораторная работа № 5. Построение динамических моделей цифровых систем управления электроэнергетики в среде MATLAB/Simulink

5.1 Цель лабораторной работы № 5

1 Изучение методов составления, анализа и преобразования дискретных математических моделей цифровых систем управления (СУ) электроэнергетики.

2 Получение практических навыков работы при анализе линейных моделей метауровня электроэнергетических систем в среде MATLAB.

5.2 Задание к лабораторной работе № 5

Заданием к лабораторной работе является построение компьютерной модели одноконтурной электромеханической системы частотного регулирования скорости асинхронного электродвигателя (АД) с короткозамкнутым ротором с цифровым ПИД-регулятором скорости, описываемого дискретной передаточной функцией

$$W_{pc}(z) = K_{п} + \frac{K_{и} \cdot T_s}{z-1} + \frac{K_{д}(z-1)}{z}, \quad (5.1)$$

где $K_{п}$ – коэффициент передачи пропорциональной части регулятора;

$K_{и}$ – коэффициент передачи интегрирующей части регулятора;

$K_{д}$ – коэффициент передачи дифференциальной части регулятора.

Для заданной на рисунке 5.1 метамоделли в виде структурной схемы варианты индивидуальных заданий параметров передаточных функций указаны в таблице 5.1.

Требуется выполнить следующие операции:

1) в среде MATLAB создать m-файл и задать параметры передаточных функций согласно варианту задания;

2) в среде Simulink спроектировать компьютерную модель на основании математической модели рисунка 5.1, в которой реализовать гибридную модель с дискретной частью для цифровой системы управления и непрерывной частью для моделей преобразователя частоты и электродвигателя;

3) реализовать ступенчатую модель изменения рабочей реактивной нагрузки от нулевого до заданного значения;

4) выполнить вычислительный эксперимент с созданной компьютерной моделью;

5) зафиксировать динамические характеристики скорости и момента АД, частоты на выходе преобразователя и цифрового кода на выходе СУ;

б) определить показатели качества регулирования скорости: конечное значение скорости, величину статической ошибки, максимальную динамическую ошибку скорости при набросе нагрузки, время регулирования скорости, число колебаний;

7) построить график статической характеристики замкнутой системы регулирования скорости.

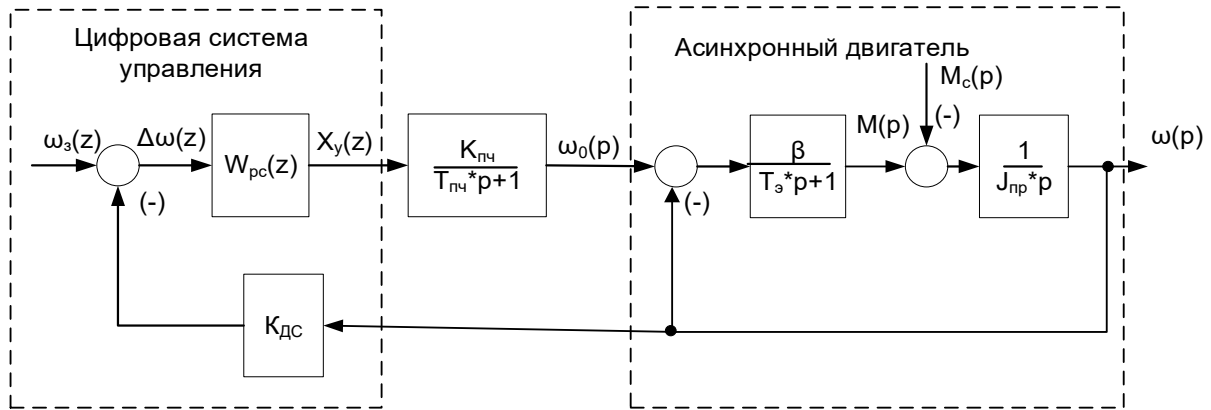


Рисунок 5.1 – Математическая модель задания к лабораторной работе № 5

Таблица 5.1 – Параметры метамодели для вариантов заданий

Вариант	1	2	3	4	5	6	7	8	9	10
$\omega_z, \text{с}^{-1}$	300	260	220	200	180	160	120	100	80	50
$\omega_{z\max}, \text{с}^{-1}$	314	314	314	314	314	314	314	314	314	314
$J_{\text{пр}}, \text{кг}\cdot\text{м}^2$	0,04	0,05	0,055	0,06	0,08	0,1	0,12	0,14	0,18	0,34
$K_{\text{пч}}$	1	1	1	1	1	1	1	1	1	1
$T_{\text{пч}}, \text{с}$	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01
$K_{\text{дс}}, \text{с}$	1	1	1	1	1	1	1	1	1	1
$\beta, \text{Н}\cdot\text{м}\cdot\text{с}$	0,240	0,452	0,632	0,837	1,468	1,463	5,642	5,643	8,663	10,52
$T_s, \text{с}$	0,092	0,123	0,141	0,142	0,246	0,253	0,381	0,419	0,353	0,414
$K_{\text{п}}$	2,11	1,69	1,08	2,10	1,52	1,318	1,23	1,04	0,97	0,68
$K_{\text{и}}$	9,73	10,93	10,90	12,92	6,1	5,52	5,02	4,04	3,98	5,04
$K_{\text{д}}$	0,11	450	650	750	800	880	900	1400	1120	960
$M_c, \text{Н}\cdot\text{м}$	3	4	5	8	10	12	15	24	35	42
$t_{\text{н}}, \text{с}$	2	2,5	3	3	3	3	3	3	3	3
$t_{\text{сч}}, \text{с}$	4	5	5	5	5	5	5	5	5	5

На рисунке 5.1 и в таблице 5.1 введены следующие обозначения:

- ω_z – цифровой код задания скорости;
- $\Delta\omega$ – цифровая ошибка задания скорости;
- X_y – цифровой код управления на входе преобразователя;
- ω_0 – частота питания статора асинхронного двигателя;
- M – электромагнитный момент на валу двигателя;
- M_c – момент сил статических сопротивлений;
- ω – скорость вращения;
- $\omega_{z\max}$ – максимальное значение цифрового кода задания скорости;

- $J_{пр}$ – момент инерции привода, приведенный к валу двигателя;
- $K_{пч}$ – коэффициент передачи преобразователя по частоте;
- $T_{пч}$ – постоянная времени преобразователя по частоте;
- $K_{дс}$ – коэффициент передачи цифрового датчика скорости;
- β – жесткость механической характеристики двигателя;
- $T_э$ – эквивалентная постоянная времени электрической цепи системы;
- $K_п$ – коэффициент передачи пропорциональной части цифрового регулятора скорости;
- $K_{и}$ – коэффициент усиления интегрирующей части цифрового регулятора скорости;
- $K_д$ – коэффициент усиления дифференцирующей части цифрового регулятора скорости;
- t_n – время наброса реактивной нагрузки;
- $t_{сч}$ – время окончания расчетов вычислительного эксперимента.

5.3 Ход выполнения работы № 5

5.3.1 Загрузка среды MATLAB/Simulink.

Для выполнения работы первоначально следует выполнить на персональном компьютере запуск программной среды математической системы MATLAB/Simulink аналогично п. 4.3.1.

5.3.2 Создание файла Simulink-модели.

Создание новой S-модели в окне Simulink выполняется выбором пункта *Blank Model*. Для удобства модификации параметров модели следует в окне MATLAB создать m-файл исходных данных комбинацией клавиш *Ctrl+N*. Для S-модели необходимо указать сопроводительную информацию в специальном окне, вызываемом с помощью раздела S-модели *File/Model Properties*.

5.3.3 Формирование непрерывной части модели исследуемой системы.

В непрерывную часть модели входит асинхронный двигатель с нагрузкой и преобразователь частоты. Модель АД выполняется аналогично п. 2.3.2 на основе из блоков *Transfer Fcn* (из *Continuous*) и сумматоров *Sum* (из *Math Operation*). Метамодель преобразователя частоты выполняется аналогично п. 2.3.4 на основе блока *Transfer Fcn* (из *Continuous*). В параметрах блоков *Transfer Fcn* указываются имена переменных среды MATLAB, представляющие соответствующие исходные данные в m-файле. В параметрах блоков сумматоров указывается требуемое расположение знаков и ориентация входов. При необходимости моделирующий блок поворачивается клавишами *Ctrl+R*. Пример модели непрерывной части исследуемой электромеханической системы показан на рисунке 5.2.

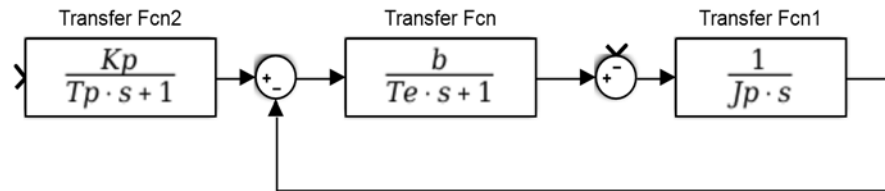


Рисунок 5.2 – Пример построения S-модели непрерывной части системы

5.3.4 Моделирование момента сил нагрузки.

Моделирование изменения ступенчатой нагрузки на валу двигателя выполняется с помощью блока *Step* (из *Sources*). В параметрах которой указывается первоначальным значением ноль, а конечным – M_c из таблицы 5.1. Заданное время переключения t_n устанавливается в поле *Sample Time*. Для моделирования реактивного характера нагрузки следует перемножить выход блока *Step* со знаком выходного сигнала скорости двигателя с помощью блока *Product* (из *Math Operations*). Знак сигнала выделяется блоком *Sign* (из *Math Operations*), на вход которого подается выходной сигнал скорости системы. Модель системы примет вид, представленный на рисунке 5.3.

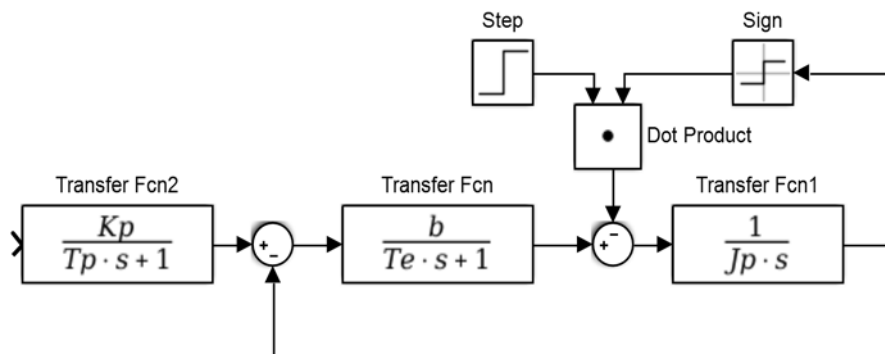


Рисунок 5.3 – Пример моделирования реактивной ступенчатой нагрузки

5.3.5 Создание модели цифровой системы управления.

Цифровая система управления, за исключением дискретного датчика скорости, должна помещаться внутри подмодели *Subsystem* (из *Ports&Subsystems*). На единственный вход подмодели цифровой системы управления подается выход сумматора блока *Sum* разность между кодом задания скорости и кодом отрицательной обратной связи по скорости двигателя.

Цифровой датчик скорости с достаточной точностью описывается с помощью блока *Gain* (из *Math Operations*), на вход которого подается сигнал с выхода исследуемой системы. Параметром блока задается переменная коэффициента обратной связи по скорости из *m*-файла. К выходу датчика скорости подключается блок дискретизации *Quantizer* (из *Discontinuities*). В качестве параметра блока *Quantization interval* устанавливается значение длительности периода квантования системы, которое для всех вариантов принимается равным 0,01 с. Выход блока дискретизации подключается к отрицательному входу сумматора *Sum* для сравнения с задающим воздействием.

Постоянное задающее воздействия моделируется блоком *Constant* (из *Sources*). В качестве параметра устанавливается значение кода задания скорости ω_3 из таблицы 5.1. Выход блока подается на положительный вход сумматора.

Выход подмодели цифровой системы управления подключается через блок экстраполятора нулевого порядка *Zero-Order Hold* (из *Discrete*) к входу блока метамодели преобразователя частоты. При этом S-модель исследуемой системы примет вид, показанный на рисунке 5.4.

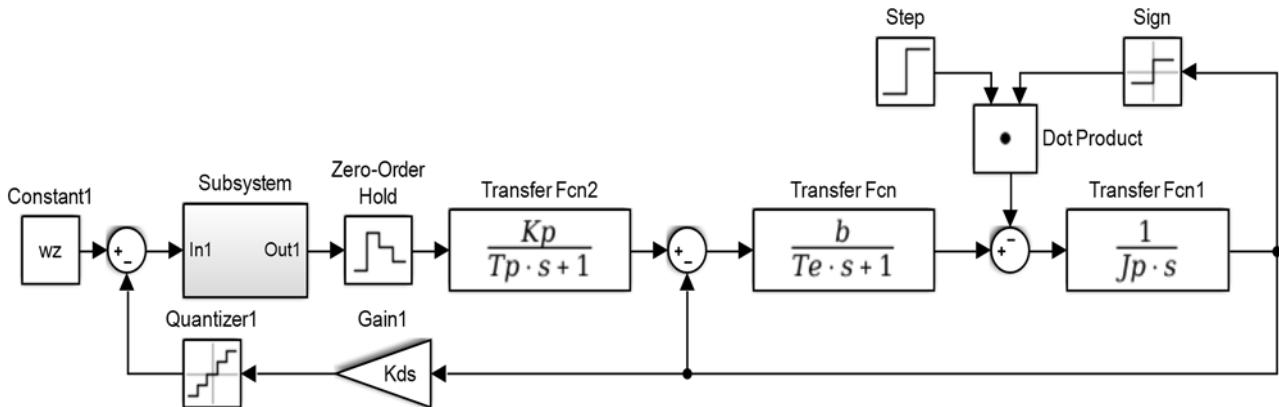


Рисунок 5.4 – Модель цифровой и непрерывной части электромеханической системы

5.3.6 Создание модели цифровой системы управления.

На рисунке 5.5 показан вид окна настроек дискретного интегратора.

Выполнив клик по блоку *Subsystem*, вызывается окно подмодели цифрового регулятора, который реализуется как параллельное соединение трех составляющих согласно выражению (5.1). Пропорциональная составляющая моделируется блоком *Gain* (из *Math Operations*), в качестве параметра которого указывается значение локальной переменной, хранящее данные K_p из таблицы 5.1.

Дискретная интегрирующая составляющая цифрового регулятора моделируется блоком *Discrete-Time Integrator* (из библиотеки *Discrete*). Параметр *Gain* данного блока устанавливается в значение локальной переменной, хранящее данные K_i из таблицы 5.1. Для ограничения значения выхода составляющей следует установить маркер в позицию *Limits output*. Эта операция активирует два поля ограничения: в *Upper saturation limit* указывается локальная переменная с данными $\omega_{3\max}$ из таблицы 5.1, в *Lower saturation limit* задается отрицательное значение локальной переменной с данными $\omega_{3\max}$, как это показано на рисунке 5.5.

Дифференцирующая составляющая цифрового регулятора описывается блоками *Gain* (из *Math Operations*) и *Difference* (из *Discrete*). В параметре *Gain* указывается имя локальной переменной, хранящей данные K_d из таблицы 5.1.

Выходы трех составляющих заводятся на три положительных входа сумматора *Sum* (из *Math Operations*). Выход сумматора подается на блок ограничения *Saturation* (из *Discontinuities*). В поле блока *Upper saturation limit* записывается локальная переменная с данными $\omega_{3\max}$, а в *Lower saturation limit* – отрицательное значение локальной переменной с данными $\omega_{3\max}$.

Пример состава подмодели цифрового регулятора показывается на рисунке 5.6.

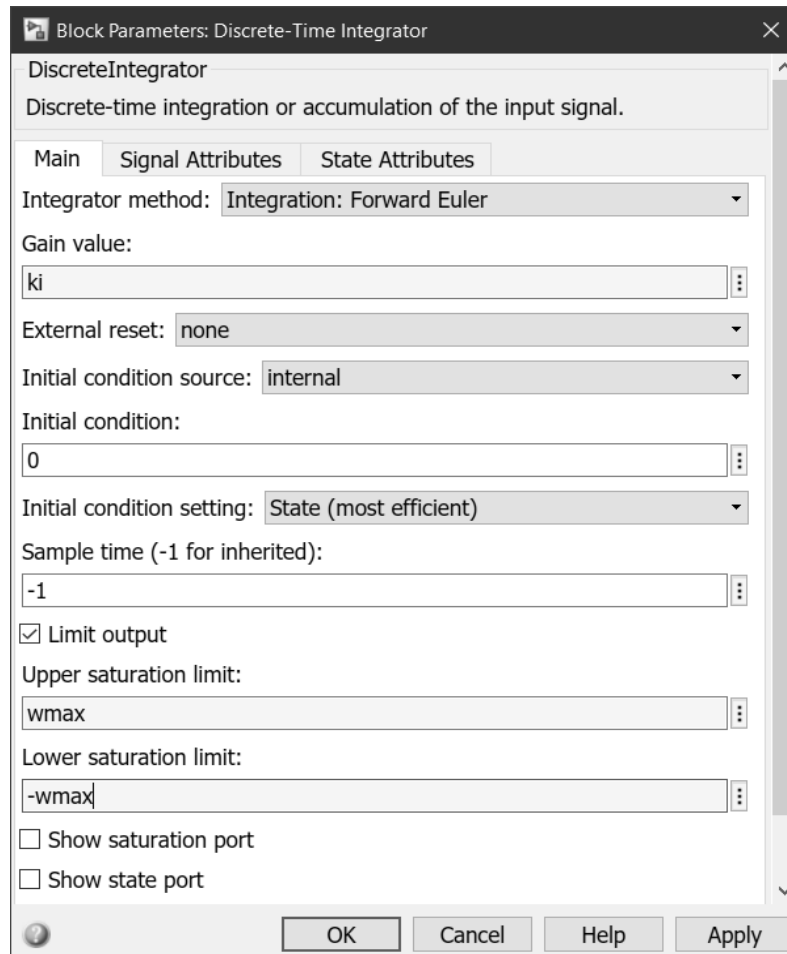


Рисунок 5.5 – Пример настройки моделирующего блока интегрирующей части

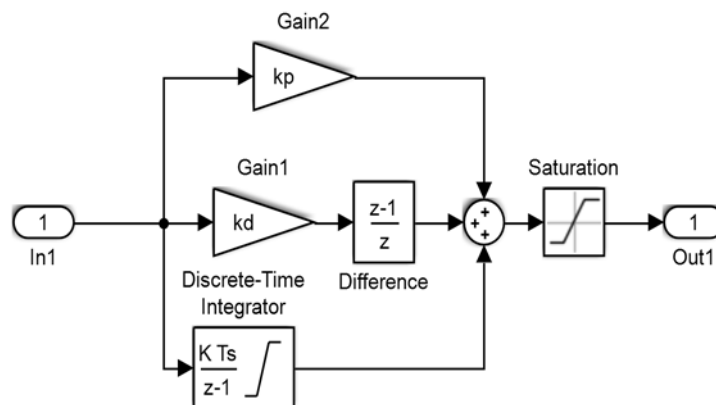


Рисунок 5.6 – Пример состава подмодели цифрового регулятора системы

5.3.7 Описание локальных параметров подмодели цифрового регулятора.

Для придания универсальности подмодели цифрового регулятора, а также обеспечения возможности ее повторного использования в других S-моделях, следует выполнить ее оформление в качестве законченного блока моделирования с локальными переменными. Для этого следует вернуться к модели исследуемой системы нажатием кнопки *Up To Parent* на панели инструментов подмодели. В окне S-модели системы выполняется клик правой кнопкой мыши по изображению подмодели, вызвав всплывающее меню, в котором следует выбрать пункт *Mask/Edit Mask* (или нажать клавиши *Ctrl+M*).

В распахнувшемся окне следует перейти на вкладку *Params&Dialogs*, в которой следует добавить четыре однострочных текстовых редактора *Edit* для задания данных пропорциональной, интегрирующей и дифференцирующей составляющих передаточной функции регулятора, а также значения максимального ограничения входа регулятора. Для этого в левой части выполняется клик мышью по элементу *Edit*, что добавляет поле строки редактора текста, параметры которого настраиваются в правой части окна в следующих разделах:

- *Name* – имя локальной переменной, которая используется внутри подмодели;
- *Value* – численное значение локальной переменной по умолчанию;
- *Prompt* – строка описания назначения локальной переменной;
- *Type* – тип элемента управления *Edit*.

Пример задания локальных переменных, описывающих параметры подсистемы цифрового ПИД-регулятора, показан на рисунке 5.7.

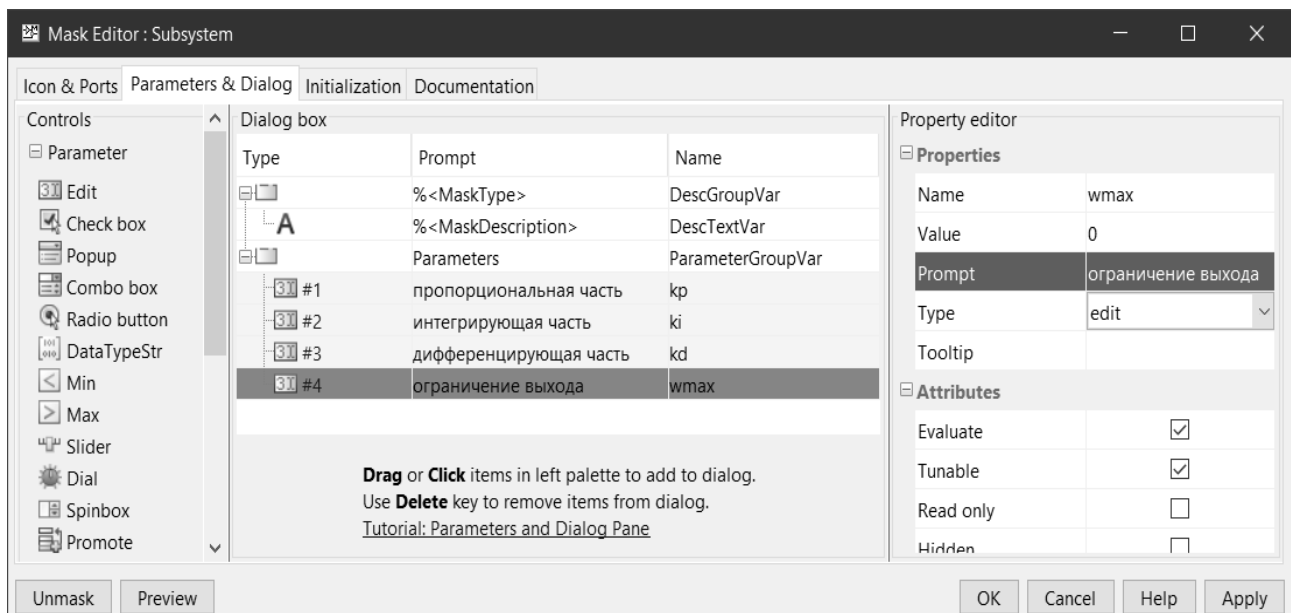


Рисунок 5.7 – Пример задания локальных переменных-параметров подмодели

Для указания текста надписи, отображающейся на теле блока подмодели, выполняется переход в окне на вкладку *Icons&Ports*. В поле *Icon drawing commands* задается команда *disp('текст подписи')*. Пример отображения надписи PID показан на рисунке 5.8.

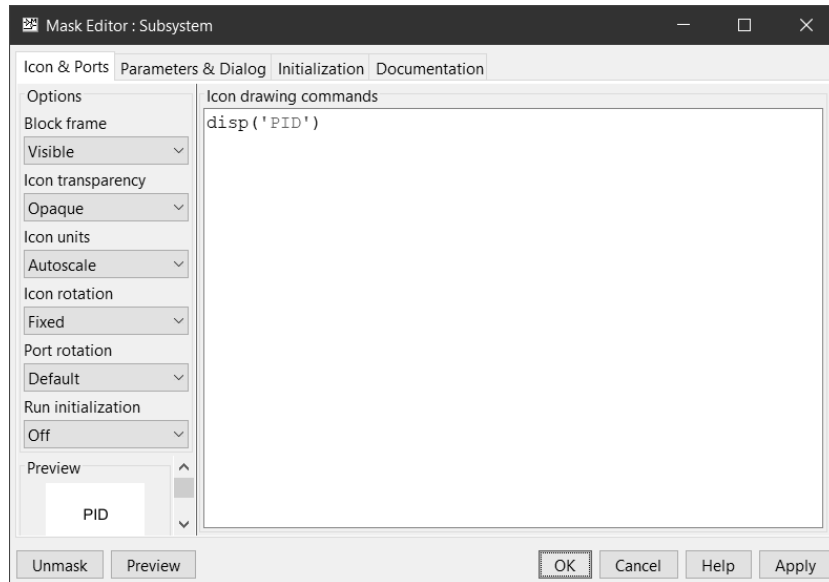


Рисунок 5.8 – Пример формирования текстовой надписи на блоке подмодели

Далее необходимо выполнить пояснение назначения созданной подмодели. Для этого выполняется переход на вкладку *Documentation*. В поле *Description* добавляется описание назначение блока подмодели, как это показано на рисунке 5.9.

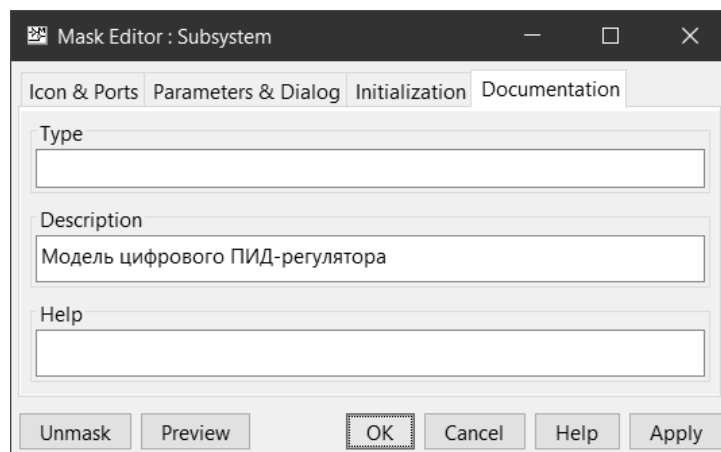


Рисунок 5.9 – Создание описания назначения блока подмодели

Завершение работы с окном настроек выполняется нажатием кнопки *OK*.

Для загрузки данных в подмодель из *m*-файла, выполняется клик мышкой по блоку подсистемы, что вызывает раскрытие диалогового окна настройки указанных параметров подмодели. В полях редакторов записываются имена соответствующих переменных среды MATLAB. Пример показан на рисунке 5.10.

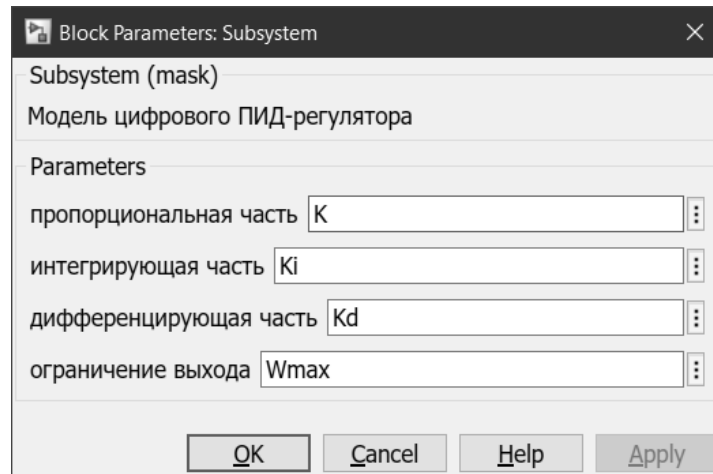


Рисунок 5.10 – Пример задание значений параметров подмодели регулятора

5.3.8 Добавление блоков фиксации результатов вычислений.

Для снятия динамических характеристик скорости, момента, частоты и выхода цифровой системы управления используются блоки *Scope* (из *Sinks*). Также необходимо добавить блоки отображения численного значения для момента и скорости двигателя *Display* (из *Sinks*). Пример показан на рисунке 5.11.

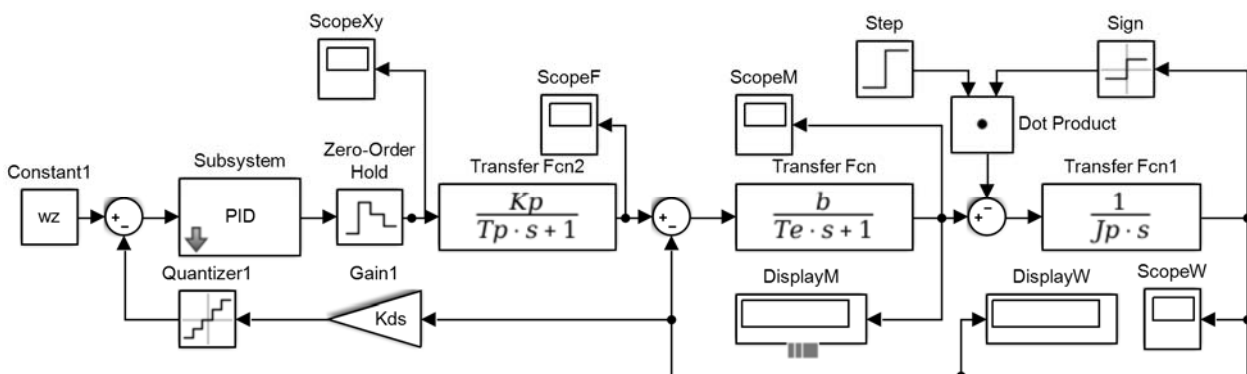


Рисунок 5.11 – Пример фиксации результатов вычислительного эксперимента

При необходимости передать результаты расчетов в глобальные переменные среды MATLAB следует выполнить следующие операции: в окне настройки параметров блока *Scope* перейти на вкладку *Logging*, установить маркер в позицию *Log data to workspace*, указать наименование переменной в поле *Variable name*, выбрать в выпадающем списке *Save format* значение *Array*.

5.3.9 Формирование состава *m*-файла с исходными данными.

После завершения работы по созданию S-модели следует перейти в окно редактора текста файла исходных данных. Рекомендуется вначале файла использовать команды очистки командного окна и пространства данных MATLAB, указав в первых двух строках команды *clc* и *clear all*.

Далее в отдельной строке указываются с помощью оператора присваивания значения каждой использованной в S-модели переменной. При этом следует

использовать только символы латинского алфавита. Имя переменной должно начинаться с буквы, но может содержать и цифры. Использование пробелов в именах не допускается. После задания значения рекомендуется после символа «%» указывать комментарий, который может содержать кириллические символы.

5.3.10 Настройка параметров модели и выполнение вычислительного эксперимента.

Для настройки параметров моделирования в окне S-модели системы выбирается пункт меню *Simulation/Model Configuration Parameters* или нажимаются клавиши *Ctrl+E*. Это отображает диалоговое окно *Configuration Parameters*. В окне устанавливается начальное время моделирования *Start Time* в значение 0, а значение времени моделирования в поле *Stop Time* должно соответствовать варианту таблицы 5.1.

В разделе окна *Solve options* следует:

- в списке *Type* выбрать тип численного метода с фиксированным шагом интегрирования (*Fixed-step*);
- в списке *Solve* указать численный метод интегрирования тетьего или четвертого порядка точности (*ode3* или *ode4*);
- в поле *Fixed-step size* указать значение шага интегрирования, которое должно быть равно 0,0001.

5.3.11 Вычислительный эксперимент и анализ результатов моделирования.

Первоначально вычисляется m-файл в среде редактора MATLAB с помощью кнопки *Run* или нажатия клавиши *F5*.

Далее проводится вычислительный эксперимент с S-моделью исследуемой системы. Запуск расчетов S-модели выполняется нажатием кнопки *Run* в панели инструментов окна модели или комбинации клавиш *Ctrl+T*.

Графики динамических характеристик переносятся в отчет с использованием раздела *Edit/Copy Figure*. При переносе графика следует использовать настройки копирования в растровый формат (метафайл).

Построение графика статической характеристики выполняется группой команд в окне *Command Windows* среды MATLAB:

```
X(1)=0           % задание начального значения момента
X(2)=M(size(W,1),2) % задание конечного значения момента
Y(1)=wz         % задание начального значения скорости
Y(2)=W(size(W,1),2) % задание конечного значения скорости
plot(X,Y)       % графика статической характеристики
grid on        % включение координатной сетки
```

Вместо наименований переменных *M* и *W* следует использовать значения переменных, указанных в полях блоков *Scope* для сигналов момента и скорости соответственно. А вместо имени переменной *wz* – наименование переменной задания скорости из m-файла. График копируется из окна в отчет через буфер обмена с помощью раздела *Edit/Copy Figure*.

Анализ показателей качества динамических характеристик процесса запуска системы на холостом ходу возможен с помощью команды *stepinfo*. Но

предварительно следует выделить часть динамической характеристики, описывающей пуск на холостом ходу с помощью команд:

```
[n]=find(M(:,1)<tn) % поиск массива номеров до времени tn
W1=W(1:max(n)-1,2) % выделение массива скоростей пуска
T1=W(1:max(n)-1,1) % выделение массива времени пуска
plot(T1,W1) % построение графика пуска
grid on % включение координатной сетки
winfo = stepinfo(W1,T1) % анализ графика скорости при пуске
```

Пример выполнения последней команды в *Command Windows* MATLAB:

```
winfo =
struct with fields:
    RiseTime: 0.1606
    SettlingTime: 1.3077
    SettlingMin: 162.0674
    SettlingMax: 222.9498
    Overshoot: 11.4380
    Undershoot: 0
    Peak: 222.9498
    PeakTime: 0.3295
```

Смысл показателей качества, возвращаемых *stepinfo* дан на рисунке 5.12.

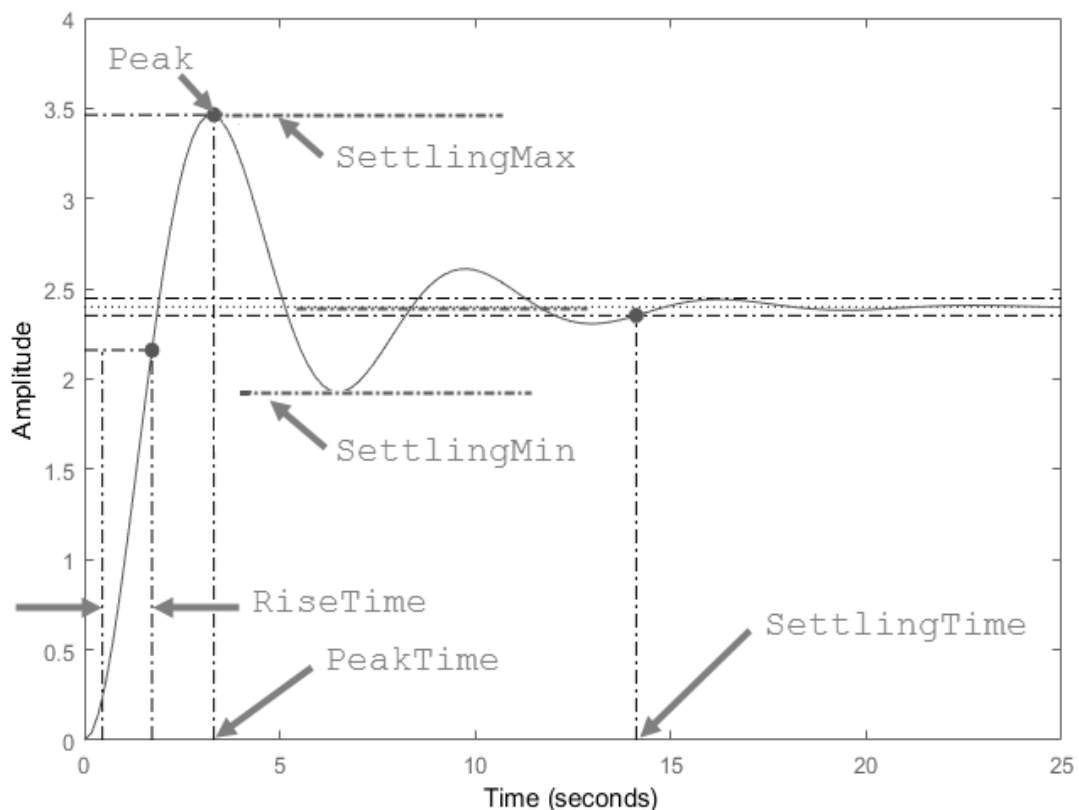


Рисунок 5.12 – Пример анализа динамической характеристики функцией *stepinfo*

Функция *stepinfo* возвращает следующие показатели:

- *RiseTime* – время нарастания характеристики в диапазоне значений от 10 до 90 % от установившегося значения;
- *SettlingTime* – время регулирования, по умолчанию принимается трубка в 2 % от конечного значения;
- *SettlingMin* – минимальное значение после роста выходной величины;
- *SettlingMax* – максимальное значение;
- *Overshoot* – величина перерегулирования в процентах;
- *Undershoot* – величина недотягивания до установившегося значения в процентах;
- *Peak* – максимальное значение анализируемой величины;
- *PeakTime* – время, когда характеристика достигает максимального значения.

Если нужно изменить настройку времени регулирования с 2 на 5 %, тогда команда *stepinfo* модифицируется следующим образом:

```
winfo = stepinfo(W1,T1, 'SettlingTimeThreshold', 0.05)
```

При анализе части динамической характеристики после наброса нагрузки используются следующие команды:

```
W2=W(max(n)+1:size(W,1),2) % выделение массива скоростей
```

```
T2=W(max(n)+1:size(W,1),1) % выделение массива времени
```

Поиск максимальной динамической ошибки скорости после наброса может выполняться также и с помощью функции *min*:

```
dWmin=(W2(size(W2,1))-min(W2))/W2(size(W2,1))*100
```

5.3.12 Окончание работы со средой MATLAB и компьютером.

Перед закрытием MATLAB выполняется сохранение всех открытых файлов. Рекомендуется выполнить резервное копирование файлов модели и отчета на мобильный носитель или в личный аккаунт компьютерной сети. Следует удалить все временные файлы черновиков или временные файлы-результаты восстановления, которые были созданы при выполнении работы или при перезагрузке операционной системы компьютера.

5.4 Содержание отчета по работе № 5

Отчеты по лабораторной работе № 5 оформляются индивидуально каждым обучающимся на листах формата А4 в соответствии с требованиями ГОСТ 7.32 на бумажном или электронном носителе в указанном ниже составе.

5.4.1 Титульный лист отчета.

5.4.2 Текст варианта индивидуального задания.

5.4.3 Текст m-файла с исходными данными системы для проведения вычислительного эксперимента.

5.4.4 Изображение реализованной Simulink-модели системы.

5.4.5 График динамической характеристики скорости вращения двигателя.

5.4.6 График динамической характеристики изменения момента двигателя.

5.4.7 График динамической характеристики изменения частоты на выходе преобразователя.

5.4.8 График динамической характеристики кода управления на входе преобразователя частоты.

5.4.9 Показатели качества регулирования скорости вращения: время регулирования, величина перерегулирования при пуске, максимальная динамическая ошибка скорости при набросе нагрузки, число колебаний скорости за время регулирования, статическая ошибка регулирования скорости после наброса нагрузки.

5.4.10 График статической характеристики замкнутой электромеханической системы регулирования скорости.

Контрольные вопросы по лабораторной работе № 5

1 Какие методы математического описания используются для цифровых систем управления электроэнергетическими системами?

2 Каким образом при математическом выполняется переход от временной области к Z-преобразованиям?

3 Каким образом в среде MATLAB формируется математическая модель цифровой части энергетической системы?

4 Как в среде MATLAB задать Z-выражение передаточной функции?

5 Какие элементы входят в состав Simulink-библиотеки Discrete?

6 Какие элементы выполняют моделирование реактивной ступенчатой нагрузки?

7 Каким образом в среде Simulink выполняется моделирование асинхронного двигателя?

8 Каким образом выполняется математическое описание и построение Simulink-модели преобразователя частоты?

9 Каким образом выполняется математическое описание и построение Simulink-модели датчика скорости вращения?

10 Каким образом в Simulink-модели реализуется ввод непрерывного сигнала в цифровую систему управления?

11 Каким образом в Simulink-модели выполняется ввод дискретного сигнала в непрерывную часть модели системы?

12 Как по графику динамической характеристики системы в графическом окне MATLAB определить величину перерегулирования?

13 Как по графику динамической характеристики системы в графическом окне MATLAB определить величину времени регулирования?

14 Какие типы экстраполяторов используются при моделировании цифровых систем управления электроэнергетики?

15 Каким образом в среде MATLAB возможно выполнение построения статической механической характеристики электромеханической системы?

16 Как в среде MATLAB выделить данные процесса пуска из массива динамической переменной системы?

17 Как в среде MATLAB выделить данные процесса наброса нагрузки из массива динамической переменной системы?

18 Каким образом в среде MATLAB можно получить показатели качества на основании массивов данных динамической переменной системы?

19 Каким образом при анализе показателей качества в среде MATLAB можно задать настройку диапазона интервала определения времени регулирования?

20 Какие параметры блоков моделирования Simulink должны учитывать интервал дискретности системы?

6 Лабораторная работа № 6. Оптимизация параметров СУ электроэнергетики с помощью MATLAB

6.1 Цель лабораторной работы № 6

1 Изучение методов составления математических моделей электромеханической системы положением для оптимизации параметров регуляторов.

2 Получение практических навыков работы по оптимизации параметров регуляторов цифровых систем управления позиционных устройств электроэнергетики в среде MATLAB/Simulink.

6.2 Задание к лабораторной работе № 6

Заданием на выполнение работы является составление динамической модели в среде *MATLAB/Simulink* с целью параметрической оптимизации настроек цифровых регуляторов двухконтурной системы управления положением исполнительного механизма. Исследуемая электромеханическая система (ЭМС) включает: два цифровых пропорционально-интегрально-дифференцирующих (ПИД) регулятора с датчиками скорости и положения, преобразователь частоты, управляющий асинхронным двигателем с короткозамкнутым ротором (АД), выход которого вращает через редуктор линейно перемещающийся исполнительный механизм. Исходная математическая модель системы имеет структурную схему, показанную на рисунке 6.1, значения параметров передаточных функций и воздействий указываются в таблице 6.1.

На рисунке 6.1 и в таблице 6.1 обозначены следующие параметры передаточных функций и сигналы:

- U_3 – величина задания положения;
- $U_{y\max}$ – максимальная величина напряжения задания положения и на входе преобразователя;
- $W_{pp}(z)$ – дискретная передаточная функция передачи ПИД-регулятора положения (РП);
- $W_{pc}(z)$ – дискретная передаточная функция передачи ПИД-регулятора скорости (РС);
- $K_{пч}$ – коэффициент передачи преобразователя частоты (ПЧ);
- $T_{пч}$ – постоянная времени преобразователя частоты;
- β – жесткость механической естественной характеристики АД;
- $T_э$ – электрическая постоянная времени АД;
- $J_{пр}$ – момент инерции системы, приведенный к валу АД;
- ρ – радиус приведения линейного перемещения исполнительного механизма к валу АД;
- $K_{дс}$ – коэффициент передачи датчика скорости (ДС);
- $K_{дп}$ – коэффициент передачи датчика положения (ДП);
- $U_3(z)$ – изображение напряжения задания с цифрового устройства;
- $\Delta U_3(z)$ – изображение кода ошибки задания скорости;
- $U_y(p)$ – изображение напряжения управления на выходе СУ;
- $\omega_0(p)$ – изображение синхронной частоты питания АД;
- $M(p)$ – изображение момента АД;
- $M_c(p)$ – изображение момента сил статических сопротивлений;
- $\omega(p)$ – изображение скорости вращения АД;
- $W_{рег}(p)$ – передаточная функция регулятора скорости (РС);
- $t_{сч}$ – время окончания вычислительного эксперимента;
- t_n – время наброса нагрузки M_c .

6.3 Ход выполнения работы № 6

6.3.1 Загрузка среды MATLAB/Simulink.

Изначально следует выполнить запуск среды системы MATLAB. После загрузки которой выполняется запуск интерфейса моделирования *Simulink*.

6.3.2 Создание файла Simulink-модели.

Для работы рекомендуется воспользоваться S-моделью и m-файлом, полученными в результате выполнения работы № 5 (см. подраздел 5.3). Для этого в окне Simulink выбирается требуемый файл в окне, вызываемом нажатием кнопки *Open*. Следует сразу выполнить сохранение открытой модели под новым именем, используя пункт меню *File/Save As*. Аналогичная процедура выполняется в главном окне MATLAB и с m-файлом исходных данных.

Необходимо также скорректировать информацию по модели в окне свойств, вызываемого выбором раздела меню S-модели *File/Model Properties*.

6.3.3 Модификация модели непрерывной части системы.

В конец исходной S-модели добавляется блок *Transfer Fcn* (из *Continuous*), описывающий интегрирующее динамическое звено метамодели исполнительного механизма. В окне настройки его параметров указывается имя глобальной переменной радиуса приведения из m-файла исходных данных. В m-файл добавляют введенные переменные. S-модель примет вид, показанный на рисунке 6.2.

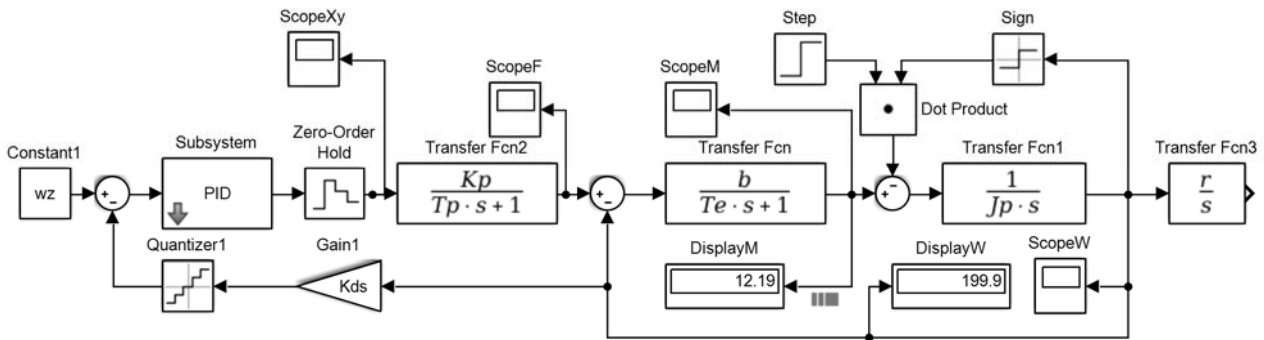


Рисунок 6.2 – Вид модифицированной непрерывной части модели системы

6.3.4 Редактирование цифровой части модели системы.

В цифровой части системы удаляется подмодель ПИД-регулятора скорости, созданная ранее. На ее место устанавливается блок *Discrete PID Controller* (из *Discrete*). Между задающим воздействием и сумматором последовательно ставится еще один блок *Discrete PID Controller* (из *Discrete*) для моделирования регулятора положения и сумматор Sum с отрицательной обратной связью по положению.

В модель следует добавить копированием или вставкой из библиотеки последовательность двух блоков, представляющих цифровой датчик положения: *Gain* (из *Math Operations*) и *Quantizer* (из *Discontinues*). В качестве параметра Gain указывается наименование глобальной переменной коэффициента передачи датчика положения, а в поле блока *Quantizer* – значение интервала дискретизации – 0,01 с. На вход ДП заводится сигнал с выхода блока исполнительного механизма. Выход *Quantizer* подается на отрицательный вход первого сумматора. Модель примет вид, показанный на рисунке 6.3.

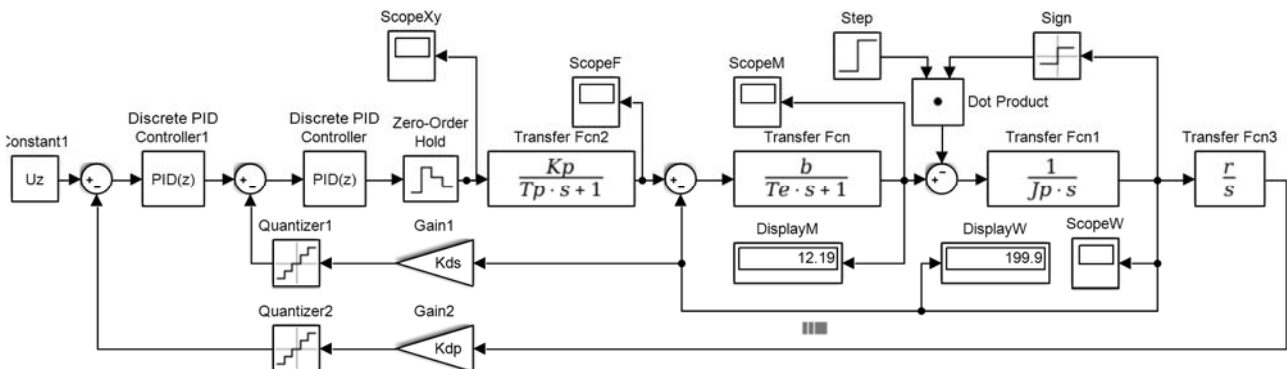


Рисунок 6.3 – Отредактированная модель цифровой части исследуемой системы

6.3.5 Редактирование блоков фиксации результатов эксперимента.

К существующему набору блоков фиксации следует добавить два элемента фиксации данных из *Sinks*:

1) блок *Scope* для хранения динамической характеристики положения, в окне параметров которого на вкладке *Logging*, устанавливается маркер в позицию *Log data to workspace*, задается имя глобальной переменной в поле *Variable name*, устанавливается в списке *Save format* значение *Array*;

2) блок *Display* для отображения численного значения положения.

S-модель исследуемой системы примет окончательный вид (рисунок 6.4).

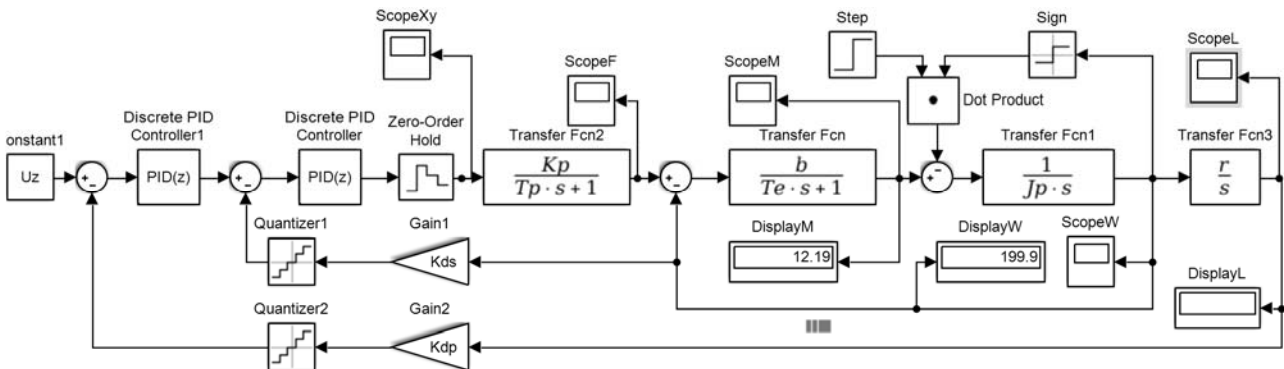


Рисунок 6.4 – S-модель исследуемой системы

6.3.6 Оптимизация регулятора скорости.

Настройка регуляторов выполняется с внутреннего контура скорости. Для оптимизации РС следует модифицировать модель, временно отключив регулятор положения и оборвав обратную связь по положению, как это показано на рисунке 6.5.

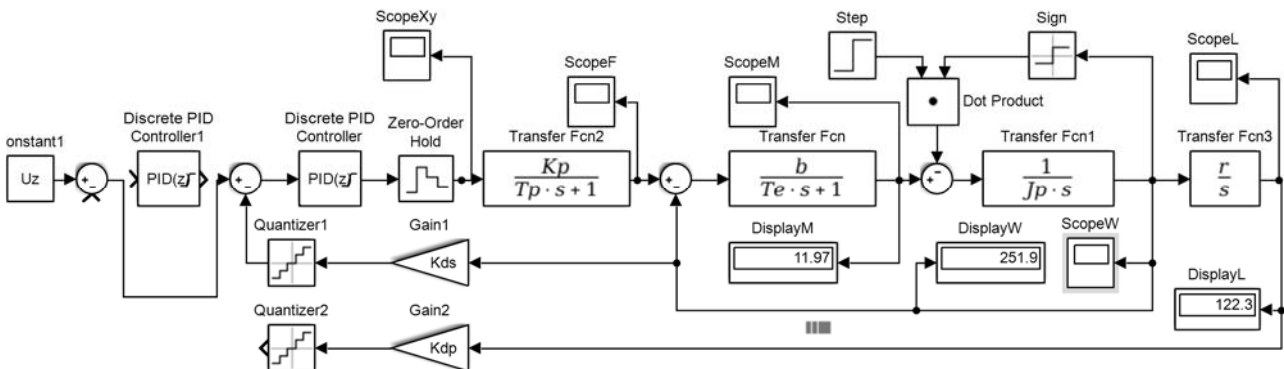


Рисунок 6.5 – Преобразование модели для настройки регулятора скорости

Перед настройкой регуляторов следует обязательно рассчитать m-файл исходных данных, нажав клавишу *F5*. Для настройки регулятора выполняется клик по блоку *Discrete PID Controller*, вызвав диалоговое окно настройки параметров. В распахнувшемся окне сначала следует перейти на вкладку *PID Advances* и установить маркер в позицию *Limit output* для задания насыщения на выходе на уровне U_{\max} . Имя глобальной переменной из файла исходных данных с этим значением должно записываться в поле *Upper saturation limit*, а в поле *Lower saturation limit* – как отрицательное значение согласно рисунку 6.6.

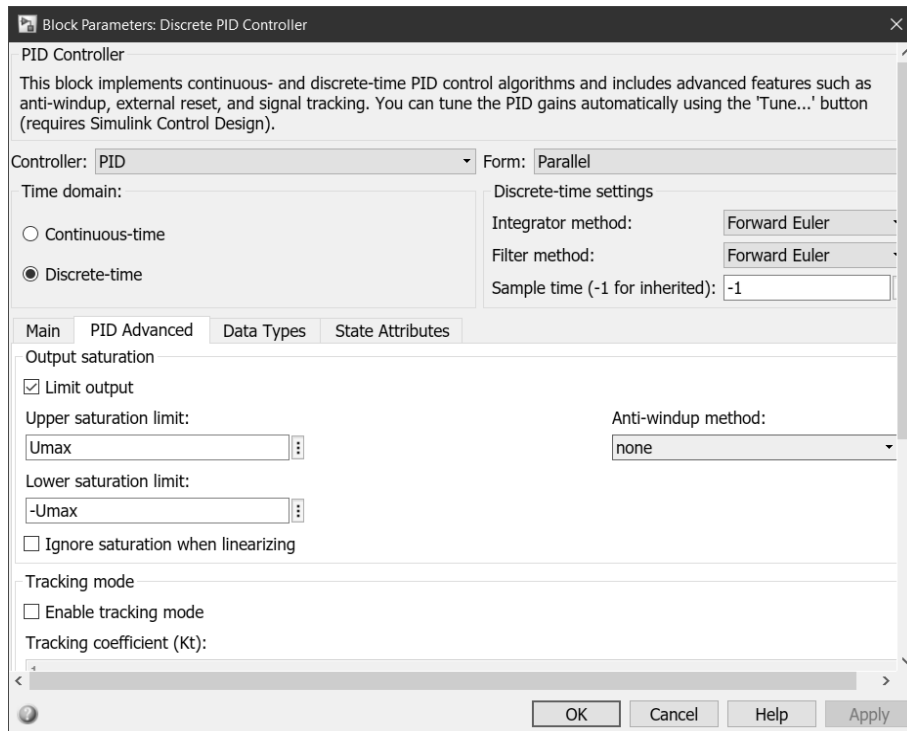


Рисунок 6.6 – Установление данных насыщения ПИД-регулятора

Затем следует перейти в окне на вкладку *Main* и нажать кнопку настройки *Tune*, показанные на рисунке 6.7.

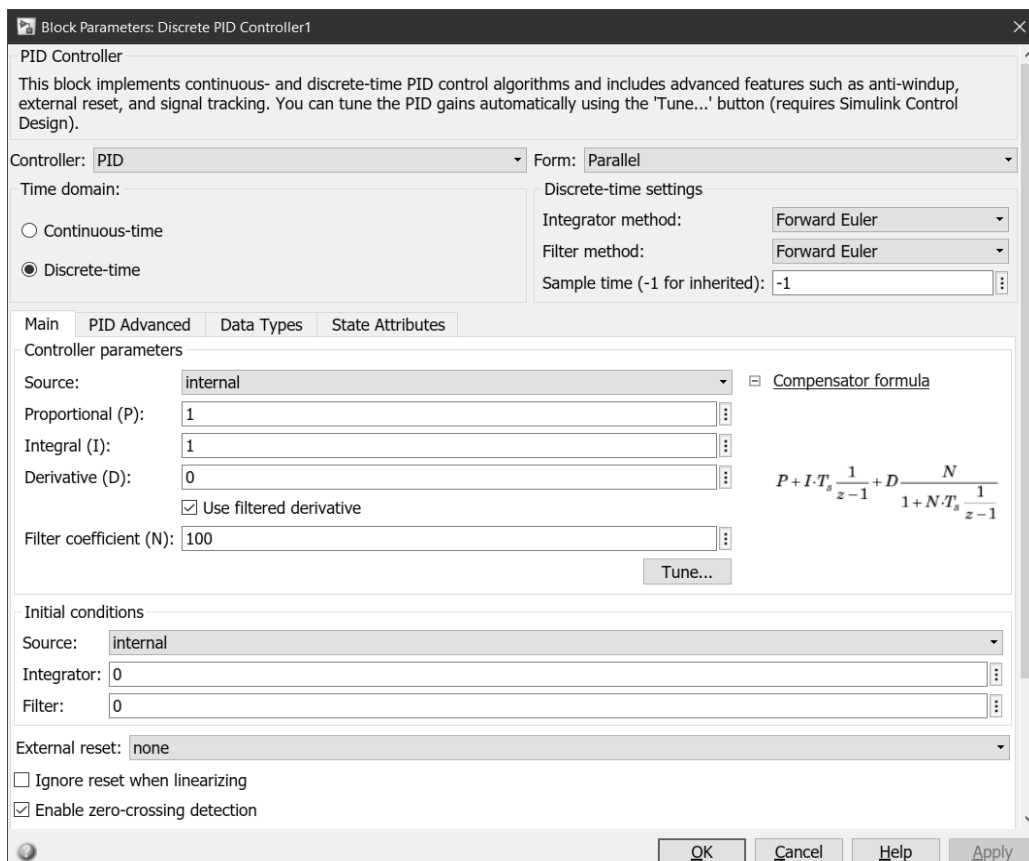


Рисунок 6.7 – Вид главного окна настройки блока *Discrete PID Controller*

После этого отобразится окно настройки параметров передаточной функции с графиком переходной характеристики, показанное на рисунке 6.8.

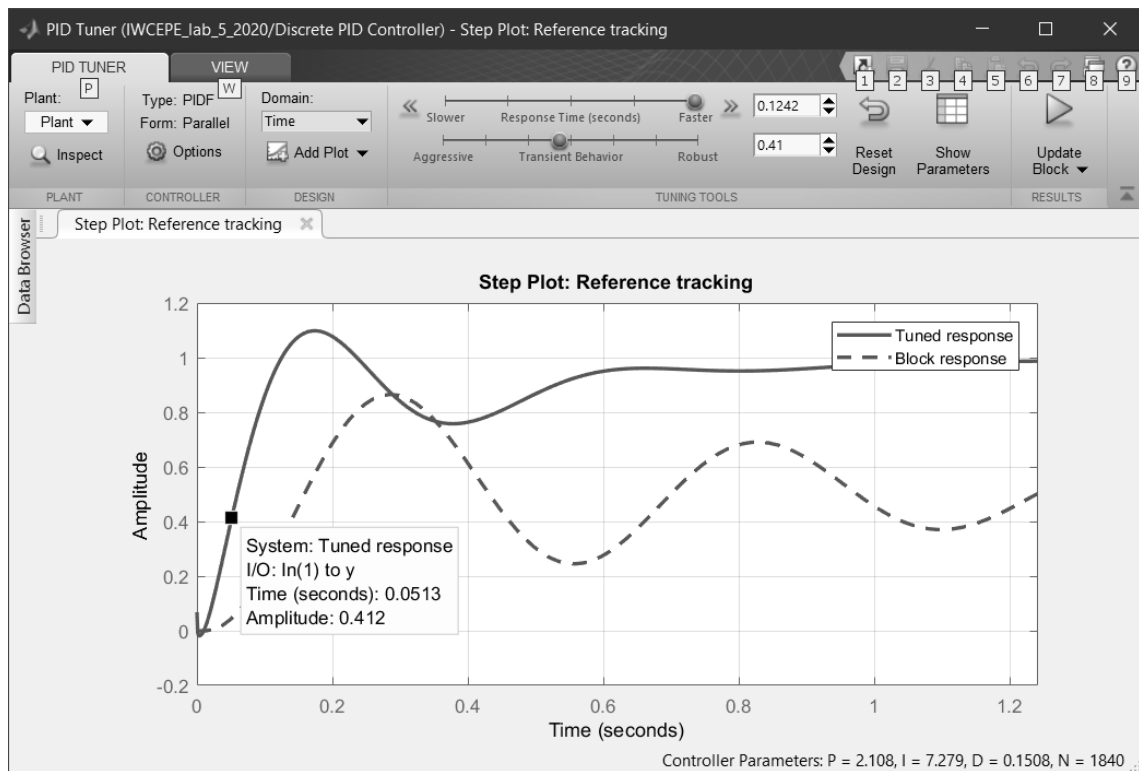


Рисунок 6.8 – Окно настройки регулятора по переходной характеристике системы

Настройка параметров регулятора выполняется с помощью двух ползунковых переключателей в верхней части окна. Сначала перемещается верхний переключатель между значениями *slower* и *faster* таким образом, чтобы число колебаний переходной характеристики не превышало три. Время регулирования должно быть не более чем половина значения времени t_n из таблицы 6.1. С помощью нижнего переключателя формируется желаемый вид переходной характеристики, величина перерегулирования которой не должна превышать 50 %. Скорректированная переходная характеристика отображается на графике сплошной линией, а предыдущий вариант настройки – штриховой. При изменении настроек параметры регулятора отображаются в строке статуса окна. В случае получения приемлемого вида переходной характеристики выполняется сохранение параметров настроек регулятора нажатием кнопки *Update Block*. Окно настроек параметров может быть закрыто.

Для проверки настройки регулятора выполняется тестовый вычислительный эксперимент с настроенным временем счета, равным значению t_n , которое можно указать в поле *Simulation Stop Time* в окне S-модели. Настройки параметров численного интегрирования совпадают с данными п. 5.3.10. Запуск расчета выполняется нажатием кнопки *Run* или клавиш *Ctrl+T*. Если вид динамической характеристики скорости удовлетворяет вышеперечисленным требованиям, тогда график характеристики копируется через буфер обмена в электронный файл отчета с помощью пункта меню *File/Copy to Clipboard* или иным образом, указанным

в п. 5.3.11. Также в отчет помещается изображение S-модели, использованной отладки регулятора и выполняется копия главного окна настроек регулятора или переписываются данные полей *Proportional*, *Integral*, *Derivative* и *Filter coefficient*.

6.3.7 Настройка регулятора положения.

Для настройки регулятора положения S-модель возвращается к виду, показанному на рисунке 6.4. Процедура настройки ПИД-регулятора положения аналогична описанной в п. 6.3.6. Отличием является то, что время регулирования переходной характеристики положения должно быть больше, чем время регулирования контура скорости не менее чем в полтора раза. Также выполняется тестирование переходной характеристики регулирования положения без наброса нагрузки путем проведения вычислительного эксперимента. В отчет копируется график переходной характеристики положения, а также полученные параметры регулятора положения.

6.3.8 Проведение вычислительного эксперимента с S-моделью.

Настройка параметров времени моделирования согласно варианту таблицы 6.1 выполняется в окне S-модели в поле *Simulation Stop Time*. При необходимости более детальной настройки выбирается пункт *Simulation/Model Configuration Parameters (Ctrl+E)*. В разделе окна *Solve options* следует уменьшить порядок точности численного метода с фиксированным шагом интегрирования (*Fixed-step*) в списке *Solve* до второго или третьего значения. Значение шага интегрирования 0,0001, показанное ранее в поле *Fixed-step size*, изменять не рекомендуется.

6.3.9 Фиксация и анализ результатов моделирования.

Запуск вычислительного эксперимента с S-моделью производится кнопкой *Run* или нажатием клавиш *Ctrl+T*. Если за указанное в таблице 6.1 время расчета система не успела отработать заданное перемещение, тогда изменяется настройка регулятора положения. Увеличить время расчета можно лишь в том случае, если после перенастройки регулятора динамический процесс положения имеет большое перерегулирование или число его колебаний за время регулирования превышает 3. Полученные графики динамических характеристик напряжения управления, момента, скорости и положения копируются в отчет с помощью *Edit/Copy Figure* с использованием векторного формата (метафайла).

Анализ показателей качества динамической характеристики внешнего контура регулирования положения выполняется аналогично п. 5.3.9.

Построение статической характеристики положения от значения момента нагрузки производится по методике п. 5.3.9, заменив переменную скорости на переменную положения.

6.3.10 Окончание работы со средой MATLAB.

По окончании работы со средой моделирования выполняется сохранение всех открытых файлов и закрывается главное окно MATLAB. Выполняется резервное копирование файлов на мобильный носитель или в сеть. Следует удалить все ненужные файлы из внешней памяти компьютера.

6.4 Содержание отчета по работе № 6

Отчеты по лабораторной работе № 6 оформляются индивидуально на листах формата А4 в соответствии с требованиями ГОСТ 7.32 на бумажном или электронном носителе в указанном ниже составе.

- 6.4.1 Титульный лист отчета.
- 6.4.2 Текст индивидуального задания со значениями параметров исходных данных.
- 6.4.3 Содержание m-файла исходных данных модели.
- 6.4.4 Simulink-модель, использованная для настройки регулятора скорости системы.
- 6.4.5 Рисунок Simulink-модели всей электромеханической системы.
- 6.4.6 Параметры настройки регулятора скорости.
- 6.4.7 Параметры настройки регулятора положения.
- 6.4.8 Динамическая характеристика настройки контура скорости.
- 6.4.9 Динамическая характеристика настройки контура положения.
- 6.4.10 Динамические характеристики положения, скорости, момента, частоты сети и изменения напряжения задания.
- 6.4.11 Статическая характеристика зависимости положения исполнительного механизма от момента нагрузки.
- 6.4.12 Результаты анализа показателей качества регулирования динамической характеристики положения исполнительного механизма: величина регулирования времени пуска, перерегулирование, число колебаний за время регулирования, максимальная динамическая ошибка при набросе нагрузки.

Контрольные вопросы по лабораторной работе № 6

- 1 Дать определение понятия «оптимизация».
- 2 Дать определение понятия «параметрическая оптимизация».
- 3 Каким образом моделируется ПИД-регулятор с помощью библиотек MATLAB/Simulink?
- 4 Каким образом выполняется настройка ПИД-регулятора скорости?
- 5 Как устанавливается ограничение выходного сигнала блока *Discrete PID Controller*?
- 6 Какие параметры используются для оптимизации настроек составляющих регулятора в блоке *Discrete PID Controller*?
- 7 Как в среде MATLAB/Simulink моделируется исполнительный механизм?
- 8 Каким образом моделируется цифровой датчик положения?
- 9 Каким образом в Simulink-модели реализуется программирование циклограммы задающего воздействия?
- 10 Каким образом выполняется построение статической характеристики зависимости положения от момента нагрузки в среде MATLAB?

11 Как определяется время регулирования динамической переменной в среде MATLAB?

12 Какие критерии используются для оптимальной настройки регулятора положения?

13 Как при моделировании в MATLAB/Simulink выполнить переход от дискретного регулятора к аналоговому (непрерывному)?

Дополнительно могут задаваться контрольные вопросы из лабораторной работы № 5: 7, 8, 12, 16, 17, 20.

7 Лабораторная работа № 7. Оптимизация структуры электротехнической системы с помощью моделирования

7.1 Цель лабораторной работы № 7

1 Изучение методов структурной оптимизации систем электроэнергетики.

2 Получение практических навыков моделирования следящих электромеханических систем с цифровыми регуляторами.

3 Получение практических навыков оптимизации структуры электромеханической системы (ЭМС) в среде *MATLAB/Simulink*.

7.2 Задание к лабораторной работе № 7

Заданием к лабораторной работе является построение Simulink-модели двух вариантов следящей ЭМС: на базе двухконтурной (математическая модель показана на рисунке 6.1) и одноконтурной (рисунок 7.1) системы управления, с последующей оценкой их эффективности для двух видов задающих сигналов (ступенчатой и линейной форм). Также необходимо исследовать влияние наличия корректирующего устройства в системе управления на точность отработки задающего воздействия (математические модели показаны на рисунках 7.2 и 7.3). Варианты параметров математических моделей указаны в таблице 7.1.

В работе следует:

1) модифицировать динамическую модель, полученную в лабораторной работе № 6, для исследования варианта следящей ЭМС с общим цифровым ПИД-регулятором для обратных связей по скорости и положению в среде MATLAB/Simulink;

2) выполнить оптимальную настройку параметров общего ПИД-регулятора скорости и положения при ступенчатом задающем воздействии, сохранив параметры общего регулятора;

3) исследовать режим отработки ступенчатого воздействия с пуском на холостом ходу и набросом возмущающего воздействия, имеющего реактивный характер и изменяющегося скачком от нуля до заданного значения M_c ,

зафиксировав в качестве результатов динамические характеристики положения, скорости, момента на валу АД, величины напряжения управления;

4) сравнить полученные данные с результатами выполнения лабораторной работы № 6 и сделать выводы о том, какая из исследованных систем является оптимальной для режима позиционирования;

5) совместить две рассмотренные ранее модели в одном файле среды Simulink и изменить форму задающего воздействия на линейно возрастающий сигнал согласно данным таблицы 7.1;

6) выполнить расчет единой модели, зафиксировав на одном графике задающее воздействие, а также его обработку системой подчиненного регулирования и ЭМС с общим регулятором. По результатам вычислительного эксперимента следует сделать выводы об эффективности каждого из двух вариантов реализации ЭСМ;

7) добавить в систему управления каждой ЭМС корректирующее устройство (КУ) с дифференцирующим звеном с передаточной функцией

$$W_{\text{ку}}(z) = \frac{K_{\text{ку}}(z-1)}{z}; \quad (7.1)$$

8) выполнить вычислительный эксперимент, по результатам которого выбрать оптимальный вариант реализации следящей системы для линейного варианта задающего воздействия.

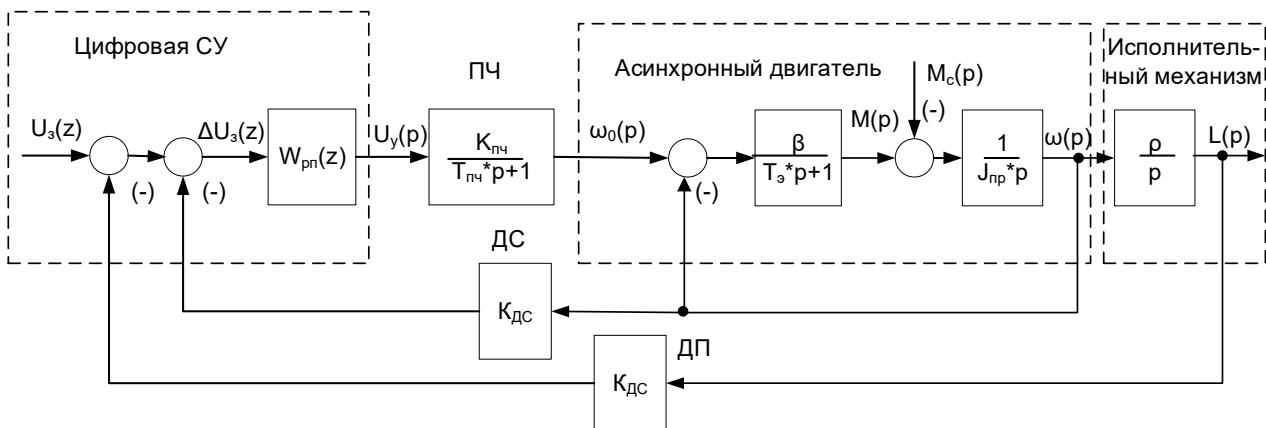


Рисунок 7.1 – Структурная схема математической модели ЭМС с общим регулятором

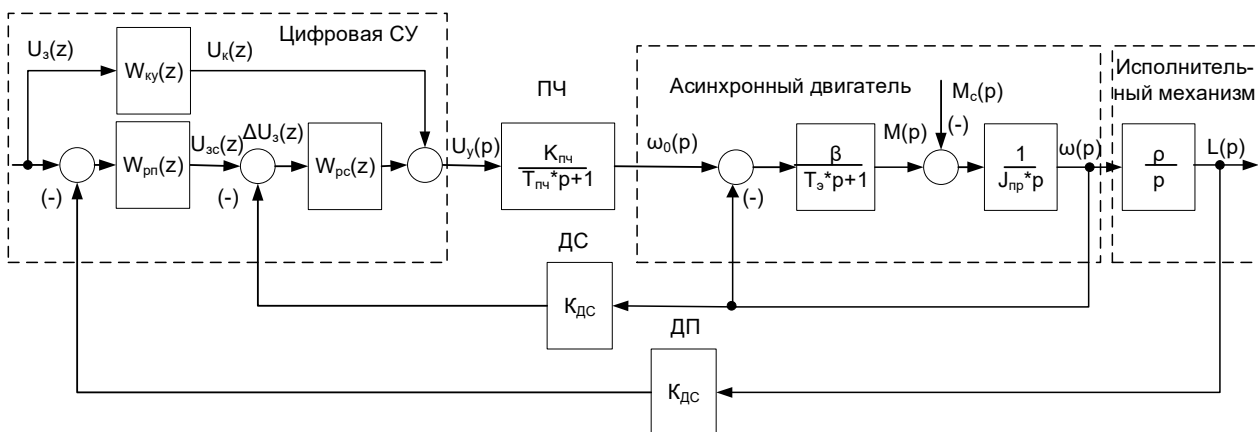


Рисунок 7.2 – Схема двухконтурной следящей ЭМС с корректирующим устройством

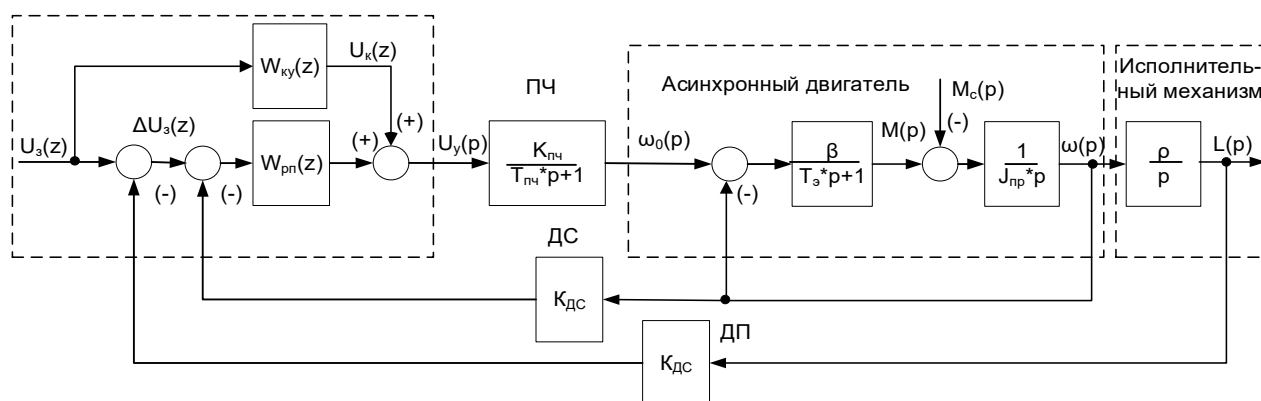


Рисунок 7.3 – Схема следящей ЭМС с общим регулятором скорости и положения с КУ

Таблица 7.1 – Варианты индивидуальных заданий к лабораторной работе № 7

Вариант	1	2	3	4	5	6	7	8	9	10
$U_3, В$	8	8	8	8	9	8	8	8	8	8
$U_{у\max}, В$	10	10	10	10	10	10	10	10	10	10
$J_{пр}, кг \cdot м^2$	0,04	0,05	0,055	0,06	0,08	0,1	0,12	0,14	0,18	0,34
$K_{пч}$	32	32	32	32	32	32	32	32	32	32
$T_{пч}, с$	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01
$K_{дс}, В \cdot с$	0,0318	0,0318	0,0318	0,0318	0,0318	0,0318	0,0318	0,0318	0,0318	0,0318
$K_{дп}, В/м$	1	1	1	1	1	1	1	1	1	1
$\beta, Н \cdot м \cdot с$	0,240	0,452	0,632	0,837	1,468	1,463	5,642	5,643	8,663	10,52
$T_э, с$	0,092	0,123	0,141	0,142	0,246	0,253	0,381	0,419	0,353	0,414
$M_c, Н \cdot м$	3	4	5	8	10	12	15	24	35	42
$\rho, м$	0,02	0,025	0,04	0,05	0,05	0,1	0,08	0,04	0,05	0,05
$K_{дп}$	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
$K_{ку}$	1000	1000	1000	1000	1000	1000	1500	1500	1500	1500
$t_{сч}, с$	6	6	6	6	6	6	6	6	6	6
$t_{н}, с$	2,5	3	3	3	3	3	3	3	3	3

Обозначения на структурных схемах рисунков 7.1–7.3 соответствуют описаниям подраздела 6.2 за исключением:

- $K_{ку}$ – коэффициент усиления корректирующего устройства;
- $W_{ку}(z)$ – передаточная функция корректирующего устройства системы управления;
- $U_{ку}(z)$ – дискретный сигнал управления на выходе корректирующего устройства.

7.3 Ход выполнения работы № 7

7.3.1 Загрузка среды MATLAB/Simulink.

Сначала запускается среда MATLAB/Simulink согласно методике п. 6.3.1. Не следует непосредственно работать с мобильным носителем данных (флешкой) в среде MATLAB.

7.3.2 Создание файла Simulink-модели.

Создание S-модели исследуемой ЭМС выполняется на основе файла, разработанного при выполнении лабораторной работы № 6. С помощью раздела *File/Open* или кнопки *Open* в среде *Simulink* открывается файл модели из работы № 6. Необходимо сразу сохранить открытую модель под новым именем с помощью раздела *File/Save* и изменить справочную информацию по ней.

Также требуется открыть и сохранить под новым именем в среде MATLAB m-файл с исходными данными к лабораторной работе № 6.

7.3.3 Формирование модели следящей ЭМС с общим регулятором.

S-модель системы с общим регулятором получается удалением регулятора положения из предыдущей модели и соединением выхода первого сумматора с положительным входом второго. Пример полученной S-модели показан на рисунке 7.4.

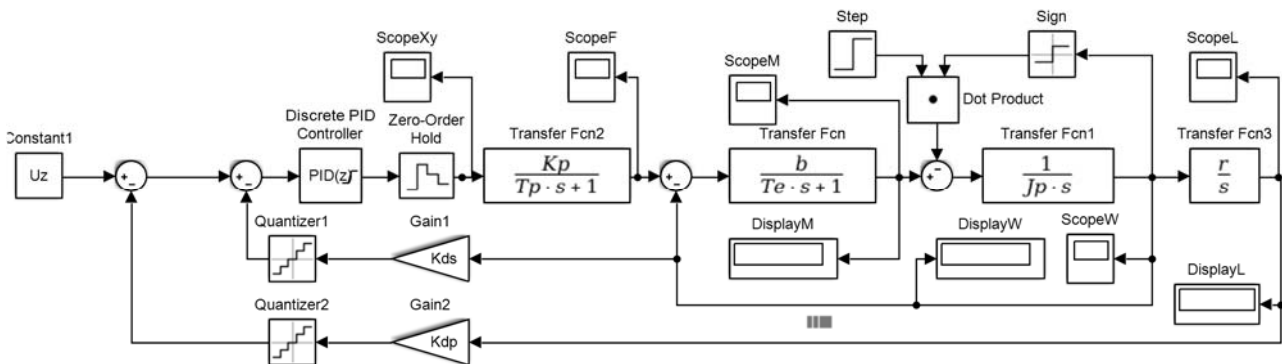


Рисунок 7.4 – S-модель позиционной ЭМС с общим регулятором

7.3.4 Оптимизация общего регулятора.

Перед настройкой регулятора обязательно выполняется в среде MATLAB m-файл исходных данных нажатием клавиши *F5* или кнопки *Run*.

Процедура оптимальной настройка идентична пп. 6.3.6 и 6.3.7 из лабораторной работы № 6. Следует добиться такого быстродействия системы, при котором система обрабатывала бы заданное положение за время, меньшее указанного в значении t_n из таблицы 7.1. Величина перерегулирования динамической характеристики положения при этом не должна превышать 20 %, а число колебаний за время регулирования – быть не более пяти.

7.3.5 Проведение вычислительного эксперимента с моделью ЭМС.

Запуск процесса расчета S-модели выполняется кнопкой *Run* или клавишами *Ctrl+T*. При этом настройка параметров моделирования соответствует работе № 6 (см. п. 6.3.8). Полученные графики динамических характеристик момента, скорости и положения копируются в отчет с помощью *Edit/Copy Figure* с использованием векторного формата.

По завершении численного интегрирования выполняется оценка показателей качества регулирования положения (времени регулирования, перерегулирования, числа колебаний и максимальной динамической ошибки от наброса нагрузки) аналогично методике п. 6.3.9.

7.3.6 Составление модели для сравнения двух ЭМС.

Полученная в п. 7.3.4 модель сохраняется в среде Simulink под новым именем. Далее открывается модель, созданная в лабораторной работе № 6. С помощью мыши выделяются все блоки открытой модели и копируются через буфер обмена в файл S-модели ЭМС с общим регулятором. Вставленная модель располагается мышкой под имеющейся в файле.

Удаляются блоки *Constant*, моделирующие ступенчатое задающее воздействие. Для моделирования линейного задающего воздействия используется соединение блока *Ramp* (из *Sources*) и блок *Saturation* (из *Discontinuous*). В блоке *Ramp* в поле параметра *Slope* указывается наименование глобальной переменной U_3 , деленное на 4. Для блока *Saturation* устанавливается в поле *Upper limit* наименование глобальной переменной U_{\max} , а в поле *Lower limit* – нулевое значение. Выход с блока *Saturation* подается на оба входа моделей ЭМС.

Для визуальной фиксации отличий в воспроизведении задающего воздействия устанавливается дополнительный блок *Scope* (из *Sinks*), у которого в параметре *Number of input ports* указывается три входа. На первый вход заводится сигнал линейно изменяющегося задания с выхода *Saturation*, на второй – сигнал положения с выхода ЭМС подчиненного регулирования, на третий – положение с выхода ЭМС с общим регулятором.

Для выбора наиболее эффективной структуры следящей системы необходимо выполнить расчет суммарной ошибки каждой ЭМС по отработке положения. Так как ошибка отработки положения может иметь различный знак, то для предотвращения ее уменьшения при суммировании следует предварительно возвести сигнал в квадрат с помощью блока *Math Function* (из *Math Operation*), параметр *Function* следует установить в значение *magnitude^2*. Для суммирования дискретных сигналов используется блок *Discrete-Time Integrator* (из *Discrete*). Для отображения полученного значения суммарной ошибки отслеживания положения используется блок *Display* (из *Sinks*). Полученная для проведения исследования модель примет вид, показанный на рисунке 7.5.

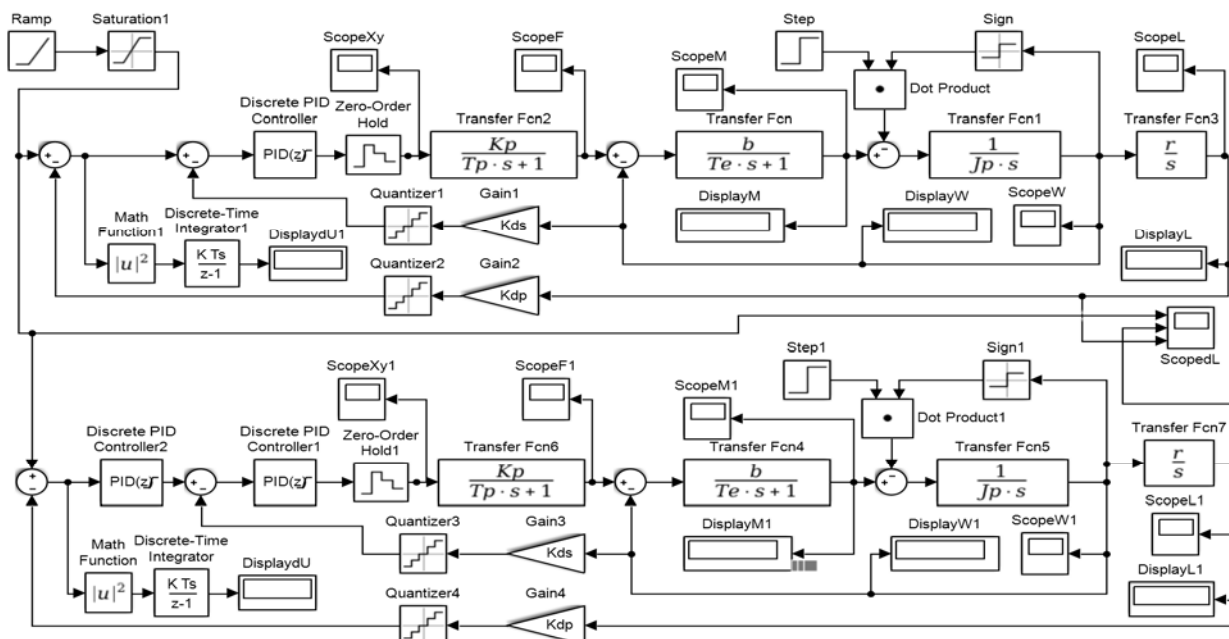


Рисунок 7.5 – Пример модели для сравнения эффективности двух вариантов ЭМС

Рекомендуется сохранить вариант перед дальнейшим использованием.

7.3.7 Проведение эксперимента и анализ данных S-модели двух ЭМС.

Не изменяя параметров настройки численного интегрирования, установленных в п. 6.3.8, следует запустить вычислительный процесс кнопкой *Run* или клавишами *Ctrl+T*.

После завершения расчетов фиксируются графики с общими динамическими характеристиками отработки положения, а также отдельные характеристики изменения момента и скорости для каждой ЭМС. Необходимо также зафиксировать данные суммарных квадратичных ошибок для каждой из ЭМС. Полученные данные копируются в файл отчета и на их основании делается вывод об эффективности рассматриваемых систем.

7.3.8 Добавление корректирующих устройств в модель ЭМС.

Перед добавлением в модели ЭМС корректирующих устройств следует сохранить S-модель в новом файле, чтобы оставить неизменным разработанный ранее вариант модели.

В новом файле модели параллельно основным регуляторам систем управления ЭМС размещается последовательное соединение блоков *Gain* (из *Math Operation*) и *Difference* (из *Discrete*). В качестве параметра *Gain* указывается значение $K_{ку}$ из таблицы 7.1. На вход корректирующего устройства подается линейное задающее воздействие. Выход корректирующих устройств суммируется с помощью блока *Sum* (из *Math Operation*) с сигналов выхода остальной части системы управления. На выходе этого сумматора устанавливается ограничитель в виде блока *Saturation* (из *Discontinues*). В параметрах в *Upper limit* указывается глобальная переменная с данными $U_{уmax}$, а в *Lower limit* задается отрицательное значение глобальной переменной с данными $U_{уmax}$. Пример модели ЭМС с корректирующими устройствами показан на рисунке 7.6. Следует выполнить сохранение разработанной модели перед проведением дальнейших исследований.

В m-файл исходных данных модели следует добавить строку с присвоением глобальной переменной коэффициента усиления корректирующего устройства $K_{ку}$ значения из таблицы 7.1.

7.3.9 Эксперимент и анализ данных S-модели ЭМС с КУ.

Перед выполнением эксперимента с S-моделью следует запустить на выполнение в среде MATLAB весь m-файл клавишей *F5*. Если заново выполнять по каким-либо причинам весь файл нежелательно, то возможно выполнить только добавленную в п. 7.3.8 строку данных, предварительно выделив ее и нажав клавишу *F9*.

Для проведения эксперимента не нужно изменять параметры численного интегрирования, заданные при выполнении в п. 7.3.7. Исключение возможно только в изменении величины шага интегрирования с целью увеличения точности расчетов. Запуск вычислительного процесса выполняется кнопкой *Run* или клавишами *Ctrl+T*.

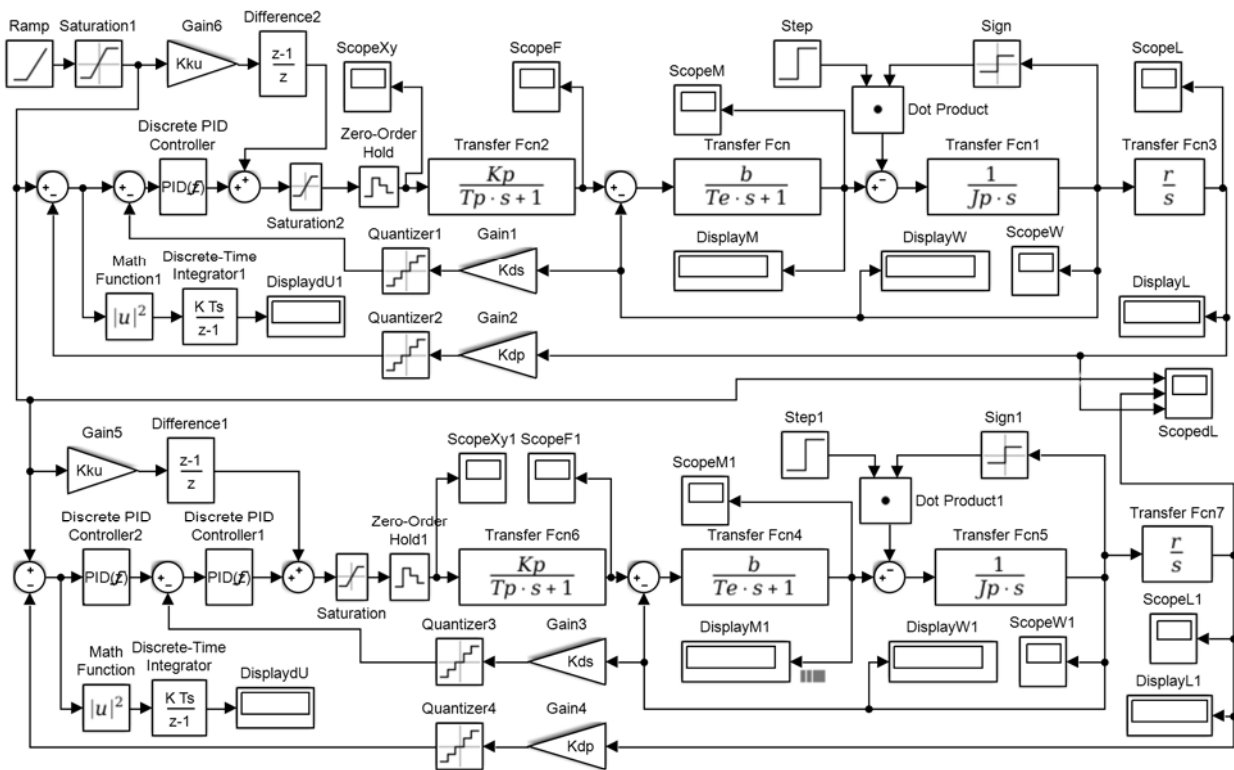


Рисунок 7.6 – Модель сравнения следящих ЭМС с корректирующими устройствами

По завершении расчетов фиксируются графики общих динамических характеристик регулирования положения, отдельные характеристики изменения момента и скорости для каждой ЭМС. Следует зафиксировать данные суммарных квадратичных ошибок для каждого из вариантов ЭМС. Данные расчетов копируются в файл отчета и делается вывод об эффективности вариантов рассматриваемых ЭМС.

7.3.10 Окончание работы со средой MATLAB и компьютером.

Перед закрытием среды MATLAB необходимо сохранить всех открытые файлы моделей и данных. Рекомендуется выполнить резервное копирование файлов моделей и электронного отчета на мобильный носитель или в личный аккаунт компьютерной сети. Также нужно удалить все ненужные временные файлы.

7.4 Содержание отчета по работе № 7

Отчеты по лабораторной работе № 7 оформляются индивидуально на листах формата А4 на бумажном или электронном носителе и имеют описанный ниже состав.

7.4.1 Титульный лист отчета.

7.4.2 Текст индивидуального задания.

7.4.3 Изображение S-модели двух вариантов реализации следящих ЭМС с корректирующими устройствами.

7.4.4 Динамические характеристики положения, скорости, момента ЭМС с общим регулятором положения и скорости при отработке ступенчатого задающего воздействия.

7.4.5 Показатели качества регулирования положения при отработке ступенчатого воздействия: время регулирования, величина перерегулирования, число колебаний за время регулирования, максимальная динамическая ошибка при набросе реактивной нагрузки.

7.4.6 Совместный график динамических характеристик отработки положения двух вариантов ЭМС и задающего воздействия при отработке линейного задающего воздействия без корректирующего устройства.

7.4.7 Выводы по эффективности структур ЭМС без корректирующего устройства, сделанные на основе анализа динамических характеристик и данных суммарных квадратичных ошибок.

7.4.8 Совместный график динамических характеристик отработки положения двух вариантов ЭМС и задающего воздействия при отработке линейного задающего воздействия с корректирующим устройством.

7.4.9 Выводы по эффективности структур ЭМС с корректирующими устройствами, сделанные на основе анализа динамических характеристик и данных суммарных квадратичных ошибок.

Контрольные вопросы по лабораторной работе № 7

- 1 Дать определение понятия «структурная оптимизация».
- 2 Дать определение понятия «целевая функция».
- 3 Какие критерии эффективности используются при структурной оптимизации ЭМС?
- 4 Дать определение понятий «минимизация» и «максимизация».
- 5 Какие программные ресурсы MATLAB могут использоваться для оптимизации электротехнических систем?
- 6 Как в среде Simulink сформировать модель корректирующего устройства для следящей ЭМС?
- 7 Каким образом в среде Simulink формируется задающее воздействие?
- 8 Каким образом в среде Simulink выполняется сравнение эффективности отслеживания задающего воздействия каждого из вариантов ЭМС?
- 9 Каким образом в среде Simulink можно выполнить расчет суммарного квадратичного значения контролируемого сигнала за время моделирования?
- 10 Какие группы численных методов используются для поиска экстремума целевой функции?
- 11 Какие задачи решаются при структурной оптимизации следящих ЭМС?
- 12 Какое программное обеспечение может использоваться для оптимизации систем электроэнергетики?
- 13 В чем заключаются преимущества и недостатки размещения двух вариантов построения Simulink-моделей электротехнической системы в одном файле?

14 Какие виды ограничений налаживаются на функцию качества при оптимизации ЭМС?

15 Какой вид имеет общий алгоритм при выполнении структурной оптимизации следящих ЭМС?

16 Какие виды систем управления в электроэнергетике основываются на структурной оптимизации в процессе эксплуатации установки?

Дополнительно при защите отчета можно использовать контрольные вопросы из лабораторных работ № 5: 3, 6–13 и из № 6: 3–8.

8 Лабораторная работа № 8. Моделирование нейронных сетей в электроэнергетике

8.1 Цель лабораторной работы № 8

1 Усвоение терминологии в области интеллектуальных систем управления.

2 Изучение общих принципов построения систем управления электроэнергетическими системами на базе нейронных сетей.

3 Изучение методов составления и исследования нейронных сетей в среде MATLAB.

4 Получение практических навыков по работе с моделями нейронных сетей в среде MATLAB/Simulink/Neural Networks Toolbox.

5 Получение практических навыков по проведению вычислительного эксперимента и обработки данных моделей нейронными сетями.

8.2 Задание к лабораторной работе № 8

Заданием является разработка в среде MATLAB/Simulink и Neural Networks Tools нейронных сетей для анализа табличных данных.

Используя массивы данных X и Y из таблицы 8.1, следует:

1) разработать в среде обобщенно-регрессионную нейронную сеть *GRNN* (*generalized regression neural networks*) для аппроксимации таблично заданной функции с абсолютной точностью 0,01;

2) провести вычислительный эксперимент с моделью нейронной сети *GRNN* для интерполяции данных на заданном интервале с шагом 0,02 и оценить погрешность результата;

3) создать в среде MATLAB модель нейронной сети с радиальными базисными элементами *RNB* (*radial basis network*) для аппроксимации заданной функции;

4) выполнить эксперимент с полученной моделью нейронной сети *RNB* для интерполяции данных на заданном интервале с шагом 0,02 и оценить погрешность результата;

5) с помощью программного инструментального пакета *Neural Network tool (NNT)* в среде MATLAB создать модель двухслойной нейронной сети с прямой передачей сигнала и обратным распространением ошибки (*feed-forward backprop*), выполнив ее обучение, передать полученную модель в рабочую область MATLAB;

6) выполнить эксперимент с созданной *NNT*-моделью нейронной сети по интерполяции данных на заданном интервале с шагом 0,02 с последующей оценкой погрешности результата;

7) передать разработанную модель в среду Simulink.

Таблица 8.1 – Варианты индивидуальных заданий к лабораторной работе № 8

Вариант	Табличная функция исходных данных								
	X	0	1	2	3	4	5	6	7
1	X	0	1	2	3	4	5	6	7
	Y	1	2,5	3	5	5,5	7	8	9
2	X	0	1	2	3	4	5	6	8
	Y	-1	-0,5	1	5	6,5	9	11,5	15
3	X	0	1	2	3	4	5	7	9
	Y	0,5	1	2,5	5	8,5	9,5	15	17,5
4	X	0	2	4	6	8	10	12	14
	Y	1	4	9	13	15	21	23	29
5	X	0	3	6	9	12	15	18	21
	Y	-2	8	16	25	35	45	52	62
6	X	0	2	4	6	10	16	20	25
	Y	-22	-14	-2	6	15	25	35	50
7	X	0	1	2	3	4	5	6	7
	Y	2	4	5	6	6,5	7	7,5	8
8	X	0	1	2	3	4	5	6	7
	Y	1	2,5	7	14,5	25	38,5	55	74,5
9	X	0	1	2	3	4	5	6	7
	Y	-1	0	3	8	15	24	35	48
10	X	0	1	2	3	4	5	6	7
	Y	-2	0	0,828	1,464	2	2,472	2,9	3,292

8.3 Ход выполнения работы № 8

8.3.1 Загрузка среды MATLAB.

Первоначально выполняется запуск среды MATLAB. Не рекомендуется выполнять на компьютере какие-либо операции пока не раскроется главное окно среды MATLAB. Сразу следует перейти в свой рабочий каталог (папку).

8.3.2 Создание *m*-файла расчетов лабораторной работы.

С помощью кнопки *New Script...* или клавиш *Ctrl+N* в главном окне MATLAB создается новый файл для формирования данных и команд в лабораторной работе. Следует сразу сохранить файл в своем персональном рабочем каталоге с уникальным именем.

8.3.3 Формирование исходных данных.

В начале m-файла следует очистить командное окно MATLAB и удалить данные и графики предыдущих расчетов с помощью команд *clc*, *clear all* и *close all*.

Исходные данные таблицы 8.1 помещаются с помощью операторов присваивания в два одномерных массива. Элементы массива указываются через пробел внутри квадратных скобок. Например:

```
x=[-1 -0.8 -0.6 -0.4 -0.2 0 0.2 0.4 0.6 0.8 1] % переменная X
y=[1 0.64 0.36 0.16 0.04 0 0.04 0.16 0.36 0.64 1] % массив Y
```

8.3.4 Создание модели нейронной сети GRNN.

Модель сети GRNN создается в m-файле вызовом функции *newgrnn* в формате

```
имя_модели = newgrnn (переменная_x, переменная_y, погрешность)
```

Пример создания модели сети GRNN с погрешностью 0,01:

```
modA=newgrnn(x, y, 0.01) % создание сети GRNN
```

8.3.5 Интерполяция данных на заданном интервале моделью сети GRNN.

Для проведения эксперимента следует в m-файле объявить переменную входных сигналов сети в следующем формате:

```
имя_переменной = первое_значение_X : шаг : последнее_значение_X
```

Пример описания

```
x2=-1:0.02:1 % создание переменной входа для интерполяции
```

Вычисление выходных данных с помощью нейронной сети выполняется командой *sim* в формате

```
имя_выходных_данных = sim (имя_модели, имя_переменной_входов)
```

Пример записи в m-файле:

```
y2=sim(modA, x2) % расчет данных с помощью модели сети
```

Для визуального анализа данных строятся графики исходных данных и результатов расчета в декартовой системе координат командой *plot*.

Пример:

```
plot(x, y, x2, y2) % построение графика
grid on           % включение отображения координатной сетки
```

Расчет m-файла запускается нажатием клавиши F5 или кнопки *Run*.

Полученный рисунок копируется в файл отчета с помощью пункта меню графика *Edit/Copy Figure*.

8.3.6 Создание модели нейронной сети RBE.

Для формирования модели сети RBE в составе m-файла вызывается функция *newrbe* в следующем формате:

```
имя_модели = newrbe (переменная_x, переменная_y)
```

Пример создания модели сети RBE:

```
modB=newrbe(x, y) % создание модели сети RBE
```

8.3.7 Интерполяция данных на заданном интервале моделью сети RBE.

Интерполяция данных, сформированных в п. 8.3.5, с помощью созданной нейронной сети *RBE* выполняется добавлением команды *sim* в *m*-файл. Пример:

```
y3=sim(modB,x2) % расчет данных с помощью модели сети RBE
```

Визуализация исходных данных и результатов расчета выполняется командой *plot*. Предварительно следует создать новое графическое окно командой *figure*. Пример:

```
figure % открытие нового окна построения графика
plot(x,y,x2,y3) % построение графика
grid on % включение отображения координатной сетки
```

Чтобы не запускать повторно расчет всего *m*-файла рекомендуется выделить требуемые строки и выполнить описанные в них команды нажатием клавиши F9. Полученный рисунок графиков копируется в файл отчета.

8.3.8 Создание модели сети с помощью *Neural Network Tools*.

В командном окне среды MATLAB с помощью функции *nntool* вызывается окно инструментальной среды моделирования нейронных сетей. Пример окна показан на рисунке 8.1.

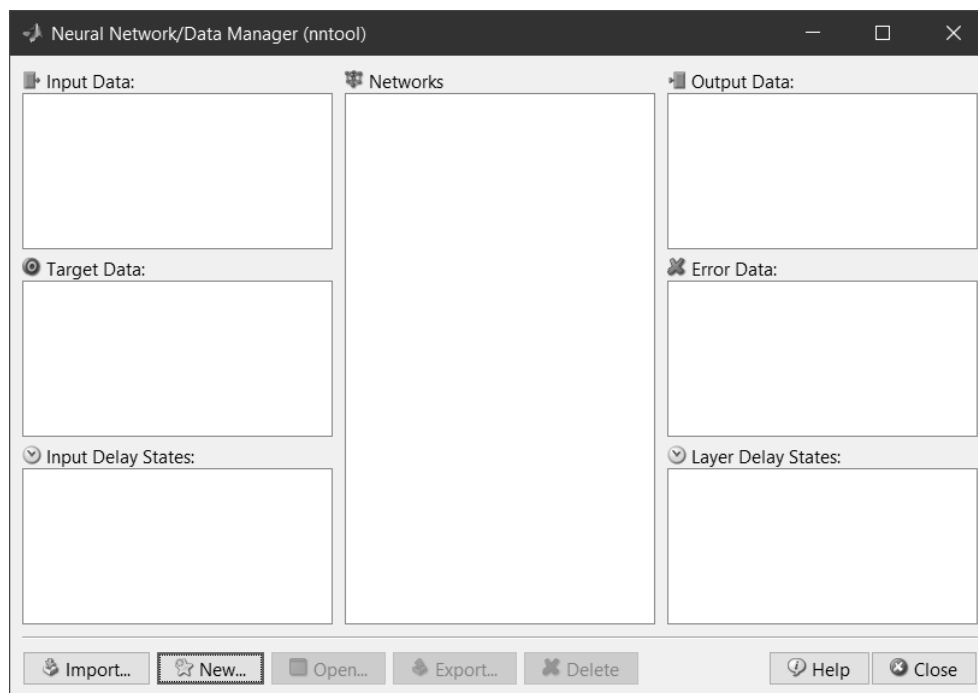


Рисунок 8.1 – Вид нового окна инструментального пакета *NNTool*

Сначала необходимо выполнить загрузку (импорт) исходных данных из среды MATLAB. Для этого следует нажать кнопку *Import*, что вызовет открытие окна *Data Manager*, показанного на рисунке 8.2.

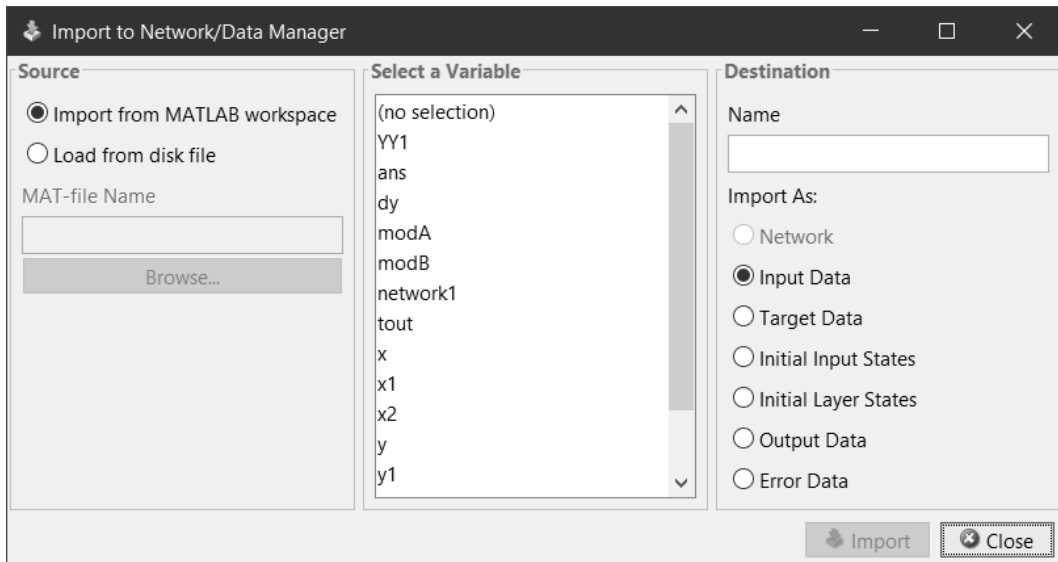


Рисунок 8.2 – Окно менеджера данных пакета *NNTool*

Для загрузки исходных данных переменной X в списке *Import As* устанавливается маркер в позицию *Input data*, а в окне *Select a variable* указывается имя глобальной переменной исходных данных X . Ввод данных выполняется кнопкой *Import*. Данные зависимой переменной Y в модель нейронной сети загружаются аналогично с выбором позиции *Target data* и указанием соответствующего имени. Окончание работы с окном выполняется нажатием кнопки *Close*.

Создание модели нейронной сети в *NNTool* выполняется нажатием кнопки *New*, что раскроет окно настройки параметров сети, показанное на рисунке 8.3.

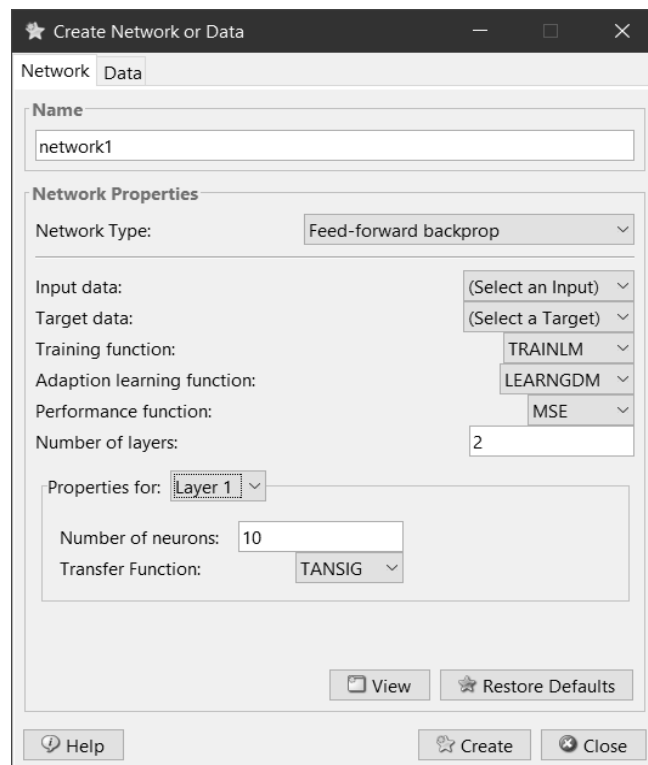


Рисунок 8.3 – Окно настройки параметров создаваемой нейронной сети

В поле *Name* записывается имя создаваемой нейронной сети.

В выпадающем списке *Network Type* указывается тип создаваемой нейронной сети с прямой передачей сигнала и обратным распространением ошибки (*feed-forward backprop*). В поле *Number of neurons* задается число слоев сети равное двум.

В списке *Input data* указывается имя переменной массива X , в *Target Data* – имя переменной массива Y с исходными данными из таблицы 8.1.

Просмотреть структурную схему выбранной нейронной сети можно нажатием кнопки *View*. При этом откроется окно, показанное на рисунке 8.4, которое рекомендуется скопировать в файл отчета.

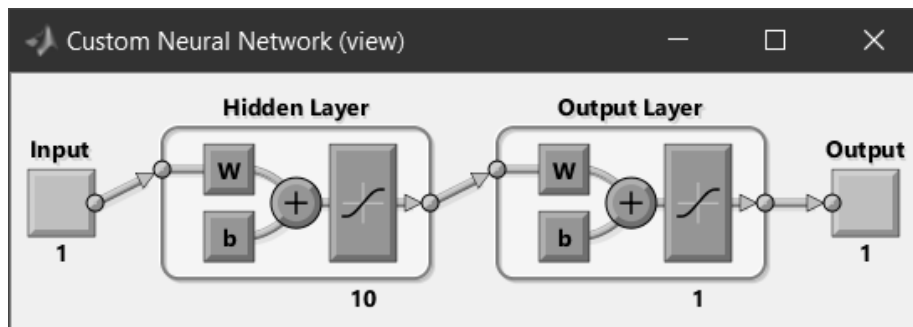


Рисунок 8.4 – Окно с изображением структуры нейронной сети

Завершение формирования модели сети выполняется нажатием кнопки *Create*.

8.3.9 Обучение нейронной сети.

После формирования модели нейронной сети следует выполнить ее обучение. Для этого в окне *NNTool* в поле *Networks* выполняется двойной клик мышкой по имени сформированной сети. Это распахнет окно сети с ее схемой и системой вкладок, показанное на рисунке 8.5.

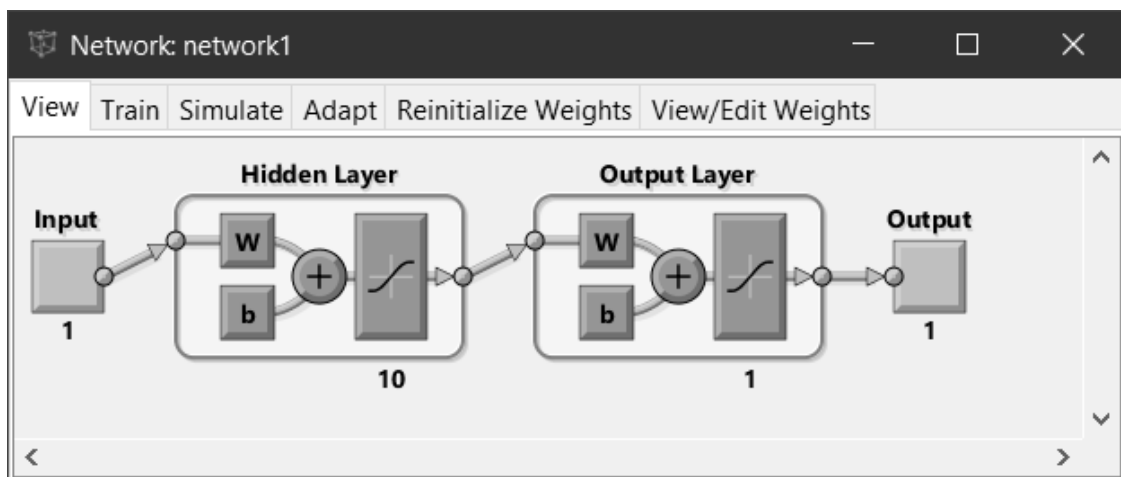


Рисунок 8.5 – Окно работы с моделью нейронной сети

Обучение сети выполняется с помощью вкладки меню *Train*. Это раскрывает на экране окно, показанное на рисунке 8.6.

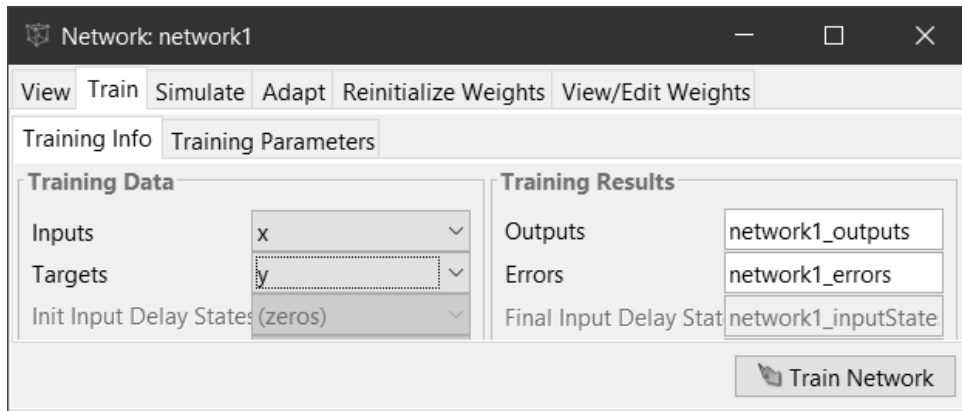


Рисунок 8.6 – Окно настройки обучения нейронной сети

В списке *Inputs* указывается имя переменной входных данных, в *Targets* – имя выходных данных. Процесс обучения запускается нажатием кнопки *Train Network*. По окончании обучения распахнется окно, показанное на рисунке 8.7.

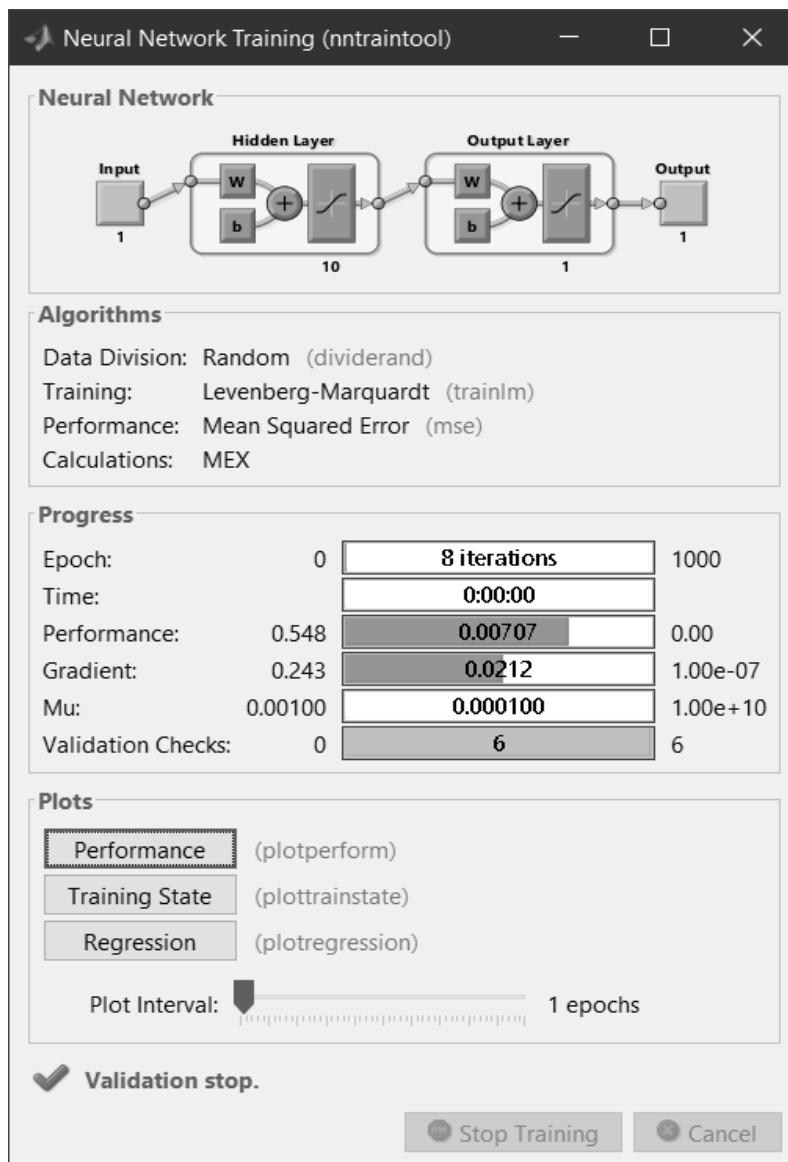


Рисунок 8.7 – Окно результатов обучения нейронной сети

Окно с результатами обучения копируется в файл отчета. По согласованию с преподавателем выполняется просмотр и копирование в отчет графиков результатов обучения из раздела *Plots*.

8.3.10 Экспорт созданной модели нейронной сети в MATLAB.

Экспорт созданной модели обратно в среду MATLAB выполняется в окне *NNTools* нажатием кнопки *Export*. После чего распахнется окно экспорта данных, изображенное на рисунке 8.8. В поле *Select Variables* указывается имя модели нейронной сети и нажимается кнопка *Export*.

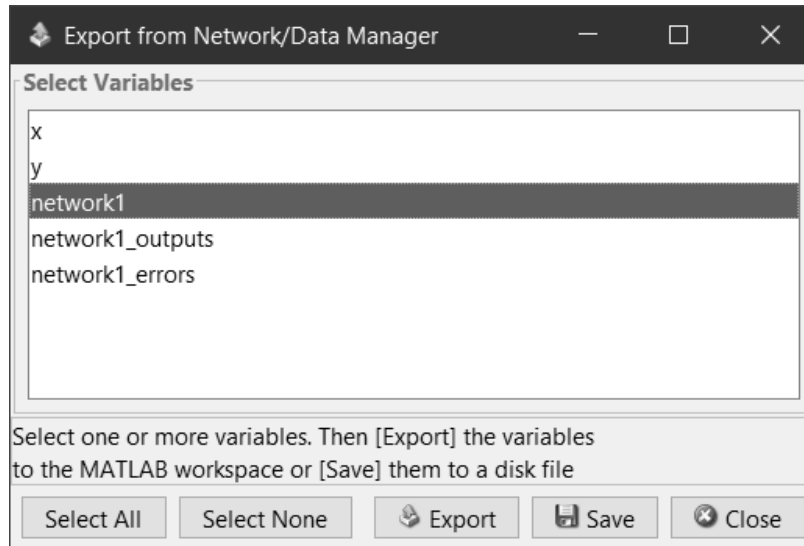


Рисунок 8.8 – Окно экспорта данных из *NNTool*

8.3.11 Эксперимент с моделью нейронной сети в среде MATLAB.

Для выполнения интерполяции в *m*-файл дописывается строка с вызовом функции *sim* и построение графиков аналогично п. 8.3.8. При этом меняется имя модели сети и имя результатов расчета. Пример:

```
y4=sim(network1,x2) % расчет данных с помощью модели сети
figure % открытие нового окна построения графика
plot(x,y,x2,y4) % построение графика
grid on % включение отображения координатной сетки
```

Полученный график копируется в файл отчета.

8.3.12 Экспорт модели нейронной сети в среду Simulink.

Для передачи модели нейронной системы в среду визуального моделирования Simulink в командном окне выполняется команда *genism* в формате: *genism(имя_модели, -1)*. После успешного выполнения команды раскроется окно Simulink-модели с переданной нейронной сетью, показанное на рисунке 8.9.

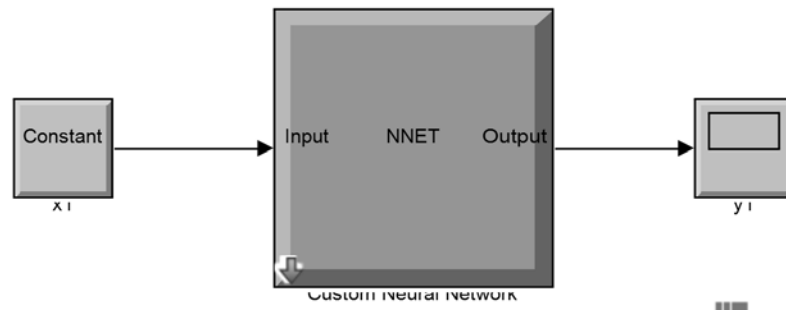


Рисунок 8.9 – Вид экспорта модели нейронной сети в Simulink

Если выполнить клик по изображению модели сети, то раскроется схема нейронной сети, пример которой демонстрируется на рисунке 8.10.

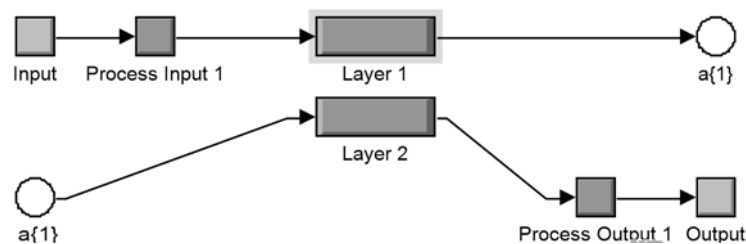


Рисунок 8.10 – Схема структуры нейронной сети в Simulink

При клике мышью по блоку *Layer 1* раскрывается схема модели первого слоя сети, показанная на рисунке 8.11, а по блоку *Layer 2* – схема второго слоя, показанного на рисунке 8.12.

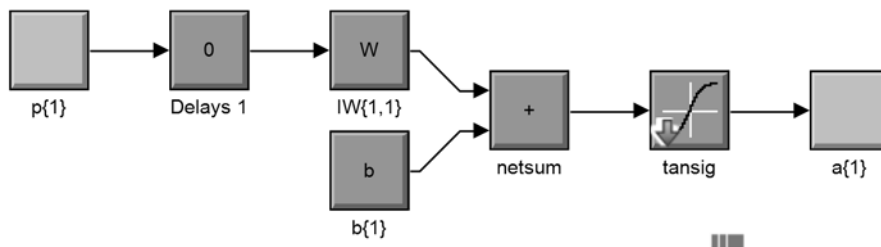


Рисунок 8.11 – Схема первого слоя нейронной сети

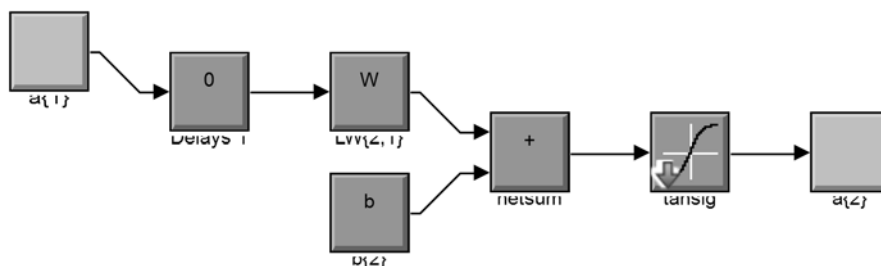


Рисунок 8.12 – Схема второго слоя нейронной сети

Указанные рисунки копируются в файл отчета.

8.3.13 Формирование отчета по результатам эксперимента с моделью.

Отчет оформляется индивидуально на электронном или бумажном носителе.

8.3.14 Окончание работы со средой MATLAB и ПК.

Перед выходом из среды MATLAB выполняется сохранение файла модели. После выхода из среды MATLAB следует выполнить резервное копирование всех файлов модели нейронных сетей и электронного отчета на мобильный носитель данных или сетевой диск.

8.4 Содержание отчета по работе № 8

Отчеты по лабораторной работе № 8 оформляются индивидуально на листах формата А4 в соответствии с требованиями ГОСТ 7.32 на бумажном или электронном носителе в соответствии с приведенным ниже составе.

8.4.1 Титульный лист отчета.

8.4.2 Текст индивидуального задания.

8.4.3 Содержание m-файла с программируемыми расчетами.

8.4.4 Копия данных результатов расчета моделей из командного окна MATLAB.

8.4.5 Графики с результатами интерполяции данных моделью *GRNN*.

8.4.6 Графики с результатами интерполяции данных моделью *RBE*.

8.4.7 Графики с результатами интерполяции данных моделью нейронной сети с прямой передачей сигнала и обратным распространением ошибки.

8.4.8 Копия экрана результатов обучения нейронной сети с прямой передачей сигнала и обратным распространением ошибки.

8.4.9 Изображение Simulink-модели с нейронной сетью с прямой передачей сигнала и обратным распространением ошибки.

8.4.10 Схема Simulink-подмодели с нейронной сетью с прямой передачей сигнала и обратным распространением ошибки.

8.4.11 Схема реализации в Simulink первого слоя с нейронной сети с прямой передачей сигнала и обратным распространением ошибки.

8.4.12 Схема реализации в Simulink второго слоя с нейронной сети с прямой передачей сигнала и обратным распространением ошибки.

Контрольные вопросы по лабораторной работе № 8

1 Дать определение понятия «искусственный интеллект».

2 Дать определения понятия «нейрон».

3 Дать определение понятия «нейронная сеть».

4 Дать определение понятия «интеллектуальная система управления».

5 Какие виды нейронных сетей используются при построении интеллектуальных систем управления?

- 6 На какие виды подразделяются нейронные сети прямого действия?
- 7 Каким образом классифицируются рекуррентные нейронные сети?
- 8 Какие этапы выделяют при создании нейронной сети?
- 9 Для каких целей выполняется обучение нейронных сетей?
- 10 Какой вид имеет общий алгоритм обучения нейронных сетей?
- 11 Какие ресурсы MATLAB используются для моделирования нейронных сетей?
- 12 Как создать модель нейронной сети в инструментальном пакете *NNTool*?
- 13 Каким образом выполняется обучение нейронных сетей в инструментальном пакете *NNTool*?
- 14 Как выполнить экспорт данных из *NNTool* в рабочее пространство MATLAB?
- 15 Каким образом выполняется экспорт моделей нейронных сетей из MATLAB в среду Simulink?
- 16 Для решения каких задач электроэнергетики используются нейронные сети?
- 17 Каким образом выполняется моделирование нейронной сети в среде Simulink?
- 18 В чем заключаются преимущества и недостатки нейронных сетей при анализе экспериментальных данных по сравнению с традиционными численными методами интерполяции и аппроксимации?
- 19 Каким образом выполняется оценка качества обучения нейронной сети?

Список литературы

- 1 **Щетинин, Ю. И.** Анализ и обработка сигналов в среде MATLAB / Ю. И. Щетинин. – Новосибирск: НГТУ, 2011. – 115 с.
- 2 **Плохотников, К. Э.** Вычислительные методы. Теория и практика в среде MATLAB. Курс лекций: учебное пособие для вузов / К. Э. Плохотников. – 2-е изд., испр. – Москва: Горячая линия-Телеком, 2013. – 496 с.
- 3 **Гилат, А.** MATLAB. Теория и практика: пер. с англ. / А. Гилат. – 5-е изд. – Москва: ДМК Пресс, 2017. – 416 с.
- 4 **Компьютерное моделирование: учебник / В. М. Градов [и др.].** – Москва: КУРС; ИНФРА-М, 2018. – 264 с.
- 5 **Моделирование систем управления с применением Matlab: учебное пособие / Под ред. А. Н. Тимохина.** – Москва: ИНФРА-М, 2019. – 256 с.
- 6 **Аксенов, М. И.** Моделирование электропривода: учебное пособие / М. И. Аксенов. – Москва: ИНФРА-М, 2019. – 135 с.
- 7 **Плохотников, К. Э.** Методы разработки математических моделей и вычислительный эксперимент на базе пакета Matlab. Курс лекций / К. Э. Плохотников. – Москва: СОЛОН-Пресс, 2017. – 628 с.
- 8 **Цифровая обработка сигналов и MATLAB / А. И. Солонина [и др.].** – Санкт-Петербург: БХВ-Петербург, 2013. – 512 с.