

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное
учреждение высшего образования
«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий и
управления в технических системах
Кафедра информатики и управления в технических системах

ИССЛЕДОВАНИЕ САУ СРЕДСТВАМИ ПАКЕТА SIMULINK

Методические указания
к выполнению лабораторной работы
по дисциплине "Теория автоматического управления"
для студентов направления подготовки
27.03.04 – "Управление в технических системах"



Севастополь
СевГУ
2020

УДК 681.5.012:004.942 (076.5)

ББК 32.965.4

И 87

Рецензент:

А.А. Кабанов – зав. кафедрой "Информатика и управление в технических системах", кандидат техн. наук, доцент

Составители: Карапетян В.А., Крамарь В.А.

И 87 Исследование САУ средствами пакета Simulink: методические указания к выполнению лабораторной работы по дисциплине "Теория автоматического управления" для студентов направления подготовки 27.03.04 – "Управление в технических системах" / Сост. В.А. Карапетян, В.А. Крамарь. – Севастополь: СевГУ, 2020. – 29 с.

В методических указаниях приводится описание лабораторной работы, целью которой является изучение и освоение основных возможностей пакета Simulink среды MATLAB для анализа систем автоматического управления.

Методические указания предназначены для студентов дневной и заочной форм обучения направления подготовки 27.03.04 – "Управление в технических системах".

УДК 681.5.012:004.9 (076.5)

Методические указания рассмотрены и утверждены на заседании кафедры "Информатика и управления в технических системах", протокол № 6 от 28 февраля 2020 г.

Методические указания рассмотрены и рекомендованы к изданию на заседании ученого совета Института информационных технологий и управления в технических системах, протокол № 4 от 3 марта 2020 г.

© ФГАОУ ВО «Севастопольский
государственный университет», 2020

СОДЕРЖАНИЕ

1 ЦЕЛЬ РАБОТЫ	4
2 ОСНОВНЫЕ СВЕДЕНИЯ О ПАКЕТЕ SIMULINK.....	4
2.1 Запуск Simulink.....	4
2.2 Соединение блоков.....	6
2.3 Установка параметров расчета и его выполнение.....	7
2.4 Библиотека блоков Simulink.....	7
2.4.1 Sources - источники сигналов	7
2.4.2 Sinks - приемники сигналов.....	9
2.4.3 Continuous – аналоговые блоки	11
2.4.4 Использование переменных для задания параметров блоков	14
2.4.5 Запись результатов расчета в текстовый файл	15
3 МАТЕМАТИЧЕСКИЕ МОДЕЛИ И СХЕМЫ SIMULINK	16
3.1 Математическая модель двигателя постоянного тока.....	16
3.2 Моделирование одноконтурной САУ ДПТ по скорости.....	24
3.3 Моделирование следящей САУ ДПТ по углу.....	25
4 ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ.....	26
5 СОДЕРЖАНИЕ ОТЧЕТА.....	28
6 КОНТРОЛЬНЫЕ ВОПРОСЫ.....	28
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	29

1 ЦЕЛЬ РАБОТЫ

Исследование методов и средств пакета имитационного моделирования Simulink для формирования, анализа и моделирования различных типов моделей линейных стационарных систем автоматического управления.

Работа состоит из двух частей по 4 академических часа каждая.

2 ОСНОВНЫЕ СВЕДЕНИЯ О ПАКЕТЕ SIMULINK

2.1 Запуск Simulink

Для запуска Simulink необходимо предварительно загрузить среду MATLAB. Если требуется начать новый Simulink-проект, то это можно сделать различными способами, например:

- нажать кнопку  на панели инструментов командного окна MATLAB;
- в командной строке окна MATLAB подать команду `simulink`;
- нажать кнопку  на вкладке Home главного окна MATLAB и выбрать из выпадающего списка пункт **Simulink model**.

После каждого из этих вариантов действий будет открыто промежуточное окно **Simulink Start Page**, в котором следует выбрать необходимый для дальнейшей работы проект (рисунок 2.1).

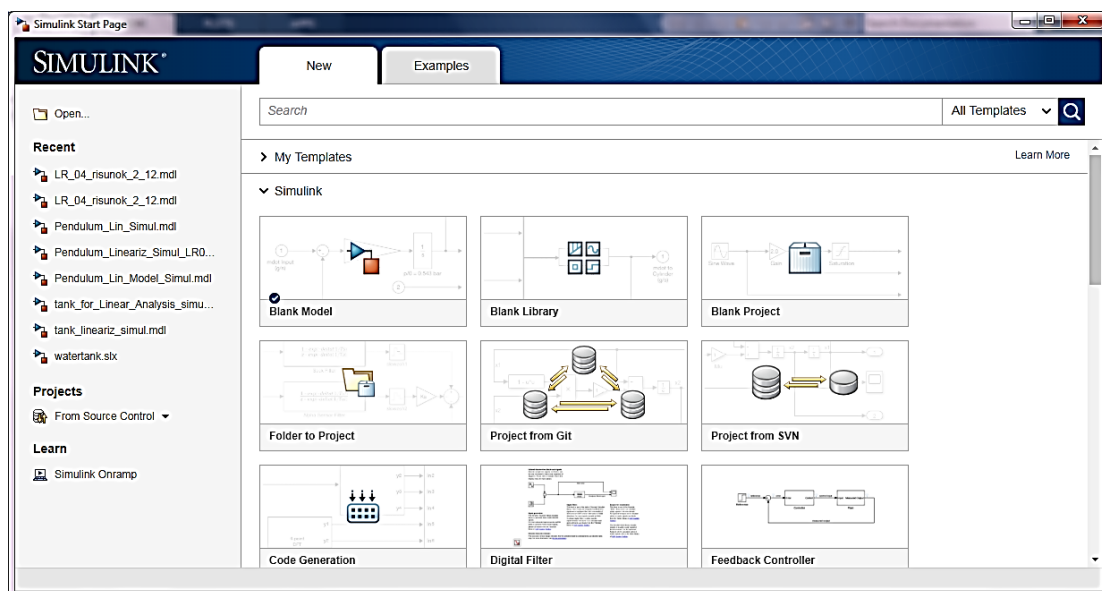
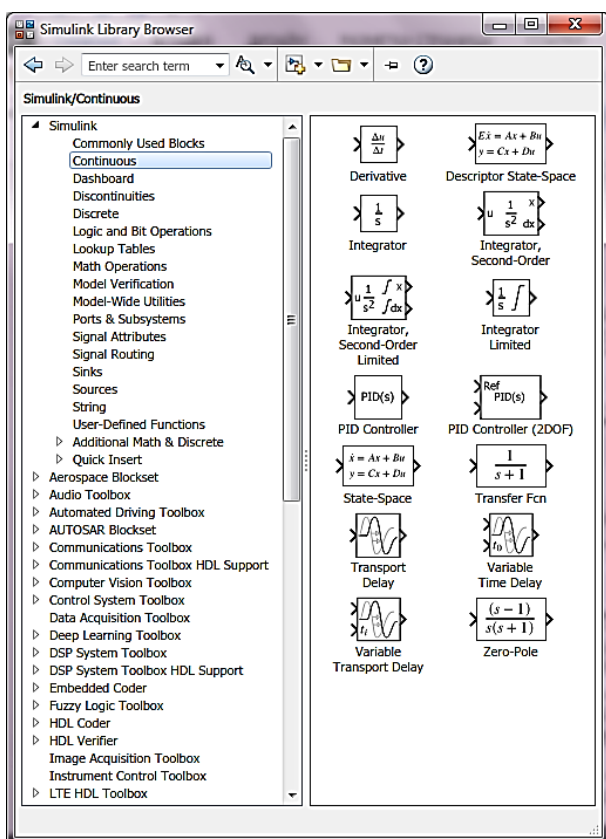


Рис. 2.1– Окно выбора типа проекта Simulink Start Page

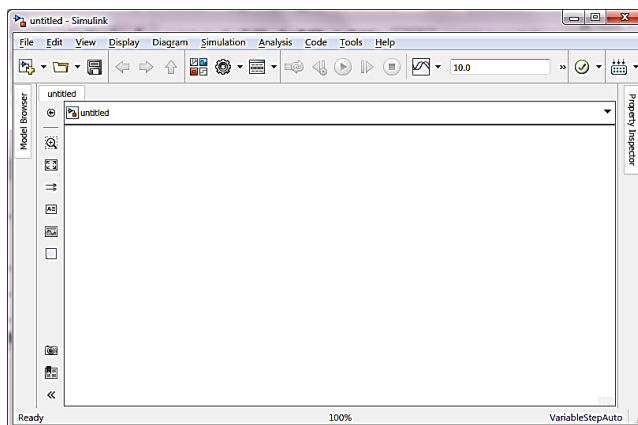
Если возникает необходимость продолжить работу с уже созданной ранее Simulink-схемой, то вызов системы Simulink происходит одновременно с открытием (Open) файла Simulink-схемы из среды MATLAB.

Для формирования новой Simulink-схемы или модификации существующей требуется получить доступ к библиотеке блоков Simulink. На рисунке 2.2 пред-

ставлены изображения браузера библиотеки Simulink (2.2 а) и нового (пустого) проекта модели Simulink (2.2 б).



а) Simulink Library Browser



б) Blank Model

Рис. 2.2– Начало работы с новым проектом Simulink

Главная библиотека Simulink содержит следующие *основные* разделы: **Continuous** – блоки непрерывных элементов; **Discontinuities** – блоки нелинейных элементов; **Discrete** – блоки дискретных элементов; **Math Operations** – блоки математических операций и функций; **Ports & Subsystems** – порты и подсистемы; **Signal Routing** – блоки преобразования и агрегации сигналов; **Sinks** - регистрирующие устройства (приёмники); **Sources** — источники сигналов и воздействий и др. (см. рисунок 2.2 а). При выборе соответствующего раздела библиотеки в правой части окна обозревателя отображаются все блоки этого раздела.

Наполнение поля Simulink-модели необходимыми блоками осуществляется путём "перетаскивания" требуемого блока из библиотеки на поле схемы. В дальнейшем блоки соединяются направленными линиями прохождения сигналов. Система Simulink предоставляет пользователю широкий набор инструментов для дизайнерских операций (толщина и цвет линий, заливка блоков, тени, текстовое оформление, выравнивание и т.п.).

В процессе формирования схемы модели желательно проводить необходимую настройку параметров устанавливаемых блоков, изменяя их значения "по умолчанию". Для этого необходимо дважды щелкнуть левой клавишей "мыши",

указав курсором на изображение блока. Откроется окно редактирования параметров данного блока (рисунок 2.3). При задании численных параметров следует иметь в виду, что в качестве десятичного разделителя должна использоваться точка, а не запятая. Изменения значений параметров настройки фиксируются кнопкой OK.

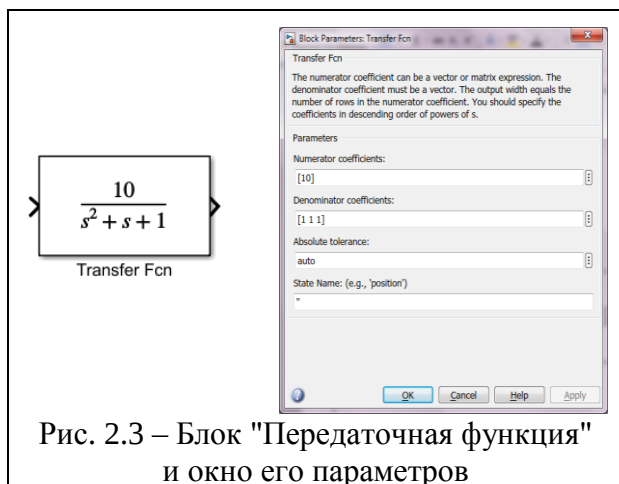


Рис. 2.3 – Блок "Передаточная функция" и окно его параметров

Схема Simulink-модели, в которой выполнены необходимые настройки и соединения блоков, проведено моделирование и получен график единичной переходной функции, показаны на рисунке 2.4.

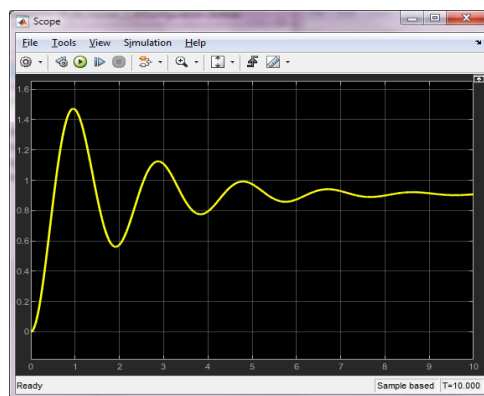
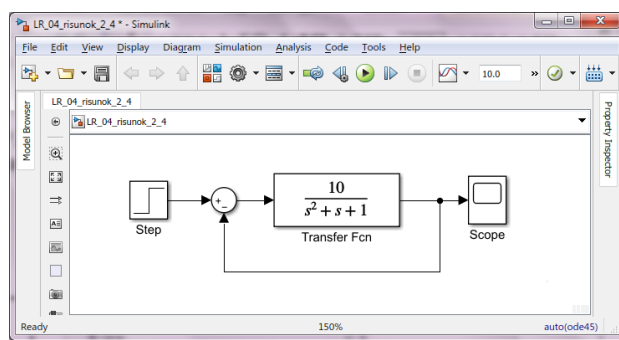


Рис. 2.4 – Схема модели и её реакция на функцию Хэвисайда

2.2 Соединение блоков

Для соединения блоков необходимо сначала установить курсор мыши на выходной порт одного из блоков. Курсор при этом превратится в большой крест из тонких линий. Держа нажатой левую кнопку "мыши", следует переместить курсор к входному порту нужного блока и отпустить левую клавишу "мыши". Свидетельством того, что соединение создано, будет жирная стрелка у входного порта блока. Возможно применение процедуры автоматического соединения блоков, для этого необходимо выделить блок-источник и затем, нажав клавишу **Ctrl**, выделить блок-приёмник.

Для создания точки разветвления в соединительной линии нужно подвести курсор к предполагаемому узлу и, нажав *правую* клавишу "мыши", протянуть линию входа в нужный блок схемы. Чтобы удалить линию необходимо её выделить левой кнопкой "мыши" (так же, как это выполняется и для блока), а затем нажать клавишу **Delete** на клавиатуре. Изменение конфигурации линии соединения выполняется путем её выделения и последующего перемещения "мышью" нужной части линии.

2.3 Установка параметров расчета и его выполнение

Перед моделированием, если это необходимо, можно изменить общие параметры расчета. Установка значений параметров расчета для текущей модели выполняется в окне Simulink-модели через опцию меню **Simulation > Model Configuration Parameters**.

Окно настройки параметров имеет несколько разделов, наиболее важный из них это раздел **Solver** (Решающее устройство) – установка параметров расчета модели. Здесь можно задать интервал моделирования, установив начальное (**Start time**) и конечное (**Stop time**) время моделирования; выбрать численный метод решения дифференциальных уравнений и способ выбора шага интегрирования; задать параметры точности расчетов и некоторые другие параметры. По умолчанию интервал моделирования задается равным 10 секундам (время начала моделирования равно нулю), численный метод интегрирования – метод Рунге-Кутты 4-го порядка с переменным шагом. Существует и глобальная настройка некоторых основных параметров пакета Simulink – в любом окне системы MATLAB через опцию меню **File > Preferences**.

2.4 Библиотека блоков Simulink

2.4.1 Sources - источники сигналов

Источник постоянного сигнала Constant – задает постоянный по уровню сигнал.

Параметры: **Constant value** - постоянная величина; **Interpret vector parameters as 1-D** – интерпретировать вектор параметров как одномерный (при установленном флажке).

Значение константы в поле **Constant value** может быть действительным или комплексным числом, вычисляемым выражением, вектором или матрицей.

Источник синусоидального сигнала Sine Wave – формирует синусоидальный сигнал с заданной частотой, амплитудой, фазой и смещением (постоянной составляющей).

Для формирования выходного сигнала блоком могут использоваться два алгоритма. Вид алгоритма определяется параметром **Sine Type** (способ формирования сигнала):

- **Time-based** – по текущему модельному времени (непрерывный сигнал).

Выходной сигнал по текущему значению времени для непрерывных систем.

Выходной сигнал источника в этом режиме соответствует выражению:

$$y = Amplitude * \sin(frequency * time + phase) + bias.$$

Параметры:

amplitude – амплитуда; **bias** – постоянная составляющая сигнала; **frequency (rads/sec)** - частота (рад/с); **phase (rads)** - начальная фаза (рад); **time** – модельное время;

— **Sample-based** – по задаваемому периоду дискретизации (дискретный сигнал).

Sample time – шаг модельного времени. Используется для согласования работы источника и других компонентов модели во времени. Этот параметр может принимать следующие значения:

- 0 (по умолчанию) – используется при моделировании непрерывных систем;
- >0 (положительное значение) – задается при моделировании дискретных систем. В этом случае шаг модельного времени можно интерпретировать как шаг квантования по времени выходного сигнала;
- минус 1 – шаг модельного времени устанавливается таким же, как и в предшествующем блоке, т.е. блоке, откуда приходит сигнал в данный блок.

Генератор ступенчатого сигнала **Step** – формирует ступенчатый сигнал.

Параметры:

Step time - время наступления перепада сигнала (с); **Initial value** - начальное значение сигнала; **Final value** - конечное значение сигнала.

Перепад может быть как в сторону увеличения сигнала (конечное значение больше чем начальное), так и в сторону уменьшения (конечное значение меньше чем начальное). Значения начального и конечного уровней могут быть не только положительными, но и отрицательными (например, изменение сигнала с уровня минус 5 до уровня минус 3).

Блок считывания данных из файла **From File** – получение данных из внешнего файла специального формата.

Параметры:

File Name - имя файла с данными; **Sample time** - шаг изменения выходного сигнала блока.

Данные в файле должны быть представлены в виде матрицы. Матрица должна состоять, как минимум, из двух строк. Значения времени записаны в первой строке матрицы, а в остальных строках находятся значения сигналов, соответствующие данным моментам времени. Значения времени должны быть записаны в возрастающем порядке. Выходной сигнал блока содержит только значения сигналов, а значения времени в нем отсутствуют. Если шаг расчета текущей модели не совпадает с отсчетами времени в файле данных, то **Simulink** выполняет линейную интерполяцию данных.

Файл данных (**mat**-файл), из которого считываются значения, не является текстовым. Структура файла подробно описана в справочной системе **MATLAB**. Пользователям **Simulink** удобнее всего создавать **mat**-файл с помощью блока **To File** (библиотека **Sinks**). На рисунке 2.6 показан пример использования данного блока. Из файла **data.mat** считываются значения синусоидального сигнала.

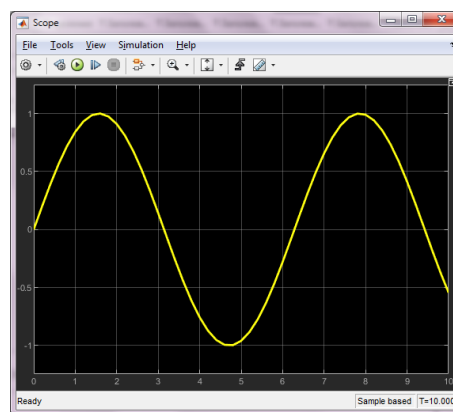
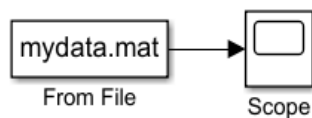


Рис. 2.5 – Использование блока From File

2.4.2 Sinks - приемники сигналов

Осциллограф Scope – строит графики исследуемых сигналов в функции времени.

Позволяет наблюдать за изменениями сигналов в процессе моделирования. Изображение блока и окно для просмотра графиков показаны на рисунке 2.6.

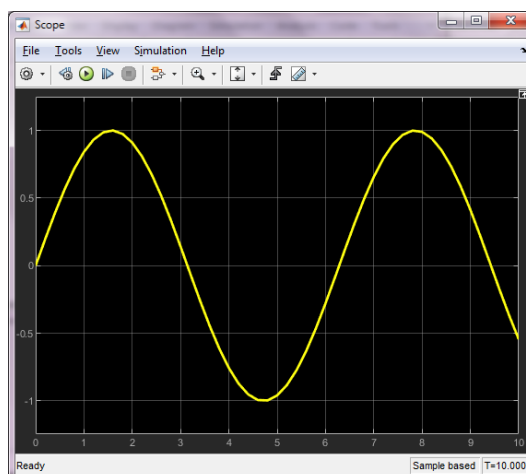
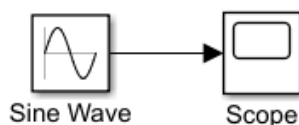


Рис. 2.6 – Осциллограф Scope

Блок Scope (Осциллограф) имеет множественные настройки, которые описаны в доступной справочной документации. Осциллограф может выступать как многолучевой и/или многооконный. Одним из главных достоинств является автоматическое масштабирование по обеим осям.

Графопостроитель XY Graph - строит график вида $Y(X)$.

Параметры: **x-min** – минимальное значение сигнала по оси X; **x-max** – максимальное значение сигнала по оси X; **y-min** – минимальное значение сигнала по оси Y; **y-max** – максимальное значение сигнала по оси Y; **Sample time** – шаг модельного времени.

Блок имеет два входа. Верхний вход предназначен для подачи сигнала, который является аргументом (X), нижний – для подачи значений функции (Y).

Графопостроитель можно использовать и для построения временных зависимостей. Для этого на первый вход следует подать временной сигнал с выхода

блока модельного времени **Clock**. Пример такого использования графопостроителя показан на рисунке 2.7

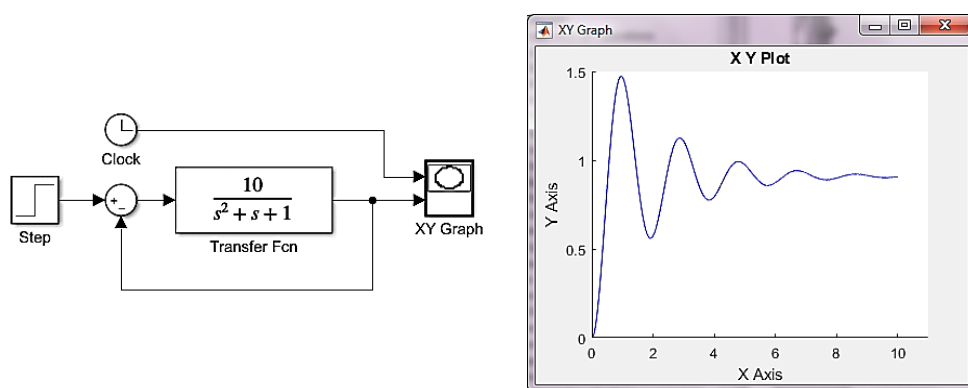


Рис. 2.7 – Пример использования блока **XY Graph** для отображения временных зависимостей

Цифровой дисплей Display - отображает текущее значение сигнала в виде числа.

Параметры:

– **Format** – формат отображения данных. Параметр **Format** для вещественных чисел может принимать следующие значения:

- **short** – 5 значащих десятичных цифр;
- **long** – 15 значащих десятичных цифр;
- **short_e** – 5 значащих десятичных цифр и 3 символа степени десяти;
- **long_e** – 15 значащих десятичных цифр и 3 символа степени десяти;
- **bank** – "денежный" формат. Формат с фиксированной точкой и двумя десятичными цифрами в дробной части числа;
- **hex, binary, decimal, octal** – целые числа в соответствующих системах счисления;

– **Decimation** – кратность отображения входного сигнала (прореживание отсчетов сигнала по времени). При **Decimation** = 1 отображается каждое значение входного сигнала, при **Decimation** = 2 отображается каждое второе значение, при **Decimation** = 3 – каждое третье значение и т.д.;

– **Floating display** (флажок) – перевод блока в "свободный" режим. В данном режиме входной порт блока отсутствует, а выбор сигнала для отображения выполняется щелчком левой клавиши "мыши" на соответствующей линии связи.

Блок **Display** может использоваться для отображения не только скалярных сигналов, но также векторных, матричных и комплексных.

Блок сохранения данных в файле To File - блок записывает данные, поступающие на его вход, в файл специального формата.

Параметры:

– **Filename** – имя файла для записи. По умолчанию файл имеет имя **untitled.mat**. Если не указан полный путь файла, то файл сохраняется в текущей рабочей папке;

– **Variable name** – имя переменной, содержащей записываемые данные;

– **Decimation** – кратность (прореживание) записи в файл входного сигнала. При **Decimation** = 1 записывается каждое значение входного сигнала, при **Decimation** = 2 записывается каждое второе значение и т.д.;

– **Save format** – формат упаковки данных в файл: **Timeseries** или **Array**:

– формат **Timeseries** – поддерживается запись данных любых встроенных типов: многомерных вещественных и комплекснозначных числовых потоков данных, логические, перечисляемые (enum), с фиксированной точкой;

– формат **Array** – поддерживает запись только векторных вещественных чисел. Данные в файле сохраняются в виде матрицы. Значения времени записываются в первой строке матрицы, а в остальных строках будут находиться значения сигналов, соответствующих данным моментам времени.

Файл данных (**mat**-файл), в который записываются данные, не является текстовым. Удобнее всего считывать данные из **mat**-файла с помощью блока **From File** (библиотека **Sources**).

2.4.3 Continuous – аналоговые блоки

Блок вычисления производной **Derivative** – выполняет численное дифференцирование входного сигнала. Для вычисления производной используется приближенная формула Эйлера.

Интегрирующий блок **Integrator** – выполняет интегрирование входного сигнала.

Параметры:

– **External reset** – внешний сброс. Тип внешнего управляющего сигнала, обеспечивающего сброс интегратора к начальному состоянию. Выбирается из списка;

– **Initial condition source** – вид источника (внутренний или внешний) установки начального значения;

– **Initial condition** – начальное значение выхода блока.

На рисунке 2.8 показан пример работы интегратора при подаче на его вход единичного ступенчатого сигнала с секундной задержкой. Начальное условие принято равным минус 2,5.

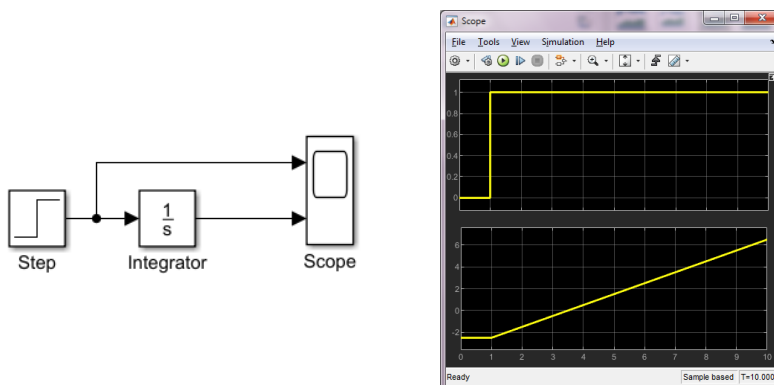


Рис. 2.8 - Интегрирование ступенчатого сигнала

Пример на рисунке 2.9 отличается от предыдущего подачей начального значения через внешний порт (минус 1,5). Установлено также ограничение выходного сигнала интегратора на уровне 6 единиц.

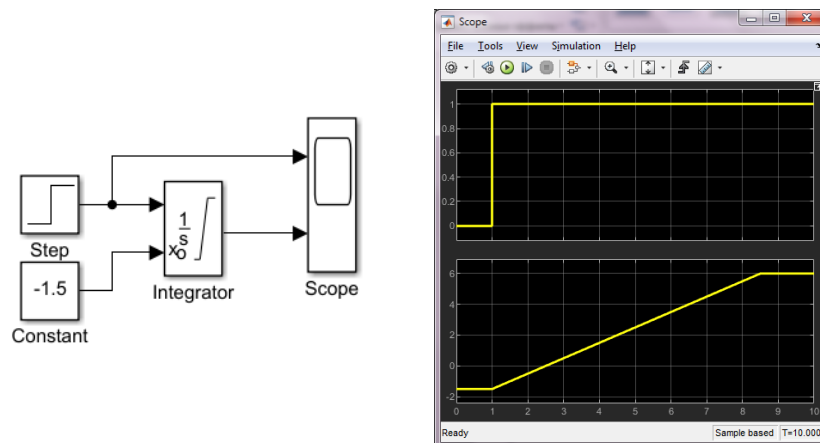


Рис. 2.9 – Интегрирование ступенчатого сигнала с внешней установкой начального значения и ограничения выходного сигнала

Блок модели динамического объекта State-Space – создает линейную стационарную (LTI-system) динамическую систему, математическая модель которой задается уравнениями в пространстве состояний:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t).\end{aligned}$$

$$\begin{cases} x' = Ax + Bu \\ y = Cx + Du \end{cases}$$

State-Space

Параметры:

- **A** – матрица динамических коэффициентов системы;
- **B** – матрица коэффициентов усиления по сигналам входа;
- **C** – матрица коэффициентов усиления по сигналам выхода;
- **D** – матрица прямого влияния входа на выход;
- **Initial condition** – вектор начальных условий.

Блок передаточной функции Transfer Fcn – задает математическую модель линейной стационарной системы (LTI-system) в виде передаточной функции (ПФ) как отношение полиномов $\text{num}(s)/\text{den}(s)$:

$$H(s) = \frac{y(s)}{u(s)} = \frac{\text{num}(s)}{\text{den}(s)} = \frac{\text{num}_1 s^m + \text{num}_2 s^{m-1} + \dots + \text{num}_m s + \text{num}_{m+1}}{\text{den}_1 s^n + \text{den}_2 s^{n-1} + \dots + \text{den}_n s + \text{den}_{n+1}},$$

где **m** и **n** – порядки полиномов числителя и знаменателя ПФ, **num** – вектор или матрица коэффициентов числителя (**numerator**), **den** – вектор (только вектор) коэффициентов знаменателя (**denominator**).

Параметры:

- **Numerator coefficient** – вектор или матрица коэффициентов полинома числителя;
- **Denominator coefficient** – вектор коэффициентов полинома знаменателя ПФ.

Порядок числителя m не должен превышать порядка знаменателя n .

Входной сигнал блока всегда должен быть скалярным. В том случае, если коэффициенты числителя заданы вектором-строкой, то выходной сигнал блока будет также скалярным (как и входной сигнал).

Если коэффициенты числителя заданы матрицей, то блок **Transfer Fcn** моделирует ЛТИ-систему с одним входом и несколькими выходами (выходов столько, сколько строк в матрице числителя). Данную векторную ПФ, которую можно интерпретировать как несколько ПФ, имеющих одинаковые полиномы знаменателя, но разные полиномы числителя (строки матрицы). При этом выходной сигнал блока является векторным и количество строк матрицы числителя задает размерность выходного сигнала.

На рисунке 2.10 показан пример блока **Transfer Fcn**, задающий векторную ПФ. Там же показана модель полностью аналогичная рассматриваемой по своим свойствам, но состоящая из отдельных блоков: **Transfer Fcn1** и **Transfer Fcn2**.

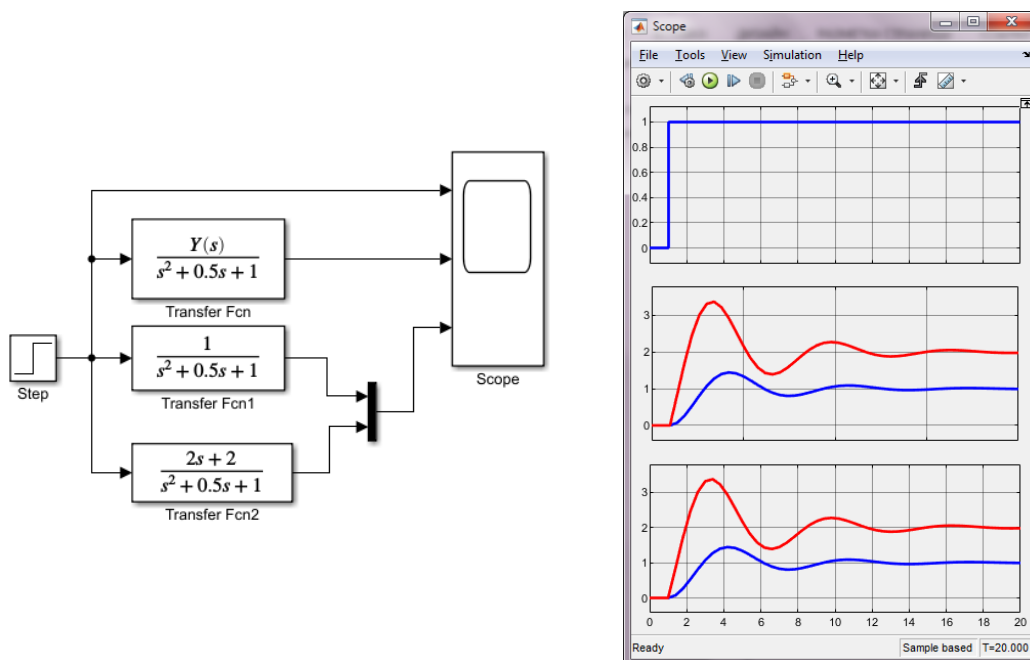


Рис. 2.10 – Моделирования векторной ПФ и её скалярного аналога

Блок передаточной функции Zero-Pole – задает математическую модель линейной стационарной системы (LTI-system) в виде ПФ, заданной нулями, полюсами и коэффициентом усиления:

$$H(s) = K \frac{(s - Z_1)(s - Z_2) \dots (s - Z_m)}{(s - P_1)(s - P_2) \dots (s - P_n)},$$

где $\{Z\}$ – вектор или матрица нулей ПФ (нулей полинома числителя Z_i), $\{P\}$ – вектор полюсов ПФ (нулей полинома знаменателя P_i), K – коэффициент усиления ПФ или вектор $\{K\}$ коэффициентов усиления (K_i), если нули ПФ заданы

матрицей. При этом размерность вектора **K** определяется числом строк матрицы нулей.

Параметры:

- **Zeros** – вектор или матрица нулей;
- **Poles** – вектор полюсов;
- **Gain** – скалярный или векторный коэффициент усиления ПФ.

Количество нулей не должно превышать число полюсов передаточной функции. В том случае, если нули передаточной функции заданы матрицей, то блок **Zero-Pole** моделирует векторную передаточную функцию.

Нули или полюса могут быть заданы комплексными числами. В этом случае нули или полюса должны быть заданы комплексно-сопряженными парами полюсов или нулей, соответственно.

2.4.4 Использование переменных для задания параметров блоков

Параметры блоков **Simulink** можно задавать не только как числовые параметры, но и как имена (идентификаторы) переменных, существующих в данный момент в рабочей области **WorkSpace** системы **MATLAB**. Значения переменных необходимо задать перед началом расчета. Такой способ задания параметров удобен, если требуется проводить расчеты для множества вариантов параметров блоков модели. На рисунке 2.11а показана схема такой модели. Параметры блока "Передаточная функция" заданы переменными **a**, **b** и **c**. Окно параметров блока "Передаточная функция" показано на рисунке 2.11б.

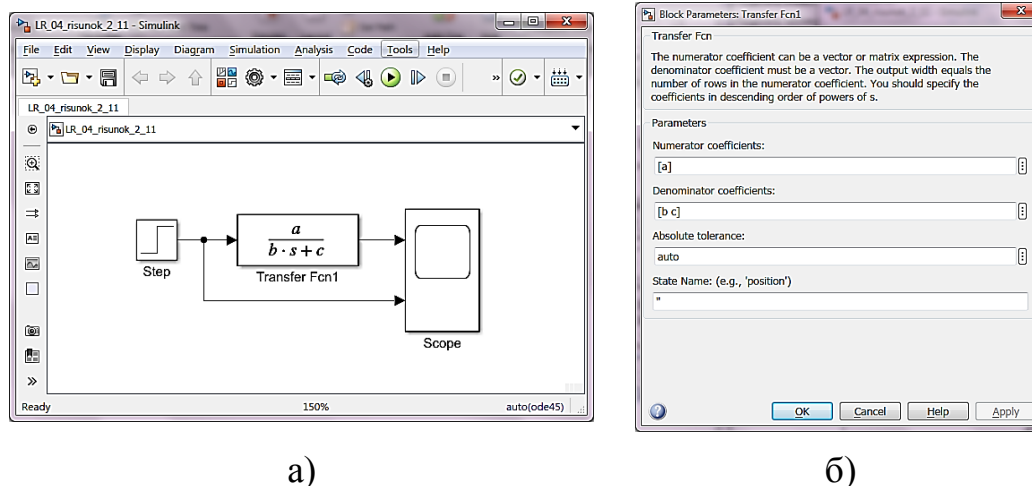


Рис. 2.11 - Параметры блока Transfer Fcn заданы как имена переменных из **WorkSpace**

В том случае, если **Simulink**-схема сложная и параметров много, можно создать отдельный **m**-файл с командами присвоения переменным значений, открыть его из командного окна **MATLAB** (файл будет открыт в окне редактора-отладчика) и запустить его на выполнение. После выполнения данного **m**-файла параметры блоков **Simulink**-модели под соответствующими именами появятся в рабочем пространстве **MATLAB** и станут доступными для настроек параметров

блоков модели. Более того, некоторые параметры блоков Simulink могут быть заранее неизвестны, и их необходимо вычислять на основе применения довольно сложных алгоритмов. Эти вычисления могут быть проведены в том же *m*-файле.

Итак, если Simulink-модель содержит вычисляемые параметры, то перед запуском Simulink-модели эти параметры должны находиться в **Workspace MATLAB**, т.е., предварительно нужно запустить на счет *m*-файл с соответствующими расчетами. Таким образом, для проведения моделирования необходимо *последовательно* запустить на счет два файла: сначала *m*-файл с расчетами коэффициентов, а затем *mdl*-файл Simulink-модели для проведения собственно моделирования. Указанные выше действия можно автоматизировать, например, такими способами:

- в *m*-файле, где проводятся расчеты коэффициентов, последней инструкцией должна быть команда запуска на моделирование предварительно открытой Simulink-модели: `sim('имя mdl-файла');`

- настроить Simulink-модель следующим образом: перейти в раздел меню окна Simulink-модели **File>Model Properties>Model Properties>Callbacks**; в левом окне закладки **Callbacks** выбрать функцию **InitFcn**, а в правом окне записать имя *m*-файла. При такой настройке, запуск Simulink-модели приведет сначала к исполнению *m*-файла и соответственно, записи параметров в **Workspace**, а затем собственно к моделированию. Коэффициенты модели будут подготовлены и взяты из **Workspace**.

2.4.5 Запись результатов расчета в текстовый файл

Для сохранения процесса моделирования (отсчётов временных сигналов) в текстовом файле следует упаковать эти сигналы в именованный числовой массив и передать его в рабочее пространство **MATLAB**, а по окончании моделирования отдельной командой записать эти данные в текстовый файл.

Пример модели, реализующей данный принцип, показан на рисунке 2.12. В окне параметров блока **To Workspace** необходимо указать имя переменной для вывода в рабочее пространство (по умолчанию, это имя – **simout**), а также установить значение параметра **Save format** (формат записи) - **Array** (массив).

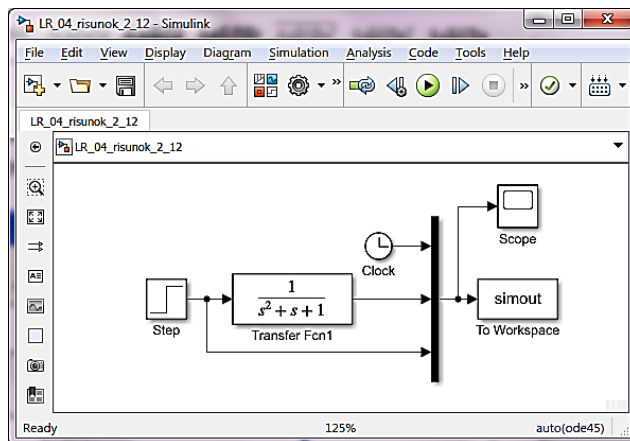


Рис. 2.12 - Вывод результатов моделирования в **Workspace**

В массиве **simout** данные будут храниться в виде трёх столбцов чисел.

Для записи данных из массива **simout** в текстовый файл необходимо в командном окне **MATLAB** задать на выполнение команду вида:

```
save('outfile.txt','-ascii','simout','-double')
```

где **outfile.txt** – имя файла для записи; **-ascii** – текстовый файл; **simout** – имя переменной, значение которой записывается в текстовый файл; **-double** – режим записи с удвоенной точностью.

Процедура записи переменной из рабочего пространства **MATLAB** в файл текстового формата можно также реализована функцией **dlmwrite**. Данная функция может иметь несколько параметров, с помощью которых настраивается процедура записи данных. В качестве примера использования данной функции рассмотрим следующее объявление:

```
dmlwrite('file.txt',simout,'delimiter','\t','precision','%20.11f');
```

здесь производится запись элементов числовой матрицы **simout** в текстовый файл с именем **file.txt**. В качестве разделителя (**delimiter**) элементов матрицы в файле выбран символ "табулятор" (**\t**), числовые данные записываются в файл как числа вещественного типа с фиксированным количеством цифр после запятой (**%20.11f**).

3 МАТЕМАТИЧЕСКИЕ МОДЕЛИ И СХЕМЫ SIMULINK

3.1 Математическая модель двигателя постоянного тока

Рассмотрим построение математической модели двигателя постоянного тока (ДПТ). Вход этого объекта – напряжение в цепи якоря $u(t)$ (в вольтах), выход – угол поворота вала $\theta(t)$ (в радианах). Схема ДПТ, управляемого по цепи якоря, приведена на рисунке 3.1. Вал двигателя начинает вращаться, когда приложено управляющее напряжение $u(t)$. Если напряжение не меняется, угловая скорость вращения вала двигателя $\omega(t)$ (в радианах в секунду) остается постоянной, при этом угол $\theta(t)$ равномерно увеличивается.

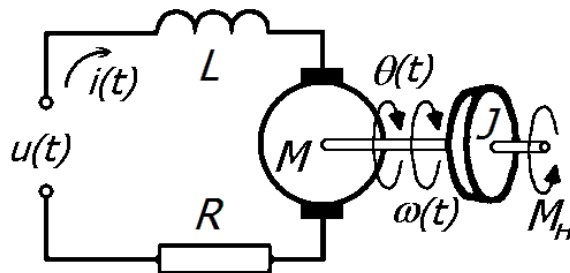


Рис. 3.1 – Схема двигателя постоянного тока

Скорость и направление вращения вала ДПТ зависит от величины и знака напряжения $u(t)$. Чем больше напряжение $u(t)$, тем быстрее вращается вал. Если подключить нагрузку, например, заставить двигатель вращать турбину, то скорость вращения постепенно уменьшается до нового значения, при котором

вращающий момент двигателя $M(t)$ будет равен моменту нагрузки $M_H(t)$ (сопротивления). Пока эти моменты равны, угловая скорость вращения $\omega(t)$ остается постоянной и ее производная $\dot{\omega}(t)$ равна нулю.

Построение математической модели ДПТ основано на применении законов механики и электротехники – в данном случае – динамики вращательного движения, второго правила Кирхгофа, законов электромеханического преобразования энергии.

Угловая скорость вращения $\omega(t)$ вычисляется как производная от угла поворота вала $\theta(t)$, т.е.

$$\omega(t) = \frac{d\theta(t)}{dt}.$$

Соответственно, угол $\theta(t)$ – это интеграл от угловой скорости $\omega(t)$.

В механике уравнение вращательного движения записывают в виде

$$J \frac{d\omega(t)}{dt} = M(t) - M_H(t),$$

где $M(t)$ – вращающий момент (измеряется в Н·м), $M_H(t)$ – момент нагрузки (возмущение, измеряется в Н·м), J – суммарный момент инерции всех вращающихся масс, приведенных к оси вала двигателя (кг·м²). Величина момента инерции говорит о том, насколько легко "разгоняется" двигатель (чем больше момент инерции, тем сложнее "разогнать").

Перейдем к электротехнической составляющей модели. Ток $i(t)$, протекающий в цепи якоря, создает вращающий момент $M(t)$, значение которого вычисляется по формуле

$$M(t) = k_m \cdot i(t),$$

где k_m – электромеханическая конструктивная постоянная, $i(t)$ – ток в обмотке якоря (в амперах) – может быть найден из уравнения, построенного на основании второго правила Кирхгофа

$$u(t) = L \frac{di(t)}{dt} + R \cdot i(t) + e(t),$$

где $u(t)$, $e(t) = k_e \omega(t)$ – напряжение и противоЭДС якоря; L , R – индуктивность и активное сопротивление; k_e – электрическая конструктивная постоянная.

Таким образом, система уравнений, которыми описываются электрические, механические и электромеханические процессы в двигателе постоянного тока с независимым возбуждением имеют вид:

$$\begin{cases} u(t) = L \frac{di(t)}{dt} + R \cdot i(t) + e(t), \\ J \frac{d\omega(t)}{dt} = M(t) - M_H(t), \\ \omega(t) = \frac{d\theta(t)}{dt}, \quad e(t) = k_e \omega(t), \quad M(t) = k_m i(t). \end{cases} \quad (3.1)$$

В качестве примера двигателя взят малоинерционный ДПТ с электромагнитным возбуждением, применяемый в робототехнических комплексах. Параметры двигателя, входящие в уравнения модели (3.1) приведены в таблице 3.1. Часть этих параметров являются паспортными данными, часть – рассчитывается по точным и эмпирическим формулам [3].

Таблица 3.1

Параметры двигателя

P_H кВт	$U_{\text{я}}$ В	$i_{\text{я}}$ А	R Ом	J кг·м ²	n_H об/мин	L мГн	k_e В·с	k_m Н·м/А
0,45	110	5,6	0,585	0,36	3000	25,2	0,34	0,26

В таблице 3.1 введены следующие обозначения:

- P_H – номинальная мощность двигателя [кВт];
- $U_{\text{я}}$ – номинальное напряжение питания якоря [В];
- $i_{\text{я}}$ – номинальный ток в цепи якоря [А];
- R – активное сопротивление обмотки якоря [Ом];
- J – момент инерции якоря и вала двигателя [кг·м²];
- n_H – номинальная скорость вращения якоря [об/мин];
- L – индуктивность обмотки якоря [мГн];
- k_e, k_m – конструктивные постоянные двигателя.

Часто достаточно знать, как будет изменяться управляемый выход объекта при подаче на вход заданного входного сигнала (управления). При этом поведением внутренних переменных объекта не интересуются. В нашем случае управляемой выходной переменной является угол поворота вала двигателя $\theta(t)$, управлением – напряжение, подаваемое в цепь якоря $u(t)$. Переходя в уравнениях (3.1) к переменной $\theta(t)$, получим уравнение

$$J \cdot L \frac{d^3\theta(t)}{dt^3} + J \cdot R \frac{d^2\theta(t)}{dt^2} + k_m k_e \frac{d\theta(t)}{dt} = k_m u(t) - R \cdot M_H(t) - L \frac{dM_H(t)}{dt}. \quad (3.2)$$

Данное уравнение является дифференциальным уравнением третьего порядка, связывающим управляемый вход $u(t)$ и нагрузку $M_H(t)$ с выходом $\theta(t)$. Все внутренние сигналы исходной модели ($i(t)$, $\omega(t)$, $e(t)$, $M(t)$) исключены из уравнения. Полученное уравнение называется уравнением (моделью) "вход-выход".

Порядком модели называют порядок соответствующего дифференциального уравнения. В данном случае мы получили модель третьего порядка.

Если считать, что управляемым выходом является угловая скорость вращения вала двигателя $\omega(t)$, то уравнение (3.2) может записано в виде

$$J \cdot L \frac{d^2 \omega(t)}{dt^2} + J \cdot R \frac{d\omega(t)}{dt} + k_m k_e \omega(t) = k_m u(t) - R \cdot M_H(t) - L \frac{dM_H(t)}{dt}. \quad (3.3)$$

Применяя к последнему уравнению преобразование Лапласа и предполагая, что начальные значения угловой скорости $\omega_0 = \omega(t)|_{t=0}$ и её первой производной равны нулю (нулевые начальные условия), получим следующее соотношение в изображениях

$$J \cdot L \cdot s^2 \omega(s) + J \cdot R \cdot s \omega(s) + k_e k_m \omega(s) = k_m u(s) - (L \cdot s + R) M_H(s).$$

Разрешая последнее алгебраическое соотношение относительно изображения угловой скорости $\omega(s)$, получим

$$\omega(s) = \frac{k_m}{J \cdot L \cdot s^2 + J \cdot R \cdot s + k_m k_e} u(s) - \frac{L \cdot s + R}{J \cdot L \cdot s^2 + J \cdot R \cdot s + k_m k_e} M_H(s)$$

или

$$\omega(s) = W_{\omega u}(s) \cdot u(s) + W_{\omega M_H}(s) \cdot M_H(s). \quad (3.3)$$

В соотношении (3.3) введены следующие обозначения:

$$W_{\omega u}(s) = \frac{k_m}{J \cdot L \cdot s^2 + J \cdot R \cdot s + k_m k_e} \quad \text{— передаточная функция ДПТ по управляющему воздействию;} \quad (3.4)$$

$$W_{\omega M_H}(s) = -\frac{L \cdot s + R}{J \cdot L \cdot s^2 + J \cdot R \cdot s + k_m k_e} \quad \text{— передаточная функция ДПТ по возмущению.} \quad (3.5)$$

Часто передаточные функции (ПФ) представляют в одном из стандартных видов, например, в нашем случае это может быть следующий вид:

$$W_{\omega u}(s) = \frac{1/k_e}{\frac{L}{R} \cdot \frac{J \cdot R}{k_m k_e} s^2 + \frac{J \cdot R}{k_m k_e} s + 1} = \frac{k_u}{T_\gamma T s^2 + T s + 1} \approx \frac{2,9}{0,1s^2 + 2,38s + 1}; \quad (3.6)$$

$$W_{\omega M_H}(s) = \frac{\frac{R}{k_m k_e} \left(\frac{L}{R} s + 1 \right)}{\frac{L}{R} \cdot \frac{J \cdot R}{k_m k_e} s^2 + \frac{J \cdot R}{k_m k_e} s + 1} = \frac{k_M (T_\gamma s + 1)}{T_\gamma T s^2 + T s + 1} \approx \frac{6,6 \cdot (0,043s + 1)}{0,1s^2 + 2,38s + 1}. \quad (3.7)$$

В выражениях (3.6) и (3.7) для ПФ ДПТ по управлению и возмущению введены параметры:

- $T_\gamma = \frac{L}{R} = 0,043$ — электрическая постоянная времени якорной цепи ДПТ;
- $T = \frac{J \cdot R}{k_e k_m} = 2,38$ — электромеханическая постоянная времени ДПТ;

– $k_u=2,9$ и $k_M=6,6$ – коэффициенты усиления по управляющему и возмущающему воздействиям, соответственно.

Simulink-модель ДПТ в терминах введенных ПФ (3.6) и (3.7) приведена на рисунке 3.2.

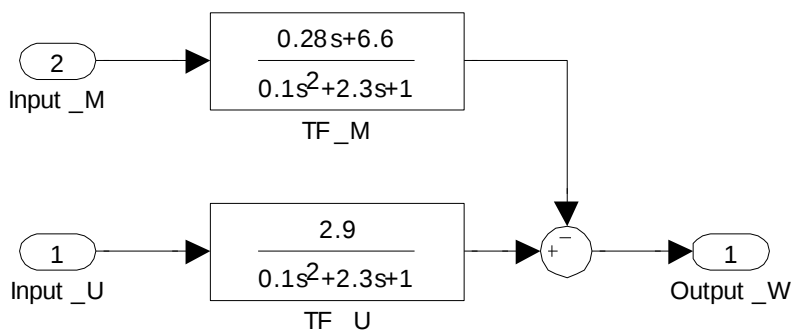


Рис. 3.2 – Simulink-модель ДПТ (модель в виде ПФ)

Перейдем к построению математической модели ДПТ в пространстве состояний [4]. Данный тип модели представляет собой систему обыкновенных дифференциальных уравнений первого порядка, разрешенных относительно производных искомых функций. В теории дифференциальных уравнений такая форма системы называется нормальной формой или формой Коши.

При построении модели объекта управления в пространстве состояний обычно в качестве переменных состояния объекта выбирают физические переменные, описывающие не только внешнее поведение объекта (как в модели "вход-выход"), но и внутренние процессы, протекающие в нем. При назначении переменных состояния имеется определенная свобода, однако, как бы ни были выбраны переменные состояния объекта, модель должна сохранять внешнее соотношение между входными и выходными сигналами (переменными). Модель в пространстве состояний дает больше информации о внутреннем поведении объекта, а значит, появляется возможность более эффективного достижения целей управления.

Для построения модели ДПТ в пространстве состояний обратимся к системе уравнений и соотношений (3.1). Выберем в качестве переменных состояния угол поворота $\theta(t)$ и угловую скорость вращения $\omega(t)$ вала ДПТ, а также ток в цепи якоря $i(t)$. Внешними воздействиями остаются управляющее $u(t)$ и возмущающее $M_H(t)$ воздействия. Выходной управляемой переменной остается угол $\theta(t)$.

Запишем дифференциальные уравнения для выбранных переменных состояния в нормальной форме –

$$\begin{cases} \frac{d\theta(t)}{dt} = \omega(t), \\ \frac{d\omega(t)}{dt} = \frac{k_m}{J} i(t) - \frac{1}{J} M_H(t), \\ \frac{di(t)}{dt} = -\frac{R}{L} i(t) - \frac{k_e}{L} \omega(t) + \frac{1}{L} u(t). \end{cases} \quad (3.8)$$

Объединение переменных состояния в вектор-столбец $[\theta(t) \ \omega(t) \ i(t)]^T$ даёт возможность представить систему (3.8) в векторно-матричной форме

$$\frac{d}{dt} \begin{bmatrix} \theta(t) \\ \omega(t) \\ i(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \frac{k_m}{J} \\ 0 & -\frac{k_e}{L} & -\frac{R}{L} \end{bmatrix} \cdot \begin{bmatrix} \theta(t) \\ \omega(t) \\ i(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} \cdot u(t) + \begin{bmatrix} 0 \\ -\frac{1}{J} \\ 0 \end{bmatrix} \cdot M_H(t). \quad (3.9)$$

Переменные $\theta(t)$, $\omega(t)$ и $i(t)$ определяют состояние двигателя в момент времени t . Это значит, что зная их значения в некоторый момент времени t_0 и входные сигналы $u(t)$ и $M_H(t)$ при всех $t \geq t_0$ можно рассчитать поведение объекта для любого последующего момента. При этом предыдущие значения $\theta(t)$, $\omega(t)$, $i(t)$, $u(t)$ и $M_H(t)$ (при $t < t_0$) не играют никакой роли. Поэтому $\theta(t)$, $\omega(t)$ и $i(t)$

называются переменными состояния, а вектор $\begin{bmatrix} \theta(t) \\ \omega(t) \\ i(t) \end{bmatrix}$ – вектором состояния.

Управляемый выход объекта $\theta(t)$ может быть представлен через вектор состояния $[\theta(t) \ \omega(t) \ i(t)]^T$ и управляющее входное воздействие $u(t)$ в виде алгебраического векторно-матричного соотношения

$$\theta(t) = [1 \ 0 \ 0] \cdot \begin{bmatrix} \theta(t) \\ \omega(t) \\ i(t) \end{bmatrix} + [0] \cdot u(t). \quad (3.10)$$

Уравнения (3.9) и (3.10) называют соответственно уравнениями состояния и уравнениями выходов объекта. В теории управления принято обозначать вектор состояния объекта через $x(t)$, управляемый выход – через $y(t)$, управляющий вход (сигнал управления) – через $u(t)$, внешнее возмущение – через $f(t)$. Тогда модель (3.9), (3.10) может быть записана в виде

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) + Ff(t), \\ y(t) = Cx(t) + Du(t). \end{cases}$$

Матрицы A , B , F , C и D приведены ниже

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \frac{k_m}{J} \\ 0 & -\frac{k_e}{L} & -\frac{R}{L} \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix}; \quad F = \begin{bmatrix} 0 \\ -\frac{1}{J} \\ 0 \end{bmatrix}; \quad C = [1 \ 0 \ 0]; \quad D = [0].$$

Аналогично структуре на рисунке 3.2, полученной на основании модели "вход-выход" и перехода к ПФ по управлению и возмущению, можно

построить структуру объекта управления и на основе модели в пространстве состояний (3.8) или (3.9).

Применим преобразование Лапласа к системе уравнений (3.8). Предполагая нулевые начальные условия, т.е. равенство нулю значений всех переменных состояния в начальный момент времени $t_0=0$, получим следующую линейную алгебраическую систему уравнений относительно изображений переменных состояния и входов $\theta(s)$, $\omega(s)$, $i(s)$, $u(s)$ и $M_H(s)$

$$\begin{cases} s\theta(s) = \omega(s), \\ s\omega(s) = \frac{k_m}{J} i(s) - \frac{1}{J} M_H(s), \\ si(s) = -\frac{R}{L} i(s) - \frac{k_e}{L} \omega(s) + \frac{1}{L} u(s) \end{cases} \quad \text{или} \quad \begin{cases} \theta(s) = \frac{1}{s} \omega(s), \\ \omega(s) = \frac{k_m}{s \cdot J} i(s) - \frac{1}{s \cdot J} M_H(s), \\ i(s) = \frac{1}{(s \cdot L + R)} (u(s) - k_e \omega(s)). \end{cases}$$

На основании второго и третьего соотношений последней системы можно составить следующую структурную схему *скоростного* контура ДПТ (управляемым выходом будем считать угловую скорость вращения вала $\omega(t)$).

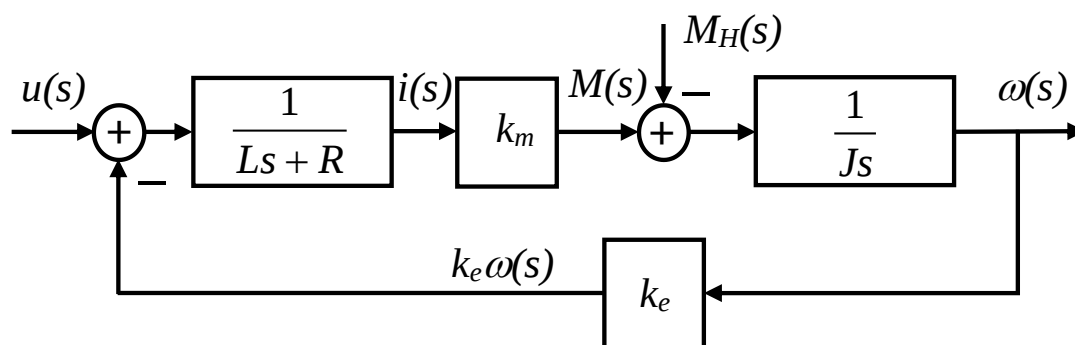


Рис. 3.3 – Структурная схема скоростного контура ДПТ

Simulink-модель в терминах *изображений* переменных пространства состояний, построенная в соответствии с уравнениями (3.8) и структурой (рисунок 3.3), представлена на следующем рисунке.

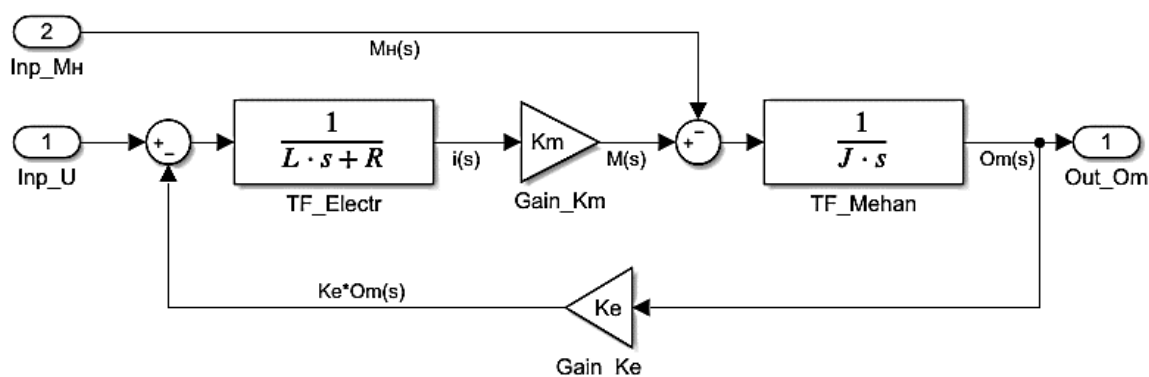


Рис. 3.4 – Simulink-модель ДПТ (модель пространства состояний)

Структурная схема системы стабилизации угла поворота вала ДПТ представлена на рисунке 3.5. Здесь структура самого ДПТ околнурена красным пунктиром. Входными воздействиями на ДПТ являются управляющая пере-

менная u и возмущающий момент нагрузки M_H , а выходным сигналом – управляемая переменная θ – угол поворота вала ДПТ. Задающее воздействие r на входе в САУ с помощью главной отрицательной обратной связи сравнивается с управляемой переменной ДПТ θ – углом поворота вала. Сигнал ошибки ε отработки задающей уставки r подаётся на регулятор P , вырабатывающий сигнал управления u – напряжение, подаваемое на обмотку якоря ДПТ.

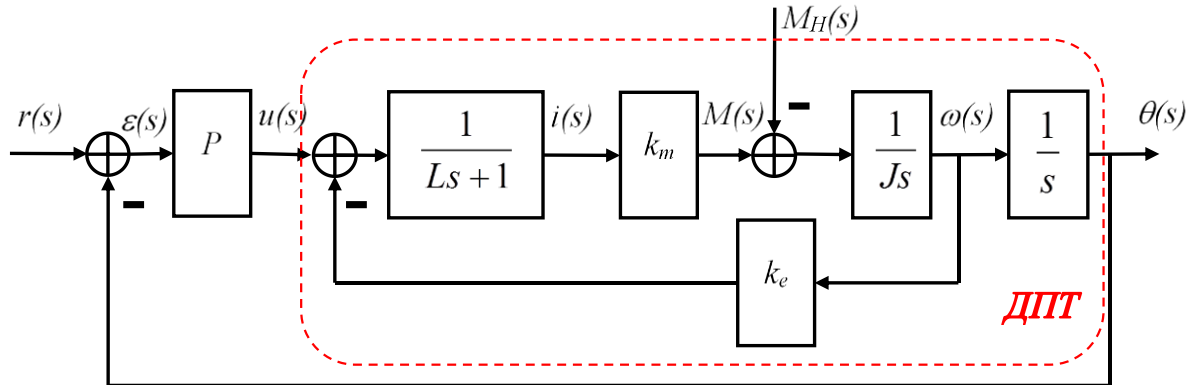


Рис. 3.5 – Структурная схема системы стабилизации ДПТ по углу

Simulink-модель, построенная в соответствии со структурой рисунка 3.5, представлена ниже.

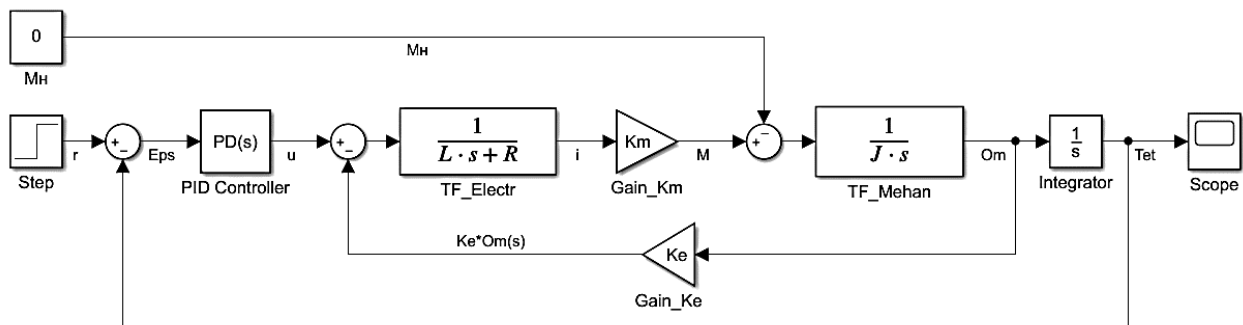


Рис. 3.6 – Simulink-модель системы стабилизации ДПТ по углу

Если для построения Simulink-схемы использовать модель ДПТ в форме модели пространства состояний (3.9), (3.10), схему рисунка 3.6 можно представить, например, так

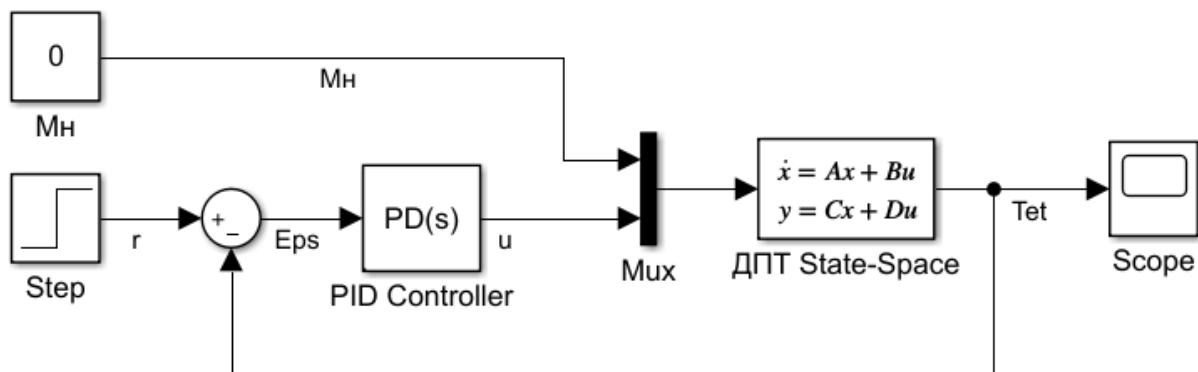
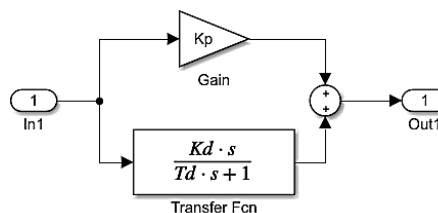


Рис. 3.7 – Simulink-модель системы стабилизации ДПТ с блоком State Space

Системы стабилизации Simulink-моделей рисунков 3.6 и 3.7 в качестве регулятора используют пропорционально-дифференциальный (ПД) регулятор PID Controller. Передаточная функция физически реализуемого ПД-регулятора и его Simulink-схема имеют следующий вид:

$$W_p(s) = k_p + \frac{k_D s}{T_D s + 1}$$

ПФ ПД-регулятора



Simulink-схема ПД-регулятора

3.2 Моделирование одноконтурной САУ ДПТ по скорости

Для придания скоростному контуру требуемых динамических свойств используются регуляторы, структура и коэффициенты которых зависят от параметров ДПТ и требований, предъявляемых к скоростному контуру.

Simulink-схема системы с исполнительным ДПТ, управляемым по цепи якоря, изображена на рисунке 3.8. Она включает сам двигатель (блоки, выделенные голубым цветом), представленный последовательно включенными апериодическим звеном первого порядка TF_Electr, идеальным усилительным звеном Gain_Km и интегрирующим звеном TF_Mehan, охваченными обратной связью через усилительное звено Gain_Ke. В качестве регулятора выбран простейший П-регулятор (PID Controller) – идеальный усилитель с заданным коэффициентом усиления – установленный в прямую цепь САУ. Управляющее воздействие $u(t)$ на ДПТ формируется по пропорциональному закону:

$$u(t) = k_p \varepsilon(t).$$

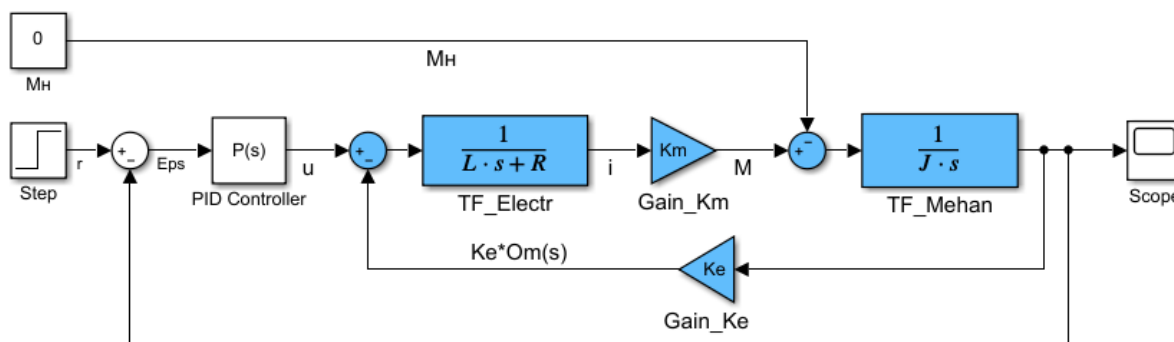
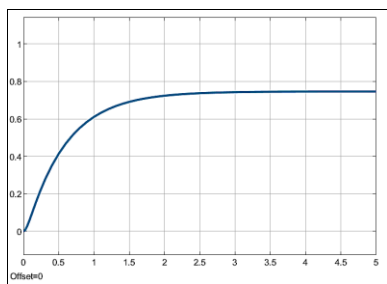
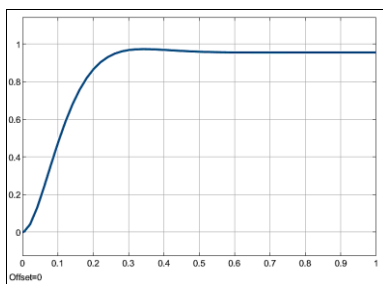


Рис. 3.8 – Simulink-схема САУ скорости ДПТ с П-регулятором

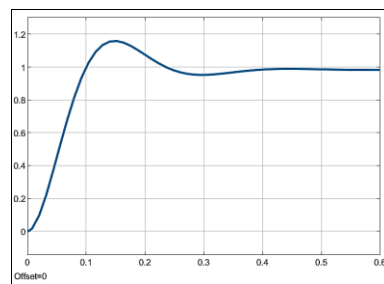
Выполнено моделирование САУ скорости ДПТ с П-регулятором для трёх значений коэффициента усиления $k_p = \{1.0; 7.3; 20.0\}$. На вход САУ подавался сигнал в виде функции единичного скачка $1(t)$ (блок Step). Внешнее возмущение не учитывалось (блок Const) $Mn=0$. Результаты моделирования приведены на рисунке 3.9 в виде графиков изменения выходной управляемой координаты – угловой скорости вращения вала двигателя $\omega(t)$.



а) $k_p=1,0$



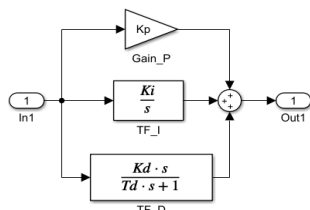
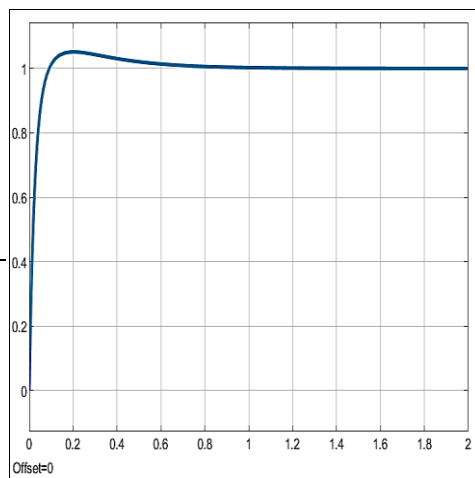
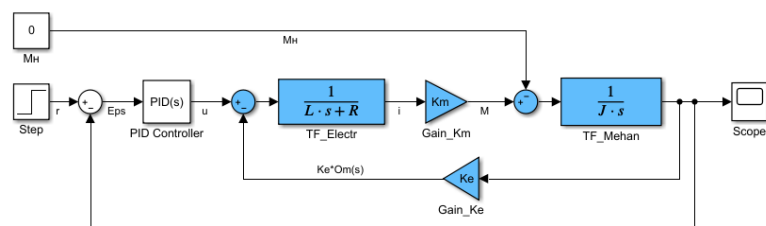
б) $k_p=7,3$



в) $k_p=20,0$

Рис. 3.9 – Результаты моделирования САУ скорости ДПТ с П-регулятором

Проводилось также моделирование САУ скорости ДПТ с ПИД-регулятором. Схема, ПФ ПИД-регулятора, его коэффициенты и результаты моделирования представлены на рисунке 3.10.



$$W_p(s) = k_p + \frac{k_i}{s} + \frac{k_d s}{T_d s + 1}$$

$$k_p = 30,9; k_i = 94,7;$$

$$k_d = 1,4; T_d = 2,26 \cdot 10^{-4}$$

а) Simulink-схема с ПИД-регулятором

б) график выхода $\omega(t)$

Рис. 3.10 – Моделирования САУ скорости ДПТ с ПИД-регулятором

3.3 Моделирование следящей САУ ДПТ по углу

Следящие системы широко используются при построении систем управления положением различных механизмов. Следящие системы по углу поворота ДПТ существенно менее быстродействующие, чем САУ по угловой скорости.

На рисунках 3.11, 3.12 и 3.13 приведены схемы результаты моделирования систем стабилизации угла поворота ДПТ.

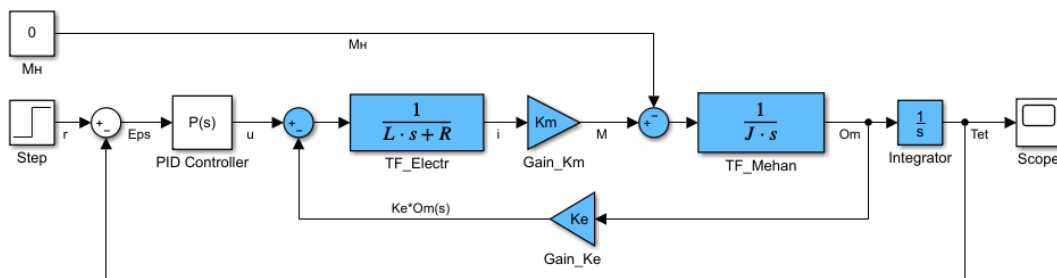
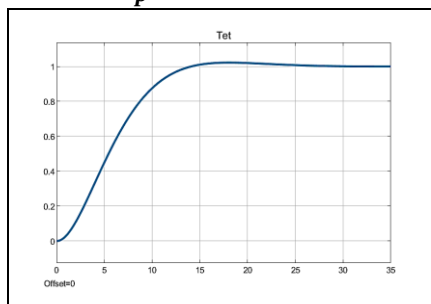


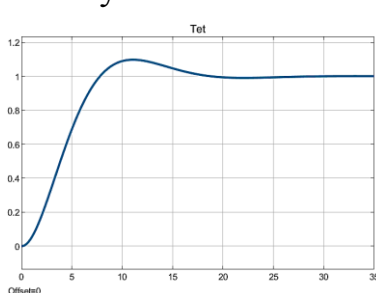
Рис. 3.11 – Simulink-схема САУ угла ДПТ с П-регулятором

Как и в предыдущем пункте, здесь рассматривается поведение САУ с двумя видами регуляторов – П и ПД. Моделирование САУ с П-регулятором прово-

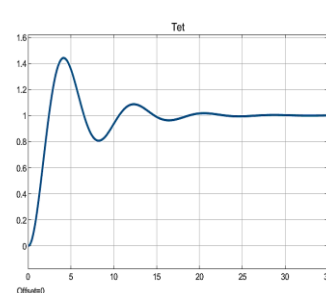
дилось для трёх значений коэффициента $k_p = \{0.06; 0.1; 0.5\}$. С увеличением коэффициента усиления регулятора увеличивается перерегулирование и при значении $k_p \approx 8$ система становится неустойчивой.



а) $k_p=0,06$



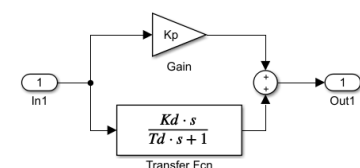
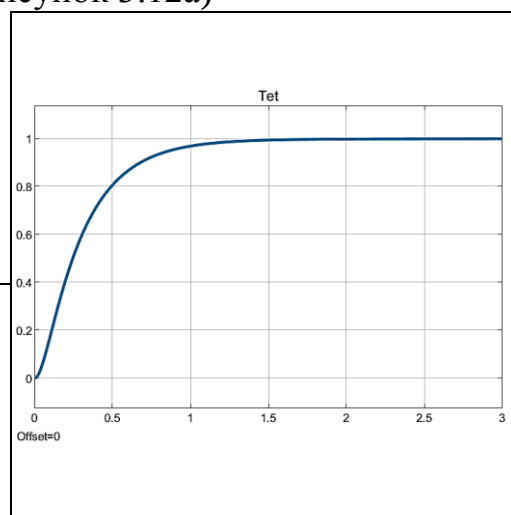
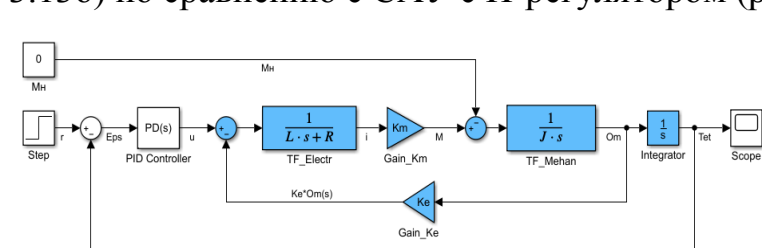
б) $k_p=0,1$



в) $k_p=0,5$

Рис. 3.12 – Результаты моделирования САУ угла ДПТ с П-регулятором

Качество работы САУ с ПД-регулятором в прямой цепи существенно выше. Время переходного процесса уменьшается примерно на порядок (рисунок 3.13б) по сравнению с САУ с П-регулятором (рисунок 3.12а)



$$W_p(s) = k_p + \frac{k_d s}{T_d s + 1}$$

$$k_p = 1,03; k_d = 2,46; T_d = 2,85 \cdot 10^{-3}$$

а) Simulink-схема с ПИД-регулятором

б) график выхода $\theta(t)$

Рис. 3.13 – Моделирования САУ угла ДПТ с ПИД-регулятором

4 ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

Первая часть работы:

а) ознакомиться с содержанием второго раздела методических указаний и выполнить все иллюстративные примеры, приведенные в нём;

б) при выполнении примеров необходимо научиться использовать различные источники и приёмники сигналов; формировать векторные сигналы из скалярных и извлекать скалярные сигналы из векторных; изменять параметры моделирования – время моделирования, процедуру интегрирования, точность расчётов и т.д.

Вторая часть работы (варианты заданий приведены в таблице 4.1):

Варианты заданий с параметрами двигателей

№ вар.	R Ом	L мГн	k_e В·с	J кг·м ²	k_m Н·м/А
1	1.48	91,5	0,343	0.06	0,250
2	0.762	56,9	0,344	0.08	0,259
3	0.945	45,9	0,341	0.14	0,261
4	0.546	31,8	0,343	0.16	0,268
5	0.585	25,2	0,340	0.36	0,260
6	0.561	14,1	0,332	0.8	0,320
7	0.50	14.0	0.30	0.85	0.310

а) сформировать Simulink-модель ДПТ по скорости и провести моделирование отдельно при управляющем воздействии, отдельно при возмущающем воздействии и при их совместном действии. Выполнить данный пункт для двух форм моделей ДПТ в виде ПФ в соответствии с рисунками 3.2 и 3.4. Графически сравнить результаты моделирования;

б) провести моделирование САУ скоростью ДПТ с П- и ПИД-регулятором (рисунки 3.8 и 3.10). Определить характер реакций системы на входные воздействия по графикам, построенным на основе данных, переданных из пакета Simulink в рабочую область Matlab (Workspace) через блок "To Workspace" библиотеки "Sinks";

в) провести моделирование следящей САУ с ПД-регулятором по углу по схеме рисунка 3.13. В качестве задающего воздействия принять угол в 5 градусов. Определить время отработки задающего воздействия. Проанализировать влияние внешнего возмущения M_H на качество и точность отработки задающего воздействия;

г) привести последовательность графо-аналитических (структурных) преобразований и получить ПФ рассматриваемой САУ по управлению и возмущению. Решить аналогичную задачу с помощью средств Matlab;

д)* построить Simulink-модель ДПТ в форме пространства состояний на основе уравнений (3.9). Уравнение выхода (3.10) сформировать с учетом возможности измерения всех переменных состояния: угла поворота $\theta(t)$ вала, угловой скорости вращения $\omega(t)$ вала и тока в якорной цепи $i(t)$. Управляющее $u(t)$ и возмущающее $M_H(t)$ воздействия организовать в один вектор внешних воздействий. Провести моделирование следящей САУ ДПТ по углу (рисунок 3.13), начальные условия принять нулевыми. Дополнительно провести моделирование данной САУ и построить собственное движение системы для случая начального отклонения по углу, равного 2 градусам.

Для синтеза (настройки параметров) регуляторов рекомендуется использовать блок PID Controller  из библиотеки Simulink>Continuous.

* Данный пункт задания выполняется факультативно.

Данный блок можно использовать в режиме самонастройки с интерактивными инструментами выбора типа регулятора и визуальной оценки качественных показателей системы управления.

5 СОДЕРЖАНИЕ ОТЧЕТА

Отчет по лабораторной работе должен содержать: цель работы; краткие теоретические сведения в части построения различного вида математических моделей линейных стационарных САУ; все необходимые предварительные соотношения, расчеты, графические построения, Matlab-программы и Simulink-схемы; результаты работы – схемы, числовые и графические данные; подробные выводы по каждому пункту и общие выводы по работе в целом; библиографический список реально использованных источников.

6 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Пояснить порядок создания Simulink-модели; перечислить основные глобальные параметры моделирования; указать способы их настройки.
2. Перечислить основные блоки-источники сигналов и их характеристики.
3. Перечислить основные блоки-приемники сигналов и их характеристики.
4. Каким образом вывести несколько графиков в одно окно блока Scope?
5. Каким образом можно построить несколько графиков в отдельных окнах одного блока Scope?
6. Привести и пояснить Simulink-блоки для задания LTI-систем.
7. Пояснить механизм взаимодействия рабочей области системы MATLAB и Simulink-модели.
8. Каким образом можно автоматизировать вычисление параметров для Simulink-модели и передачу их значений из рабочей области MATLAB в Simulink-схему?
9. Перечислите возможности внешнего оформления Simulink-блоков.
10. По модели "вход-выход" (3.2) построить модель ДПТ в пространстве состояний, выбрав в качестве компонентов вектора состояния фазовые переменные – $\theta(t)$, $\dot{\theta}(t)$ и $\ddot{\theta}(t)$.
11. При каком соотношении между постоянными времени T_γ и T из выражения (3.6) модель ДПТ будет иметь колебательные свойства?
12. Каким по величине следует выбирать сопротивление цепи якоря R , чтобы при прочих заданных параметрах, модель ДПТ имела свойства аperiodического звена?
13. Для модели следящей САУ по углу (рисунок 3.13) получить передаточную функцию замкнутой системы по задающему воздействию.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Лазарев Ю.Ф. Моделирование процессов и систем в Matlab. Учебный курс /Ю.Ф. Лазарев. – СПб.: Питер; К.: Издательская группа BHV, 2005. – 512 с.
2. Перельмутер В.М. Пакеты расширения Matlab. Control System Toolbox и Robust Control Toolbox / В.М. Перельмутер. – М.: СОЛОН-ПРЕСС, 2008. – 224 с.
3. Герман-Галкин С.Г. Matlab & Simulink. Проектирование мехатронных систем на ПК / С.Г. Герман-Галкин. – СПб.: КОРОНА-Век, 2008. – 368 с.
4. Бесекерский В.А. Теория систем автоматического управления / В.А. Бесекерский, Е.Н. Попов. – 4-е изд. – СПб.: "Профессия", 2003. – 752 с.
5. Филлипс Ч. Системы управления с обратной связью /Ч. Филлипс, Р. Харбор. – М.: Лаборатория базовых знаний, 2001. – 616 с.
6. Гайдук А.Р. Теория автоматического управления в примерах и задачах с решениями в MATLAB: уч. пособие / А.Р. Гайдук, В.Е. Беляев, Т.А. Пьявченко. – СПб.: Лань, 2017. – 464 с. – URL: <https://e.lanbook.com/book/90161>

Учебное издание

ИССЛЕДОВАНИЕ САУ СРЕДСТВАМИ ПАКЕТА SIMULINK

Методические указания

Составители:

Карапетьян Валерий Артёмович,
Крамарь Вадим Александрович

В авторской редакции

Изд. № 111/2020. Объем 1,75 п.л.
РИИЦМ ФГАОУ ВО "Севастопольский государственный университет"