



АУЭС

Образован в 1975

**Некоммерческое
акционерное
общество**

**АЛМАТИНСКИЙ
УНИВЕРСИТЕТ
ЭНЕРГЕТИКИ И
СВЯЗИ имени
Гумарбека Даукеева**

**Кафедра Автоматизации
и управления**

МОДЕЛИРОВАНИЕ И ИДЕНТИФИКАЦИЯ ОБЪЕКТОВ УПРАВЛЕНИЯ

Методические указания к выполнению курсовой работы
для студентов образовательной программы
6В07108 – «Автоматизация и управления»

—

Алматы 2022

СОСТАВИТЕЛЬ: Ибраева Л.К. Моделирование и идентификация объектов управления. Методические указания к выполнению курсовой работы для студентов образовательной программы 6В07108 – «Автоматизация и управления» – Алматы: АУЭС, 2022.

Методические указания предназначены для выполнения курсовой работы по дисциплине «Моделирование и идентификация объектов управления». В курсовой работе исследуется задача динамики каскада резервуаров.

Цель работы: привитие навыков получения физико-математической модели объекта управления, анализа динамических характеристик объекта, навыков моделирования объекта управления на ЭВМ.

Математическая модель объекта в виде системы дифференциальных уравнений выводится на основе аналитических методов моделирования. Для имитационного моделирования поведения рассматриваемой системы и моделирования управления объектом применяются пакет моделирования динамических систем Simuink и пакет проектирования событийно-управляемых систем Stateflow системы MatLab.

Содержание

Введение	4
1 Задание на курсовую работу	5
2 Описание объекта исследования	6
3 Алгоритм работы системы	7
4 Составные блоки подсистемы двух баков	8
5 Реализация модели в среде Simulink программы MatLab	8
6 Управление объектом	10
6.1 Подсистема Stateflow пакета Simulink	10
6.2 Создание диаграммы контроллера	14
7 Проведение имитационных экспериментов на модели	17
8 Варианты заданий	17
9 Требования к отчету по работе	
10 Контрольные вопросы	18
Список литературы	20

Введение

В курсовой работе исследуется задача динамики каскада резервуаров. Математическая модель объекта в виде системы дифференциальных уравнений выводится на основе аналитических методов моделирования.

Для имитационного моделирования поведения рассматриваемого объекта и управления объектом применяются пакеты Simulink и Stateflow системы MatLAB.

Пакет моделирования динамических систем Simulink предназначен для математического моделирования линейных и нелинейных динамических систем и устройств, представленных своей функциональной блок-схемой, именуемой моделью. Для построения функциональной блок-схемы моделируемых устройств Simulink имеет обширную библиотеку блочных компонентов и удобный редактор блок-схем. Последний основан на использовании графического интерфейса пользователя и, по существу, является средством визуального программирования. Используя палитры компонентов (наборы) блок-схем, пользователь с помощью мыши переносит нужные компоненты на рабочий стол пакета и соединяет линиями входы и выходы блоков. Таким образом создается блок-схема модели.

Simulink автоматизирует следующий наиболее трудоемкий этап моделирования: он составляет и решает сложные системы алгебраических и дифференциальных уравнений, описывающих заданную функциональную схему (модель), обеспечивая удобный и наглядный визуальный контроль за поведением созданного пользователем виртуального устройства. Ценность Simulink заключается и в обширной библиотеке компонентов (источники сигналов с любыми временными зависимостями, преобразователи с разнообразными формами передаточных характеристик, интегрирующие и дифференцирующие блоки и т.д.). Также в библиотеке имеется набор регистрирующих устройств.

Важное достоинство пакета - возможность задания в блоках произвольных математических выражений. Как программное средство Simulink - типичный представитель визуально-ориентированного языка программирования. На всех этапах работы пользователь не имеет дела с обычным программированием. Программа автоматически генерируется в процессе ввода выбранных блоков компонентов, их соединений и задания параметров компонентов.

Пакет проектирования событийно-управляемых систем Stateflow представляет собой графический инструмент для проектирования сложных систем управления и является самым значительным дополнением к среде Simulink. Он дает возможность моделировать поведение сложных событийно-управляемых систем.

Процесс создания модели системы управления в Stateflow обычно подразумевает следующие этапы:

- создание новой модели управляемого объекта в Stateflow или использование уже существующей модели;
- создание диаграммы в Stateflow;
- добавление к Stateflow-блоку интерфейса событий и данных;
- отладка модели;
- генерация кода.

Для создания Stateflow-диаграммы необходимо сначала создать модель в Simulink или открыть уже существующую. Затем нужно выбрать в библиотеке блок, обозначающий диаграмму Stateflow, и перенести его в свою модель. После двойного щелчка по нему мышью, откроется окно Stateflow (Chart), в котором, пользуясь меню и панелью инструментов, можно описывать сами состояния, связи между ними, определять переменные, условия переключения и другие, необходимые для управления элементы.

1 Задание на курсовую работу

1.1 Ознакомиться с объектом исследования. Обосновать вывод уравнений математической модели объекта.

1.2 Определить перечень входных и выходных переменных математической модели. Разработать подробный алгоритм работы объекта.

1.3 Для исследования поведения рассматриваемой динамической системы использовать пакет Simulink системы MatLab. Собрать блок-диаграмму модели в пакете Simulink в соответствии с логикой работы системы.

1.4 Используя графический инструментарий для проектирования систем управления Stateflow, провести моделирование управления объектом.

1.5 Провести имитационные эксперименты для исследования поведения рассматриваемой системы (параметры объекта выбираются по варианту).

1.6 Оформить отчет по работе.

2 Описание объекта исследования

Исследуемая система представляет собой два цилиндрических бака, расположенных вертикально на разной высоте таким образом, что дно первого бака находится на расстоянии $H=0,310$ м от дна второго (рис. 2.1). Баки имеют одинаковую высоту $h=1$ м и различные диаметры: первый – $D_1=12$ см, второй – $D_2=5$ см. Система имеет входную трубу, находящуюся в первом баке на расстоянии h от его дна. Баки соединены трубой, являющейся выходной трубой первого бака (и расположенной у самого его дна) и входной трубой второго бака (расположенной на расстоянии H от его дна).

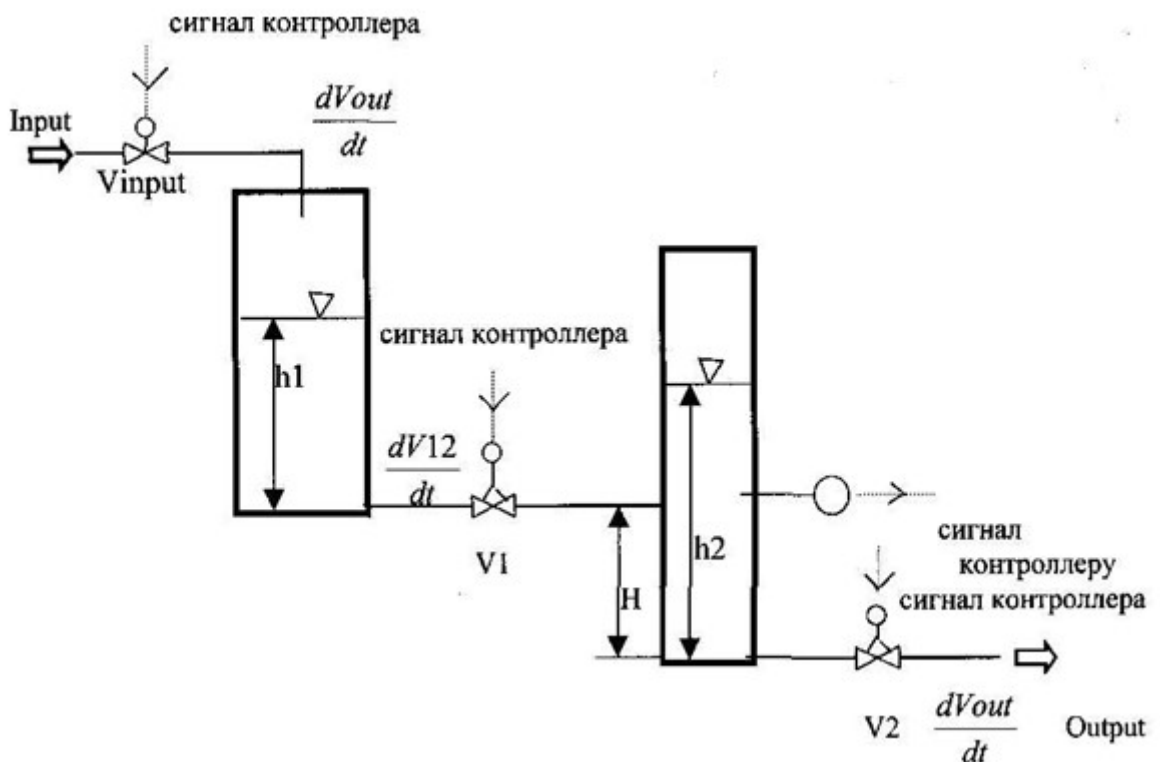


Рисунок 1 - Система двух баков

Также система имеет выходную трубу, расположенную у самого дна второго бака. Входная труба системы снабжена входным краном V_{input} , который открывается мгновенно, и скорость входного потока воды определяется как (л/ч)

$$\frac{dV_{input}}{dt} = \begin{cases} 0, & \text{если } V_{input} \text{ закрыт,} \\ 400, & \text{если } V_{input} \text{ открыт} \end{cases} \quad (1)$$

Краны V_1 и V_2 являются медленными устройствами, они открываются и закрываются с одной и той же постоянной скоростью, так что от момента начала открытия (закрытия) до полного открытия (закрытия) требуется 80 с. Их

открытие и закрытие контролируется задвижкой, меняющей свое положение от значения $P=0$ (полное закрытие в условных единицах) до $P=80$ (полное открытие).

Управление открытием/закрытием кранов V_{input} , V_1 и V_2 осуществляется специальным контроллером.

Если через A_1 и A_2 обозначить площади оснований баков, то система уравнений для уровней воды в баках h_1 и h_2 запишется так

$$\begin{aligned} \frac{dh_1}{dt} &= \frac{1}{A_1} \left(\frac{dV_{input}}{dt} - \frac{dV_{12}}{dt} \right); \\ \frac{dh_2}{dt} &= \frac{1}{A_2} \left(\frac{dV_{12}}{dt} - \frac{dV_{out}}{dt} \right), \end{aligned} \quad (2)$$

где $\frac{dV_{12}}{dt}$ - скорость протекания воды по трубе между баками,

$\frac{dV_{out}}{dt}$ - скорость вытекания воды из системы.

Скорость протекания воды между баками зависит от уровней воды h_1 и h_2 , значения H и положения задержки P_1 в кране V_1 :

$$\frac{dV_{12}}{dt} = \begin{cases} K_1(P_1) \cdot \sqrt{h_1 - (h_2 - H)}, & h_2 > H; \\ K_1(P_1) \cdot \sqrt{h_1}, & h_2 \leq H. \end{cases} \quad (3)$$

Скорость вытекания воды из системы зависит от уровня воды во втором баке h_2 и положения задвижки P_2 на кране V_2 :

$$\frac{dV_{out}}{dt} = K_2(P_2) \cdot \sqrt{h_2} \quad (4)$$

Индивидуальные свойства кранов определяются функциями:

$$K_1(P_1) = \begin{cases} 1.85 \cdot 10^{-4} \cdot e^{-6 \cdot 10^{-6} \cdot P_1^3}, & 0 \leq P_1 < 80; \\ 0, & P_1 = 80. \end{cases} \quad (5)$$

$$K_2(P_2) = \begin{cases} 2.26 \cdot 10^{-4} \cdot e^{-5.7 \cdot 10^{-6} \cdot P_2^3}, & 0 \leq P_2 < 80; \\ 0, & P_2 = 80. \end{cases} \quad (6)$$

3 Алгоритм работы системы

Работа всей системы описывается следующим алгоритмом.

В исходном состоянии все краны закрыты и оба бака пусты. В процессе заполнения баков, уровень воды во втором баке должен удовлетворять

следующему требованию: он не должен опускаться ниже установленного уровня L_{\min} и превышать также заранее заданный уровень L_{plus} .

В начальный момент контроллер посылает сигнал входному крану V_{input} , тот мгновенно открывается и в течение времени Time1 [с] наполняется только первый бак.

По истечении времени Time1 контроллер посылает команду открыть кран V_1 , и вода начинает поступать во второй бак. Второе состояние сохраняется на протяжении Time2 [с].

По истечении времени Time2 начинает контролироваться положение крана V_2 . А именно, если контроллер обнаруживает, что уровень воды во втором баке опустился ниже значения L_{\min} [м], поступает команда закрыть выходной кран, если вода во втором баке превышает уровень L_{plus} [м] - выдается команда открыть выходной кран.

Аварийными считаются ситуации, когда переполняется один из баков или происходит периодическое открытие и закрытие выходного крана.

Нормальным режимом системы считается состояние, когда все краны открыты, и вода протекает через систему с постоянной скоростью.

Сигналы о значениях V_{input} , V_1 , V_2 подаются от контроллера. Эти переменные являются входами модели.

4 Составные блоки подсистемы двух баков

Система двух баков должна реагировать на сигналы контроллера, а именно: управлять открытием/закрытием кранов, а также отслеживать ситуацию, когда вода во втором баке поднимается выше уровня H , и при возникновении такой ситуации изменять скорость поступления воды во второй бак. Также система должна отслеживать ситуацию переполнения баков и должным образом на нее реагировать.

Исходя из этих требований, можно провести декомпозицию системы на несколько частей, а именно:

- выделить элемент, отвечающий за открытие/закрытие крана V_{input} ; назовем эту подсистему (реализуется блоком *Subsystem*) $V_{\text{in_Control}}$;
- выделить элемент, отвечающий за открытие/закрытие выходного крана; назовем эту подсистему V_{out} ;
- выделить элемент, управляющий открытием/закрытием задвижек P_1 и P_2 ; это будут подсистемы $K1_Control$ и $K2_Control$;
- в качестве еще одного элемента системы можно выделить сами два бака, на которые возлагается задача отслеживания аварийной ситуации переполнения баков и ситуации, когда вода во втором баке поднимается выше уровня H .

5 Реализация модели в среде Simulink программы MatLab

Чтобы построить диаграмму модели, необходимо выделить в системе отдельные относительно независимые элементы и определить, каким образом они взаимосвязаны между собой.

В рассматриваемой задаче в первую очередь можно выделить два основных элемента: саму систему из двух баков и устройство, управляющее открытием/закрытием кранов (контроллер). В первую очередь необходимо создать блок-диаграмму системы из двух баков.

В дальнейшем все блок-диаграммы модели, содержащие несколько составных элементов, оформляются как подсистемы. Начнем с самых простых уравнений.

Подсистема определения скорости входного потока. Для определения скорости входного потока используется одна из формул уравнения (1) в зависимости от значения сигнала контроллера V_{input} .

Подсистема для такого расчета содержит в себе функциональную схему, состоящую из стандартного блока Switch и двух экземпляров стандартного блока Const, содержащих в себе значения, соответствующие скорости входного потока воды в систему двух баков, когда входной бак открыт и закрыт. Переключение между этими значениями происходит в блоке Switch в зависимости от значения, поступающего на вход блока от контроллера. Соответствующее значение выходного сигнала подается на выход.

После реализации требуемой диаграммы по уравнению 1, необходимо ее оформить в виде подсистемы (рисунок 2).



Рисунок 2 – Блок подсистемы определения скорости входного потока

Эта подсистема имеет один вход и один выход. Сигнал на вход подсистемы поступает с контроллера. В зависимости от сигнала кран открыт, либо закрыт.

Подсистема определения скорости вытекания воды из системы.

Для определения скорости вытекания воды из системы используем формулу (4). Чтобы рассчитать V_{out} , необходимо вначале вычислить функцию $K_2(P_2)$. Эта функция рассчитывается по формулам (6) в зависимости от величины, поступающей на вход подсистемы от контроллера. В выражении (4) помимо функции $K_2(P_2)$ используется текущее значение h_2 . То есть, полученные после интегрирования дифференциальных уравнений значения уровней в баках необходимо подать на вход подсистемы расчета правых частей этих уравнений.

Диаграммы расчетов $K_2(P_2)$ и h_2 будут рассмотрены ниже.

После реализации требуемой диаграммы оформите ее, как подсистему рисунка 3. На вход подсистемы поступают значения K_2 , h_2 . В подсистеме вычисляется значение скорости вытекания воды из системы.



Рисунок 3 – Блок подсистемы для определения скорости вытекания воды из системы

На вход блока V_{out} поступают значения K_2 и h_2 . Для объединения этих значений в один вектор используется стандартный блок Mux . С этого блока вектор поступает на вход стандартного блока Fcn , в котором происходит вычисление выходной величины V_{out} (формула (4)).

Блок V_{out} имеет два входа и один выход.

Подсистемы управления краном между баками (Рисунки 4 и 5).

Чтобы рассчитать скорость протекания воды между баками используются формулы (3) - два варианта выражений для двух возможных ситуаций ($h_2 > H$; $h_2 \leq H$). Предварительно необходимо вычислить функцию $K_1(P_1)$. Эта функция рассчитывается по формулам (5) в зависимости от величины, поступающей на вход подсистемы от контроллера. В выражении (3) помимо функции $K_1(P_1)$ используются текущие значения h_1 и h_2 . То есть, полученные после интегрирования дифференциальных уравнений значения уровней в баках необходимо подать на вход подсистемы расчета правых частей этих уравнений.

Создаются две подсистемы $V12(1)$ и $V12(2)$ для вычисления выходной величины $V12$ - формула (3).

Составной блок $V12(1)$ имеет функциональную схему, состоящую из стандартного блока Mux , объединяющего в один вектор значения $K1$, h_1 и h_2 , которые поступают на его вход, и подающего этот вектор на вход стандартного блока Fcn , в котором происходит вычисление выходной величины $V12$. Блок $V12(1)$ имеет три входа и один выход.

Реализованная диаграмма оформляется в виде следующей подсистемы (рисунок 4).



Рисунок 4 – Блок подсистемы определения скорости воды между баками при $h_2 > H$

Составной блок $V12(2)$ имеет функциональную схему, состоящую из стандартного блока Mux , который объединяет в один вектор значения $K1$ и h_1 ,

поступающие на его входы, и подающего этот вектор на вход стандартного блока F_{sp} , в котором происходит вычисление выходной величины V_{12} (оформите реализованную диаграмму в виде подсистемы рисунка 5).

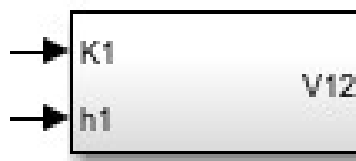


Рисунок 5 - Блок подсистемы определения скорости воды между баками при $h_2 < H$

Подсистемы управления краном между баками и выходным краном.

Блоки, где рассчитываются $K_1(P_1)$ и $K_2(P_2)$, назовем $K1_Control$ и $K2_Control$ – это окончательные подсистемы.

Они имеют идентичные функциональные схемы, различающиеся только значением коэффициентов в уравнении - формулы (5). Функциональная схема каждого из них состоит из следующих элементов:

- двух экземпляров стандартного блока $Switch$, один из которых необходим для переключения между положениями кранов V_1/V_2 открыто/закрыто, а второй для отслеживания ситуации, когда параметр P становится равным 80, и соответствующего переключения между значениями функции $K(P)$;

- двух экземпляров стандартного блока $Const$, содержащих в себе значения 1 и -1, эти значения необходимы для выполнения интегрирования, и еще один блок $Const$ для присвоения значения 0 функции $K(P)$ в ситуации, когда $P \geq 80$;

- стандартного блока $Integrator$, интегрирующего поступающее от переключателя значение (1 или -1) в пределах от 0 до 80 с начальным значением интегрируемой величины 80 (выход этого блока поступает на вход второго переключателя);

- стандартного блока F_{sp} , в котором происходит вычисление значения функции $K_1(P)/K_2(P)$.

Переключения в блоке $Switch$ происходят в зависимости от величины, поступающей на вход блока $K1_Control/K2_Control$ от контроллера.

Соответствующее значение K_1/K_2 подается на выход.

В окончательном виде схема представляется в виде подсистем $K1_Control/ K2_Control$, которые имеют один вход и один выход.

На вход подсистемы $K1_Control$ поступает сигнал от контроллера. В зависимости от этого сигнала кран либо открывается, либо закрывается. Открытие крана будем моделировать положительной константой (+1), а закрытие моделируем отрицательной константой (-1).

Диаграмма подсистемы $K1_Control$ строится следующим образом.

Если от контроллера поступает значение 1, в блоке $Switch$ оно сравнивается с «0». Так как $1 > 0$, то со $Switch$ на интегратор должно поступать

«-1» (объясните: почему?). Это необходимо для плавного открытия и закрытия кранов. Начальное значение интегратора равно 80. Далее в блоке Fcn вычисляется значение $K1(P1)$. Значение $K1(P1)$ поступает на выход, если значение на выходе интегратора меньше 80. Если значение на выходе интегратора больше или равно 80, на выход поступает 0.

Если от контроллера поступает (-1), в блоке Switch оно сравнивается с «0». Так как $-1 < 0$, на интегратор должно поступать «1» (объясните: почему?). Начальное значение интегратора равно 80. Далее в блоке Fcn вычисляется значение $K1(P1)$. Значение $K1(P1)$ поступает на выход, если значение на выходе интегратора меньше 80. Если значение на выходе интегратора больше или равно 80, на выход поступает 0.

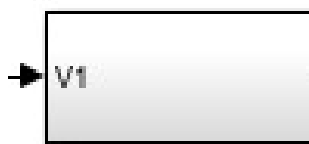


Рисунок 6 – Блок подсистемы $K1_Control$ управления первым краном

Аналогично стоит диаграмма подсистемы $K2_Control$, на вход которой поступает сигнал от контроллера. В зависимости от этого сигнала кран либо открывается, либо закрывается.

Если от контроллера поступает +1, в блоке Switch оно сравнивается с «0». Так как $(+1 > 0)$, то со Switch на интегратор должно поступать «-1». Начальное значение интегратора равно 80. Далее в блоке Fcn вычисляется значение $K2(P2)$. Значение $K2(P2)$ поступает на выход, если значение на выходе интегратора меньше 80. Если значение на выходе интегратора больше или равно 80, на выход поступает 0.

Если от контроллера поступает (-1), в блоке Switch оно сравнивается с «0». Так как $(-1 < 0)$, на интегратор должно поступать «1». Начальное значение интегратора равно 80. Далее в блоке Fcn вычисляется значение $K2(P2)$. Значение $K2(P2)$ поступает на выход, если значение на выходе интегратора меньше 80. Если значение на выходе интегратора больше или равно 80, на выход поступает 0.



Рисунок 7 – Блок подсистемы $K2_Control$ управления вторым краном

Подсистема расчета уровней жидкости в двух резервуарах.

Основными уравнениями модели является система (2) - система двух дифференциальных уравнений первого порядка для определения выходных переменных модели – h_1 и h_2 .

Для преобразования этих уравнений в блок-диаграмму используется следующая идея. Вначале формируются правые части уравнений.

Проинтегрировав эти выражения, получаем значения h_1 и h_2 . Полученные величины теперь можно использовать для формирования правых частей уравнений (обратной связью).

Сформируем подсистему расчета уровней жидкости в двух резервуарах, назовем ее `Two_Tanks`. Эта подсистема имеет входные переменные k_1 , k_2 и V_{in} , представляющие соответственно значения функций, управляющих кранами V_1 и V_2 , а также скорость входного потока воды в систему. Эти переменные поступают в рассматриваемую подсистему с выходов блоков, в которых они рассчитывались.

Выходные переменные - уровни воды в первом и втором баках h_1 и h_2 . Значения этих переменных рассчитываются по формулам (2). Правые части этих уравнений содержат величины V_{out} и V_{12} , закон изменения которых задается формулами (3) и (4). Следовательно, надо использовать подсистемы для вычисления этих переменных: V_{out} , $V_{12}(1)$ и $V_{12}(2)$ (два блока, так как формула (3) имеет две ветви).

Составной блок `Two_Tanks` имеет функциональную схему, образованную из следующих элементов:

- составных блоков V_{out} , $V_{12}(1)$ и $V_{12}(2)$;
- стандартного блока `Switch`;
- двух экземпляров стандартного блока `Мих`, объединяющих в векторы соответствующие значения - V_{in} (поступающее на вход блока `Two_Tanks` с выхода блока `Vin Control`) и V_{12} , а так же V_{12} и V_{out} ;
- блоков `Fсп`, в которых вычисляются правые части уравнений - формулы (2);
- двух экземпляров стандартного блока `Integrator`, выходами которых являются значения h_1 и h_2 , поступающие на выход.

Переключение в блоке `Switch` происходит в зависимости от значения h_2 , поступающего на вход блока `Switch` от интегратора. При этом подключается один из блоков, вычисляющих V_{12} .

В окончательном виде диаграмму надо оформить в виде подсистемы (рисунок 8).

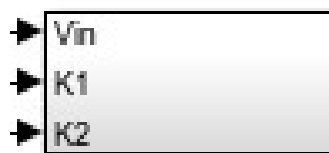


Рисунок 8 – Подсистема расчета уровней жидкости в двух резервуарах `Two_Tanks`

Окончательная диаграмма объекта исследования.

Блоки `Two_Tanks`, `Vin_Control`, `K1_Control` и `K2_Control`, соединенные между собой функциональными связями, составляют подсистему

Tank_System. Она имеет три входа, на которые подаются сигналы контроллера (V_{input} , V_1 , V_2), и два выхода – $h1$ и $h2$.

Составной блок Tank_System, и по два экземпляра стандартных блоков Const, Relational Operator и Stop Simulation (выполняющих проверку на переполнение баков), соединенных соответствующими функциональными связями, образуют подсистему Tank_System_Block.

Блок Tank_System_Block имеет три входа (на которые подаются сигналы от контроллера) и два выхода ($h1$ и $h2$). Эта подсистема является блок-диаграммой нашего объекта исследования.

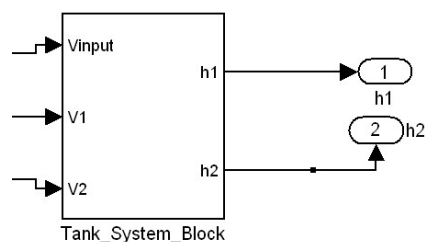


Рисунок 10 –Окончательная диаграмма модели

6 Управление объектом

6.1 Подсистема Stateflow пакета Simulink

Для описания поведения контроллера используется графический инструмент Stateflow. В основе Stateflow лежит специальная форма представления гибридного поведения – карта состояний. Основные элементы карты состояний – это состояние (state) и переход (transition).

Для создания Stateflow-диаграммы необходимо открыть модель, созданную в Simulink. Затем нужно выбрать в библиотеке блок, обозначающий диаграмму Stateflow, и перенести его в свою модель. После двойного щелчка по нему мышью откроется окно Stateflow (chart), в котором, пользуясь меню и панелью инструментов, можно описывать сами состояния, связи между ними, определять переменные, условия переключения и другие, необходимые для управления элементы.

На диаграмме Stateflow различают две основные группы элементов: графические и неграфические. Все графические элементы диаграммы представлены на левой панели основного окна - Stateflow (chart), графического редактора карт состояний. Это – state (состояние), default transition (переход по умолчанию), history junction (переход в последнее активное состояние) и connective junction (соединительный переход). Неявно на диаграмме задается еще один графический элемент – transition (переход). Состояния могут быть простыми и сложными (то есть имеющими иерархическую структуру), они также могут объединяться в цепочки состояний, функционирующих параллельно (и независимо друг от друга) и последовательно. В число графически непредставимых элементов входят data (переменные) и event (событие).

Каждое состояние описывает один режим работы событийно-управляемой системы. Состояние становится активным, если получает значение “истина” условие перехода (срабатывает переход), ведущего к нему, или если это состояние является начальным. Каждое состояние на диаграмме Stateflow имеет «родителя», которым, по умолчанию, является сама диаграмма Stateflow.

Для создания графического образа состояния необходимо, выбрав на панели соответствующую кнопку, щелкнуть кнопкой мыши в том месте, где его необходимо разместить.

Состояние имеет текстовые метки, которые определяют действия, выполняемые во время его активности. Имя состояния вводится первым. При описании самого состояния могут быть определены следующие действия:

- entry – действие, выполняемое на входе в состояние;
- during- действие, выполняемое, пока состояние активно;
- exit – действие, выполняемое на выходе из состояния;
- on: {имя события}- действие, выполняемое в момент появления события (имя которого указано в угловых скобках) при условии, что система будет находиться в данном состоянии.

Переход – это объект, связывающий между собой два состояния. На диаграмме Stateflow он представляется стрелкой, начало которой относится к состоянию-источнику, а конец – к состоянию-адресату. Для того чтобы нарисовать переход, достаточно установив мышью у состояния-источника, нажать ее левую кнопку и, удерживая ее, протянуть появившуюся стрелку до состояния-адресата. Переход имеет метку, которая описывает условия срабатывания перехода и выполняемые при этом действия.

Переход по умолчанию – это переход с заданным адресатом, но без заданного источника. Его используют как аналог начального состояния, а также в сложных состояниях для того, чтобы определить, какое подсостояние становится активным, когда система попадает в данное сложное состояние

Переменные – это неграфические объекты на диаграмме Stateflow, предназначенные для хранения числовых данных. Переменные можно использовать на любом уровне иерархии. Для того чтобы создать входные (выходные и т. д.) переменные в блоке Stateflow, необходимо в окне соответствующей диаграммы Stateflow выбрать соответствующий пункт меню: Add/Data/Input from Simulink для создания входной, Add/Data/Output to Simulink для создания выходной и Add/Data/Local для создания локальной переменной - и в открывшемся диалоговом окне Data ввести имя переменной и другие ее характеристики.

6.2 Создание диаграммы контроллера

В рассматриваемой задаче контроллер посылает системе из двух баков сигналы, управляющие поведением кранов V_{input} , V_1 , V_2 . Для правильного управления системой баков ему необходимо иметь значения интервалов

времени $Time1$, $Time2$, уровней L_plus и L_min , а также получать от системы баков значение уровня воды во втором баке h_2 , необходимое для управления поведением выходного крана V_2 .

Работу системы вместе с контроллером представим как подсистему *System*, которая представляет собой составной блок (рисунок 10). Он содержит в себе функциональную схему, включающую диаграмму *Stateflow* (представленную блоком *Controller*, являющимся экземпляром стандартного блока *Chart* и описывающим поведение контроллера), и составной блок - подсистему определения уровней воды в баке *Tank_System_Block*, которые соединены соответствующими функциональными связями.

Также в блоке *System* присутствуют часы *Clock*, подающие системное время из *Simulink* в диаграмму *Stateflow*. Это объясняется тем, что при построении моделей, в которых используются блоки *Stateflow* (содержащие в себе переходы, инициируемые истечением неких временных интервалов), необходимо синхронизировать внутреннее системное время в *Stateflow* и в *Simulink*. Для этого необходимо на отдельный вход в блоке *Stateflow* подавать системное время из *Simulink*, и именно это время использовать при составлении условий переходов (таких, как истечение отрезков времени $Time1$ и $Time2$ в задаче о двух баках).

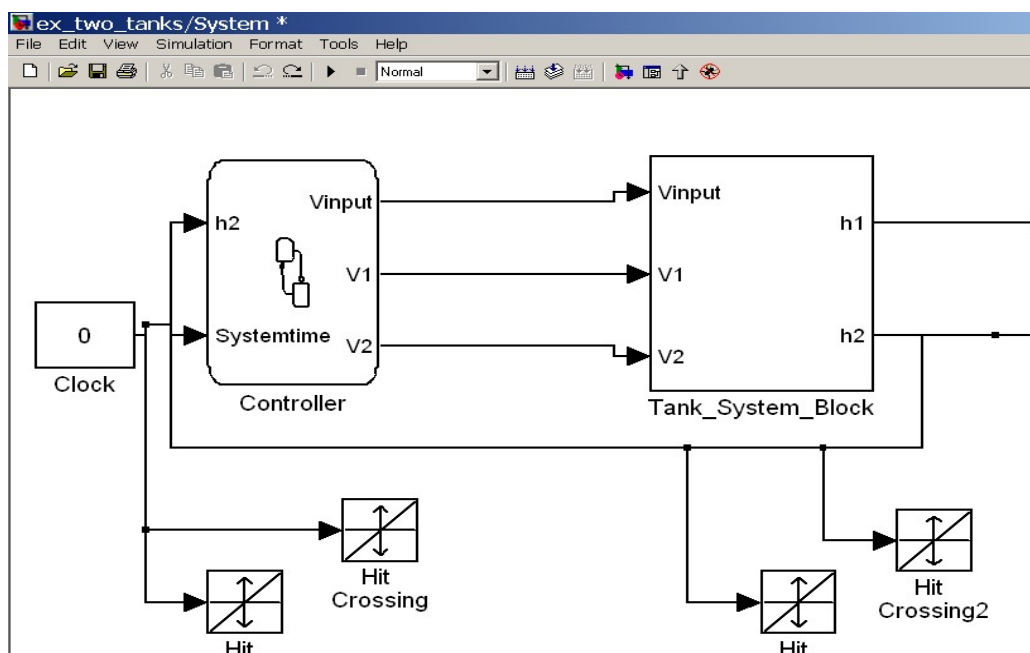


Рисунок 11 - Диаграмма подсистемы *System*

Диаграмма *Stateflow* отражает логику поведения блока *Controller*. Из исходной точки процесс мгновенно переходит в состояние *State1*, входным сигналом для этого состояния является сигнал *Vinput* на открытие крана на входе.

По истечении времени $Time1$ выполняется переход в состояние *State2*, сигналом на входе этого состояния является сигнал на открытие крана между

баками. Система находится в этом состоянии до тех пор, пока не выполнится условие $h_2 < L_{\min}$, означающее, что уровень воды во втором баке стал ниже значения L_{\min} . Если это условие выполнится, система переходит в состояние State4, входным сигналом этого состояния является подача сигнала на закрытие крана на выходе V2. Система находится в этом состоянии до тех пор, пока условие $h_2 > L_{\text{plus}}$ не будет выполнено, что означает, что уровень воды во втором баке вырос до значения L_{plus} . При выполнении условия система снова переходит в состояние State3.

К связи, соединяющей блоки Clock и Controller, подсоединяются два экземпляра блока Hit Crossing. Еще два экземпляра блока Hit Crossing подсоедините к связи блока Controller и выхода h_2 блока Tank_System_Block.

Использование блоков Hit Crossing необходимо для правильного выполнения переходов в диаграмме Stateflow, включенной в модель Simulink, в которой происходит непрерывное интегрирование. На вход блока Hit Crossing подается некая величина (в модели двух баков для двух блоков Hit Crossing это системное время, для еще двух - величина h_2). У первого блока Hit Crossing внутренней величиной является момент времени Time1, у второго – Time1 + Time2, у третьего - значение L_{plus} , у четвертого – L_{minus} .

Окончательный вид диаграммы модели объекта исследования в пакете Simulink можно представить в следующем виде (рисунок 11).

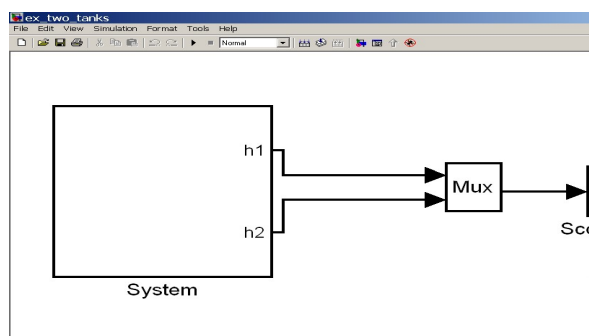


Рисунок 12 - Окончательный вид диаграммы модели

Блок System не имеет входов, но имеет два выхода – h_1 и h_2 , соединенные со стандартным блоком Mux, объединяющим их в вектор (h_1, h_2) . Последний связан со стандартным блоком вывода Scope, на который он подает вектор выходных величин (h_1, h_2) .

7 Проведение имитационных экспериментов на модели

7.1 Требуется подобрать параметры модели Time1, Time2, L_{plus} и L_{min} таким образом, чтобы смоделировать следующие ситуации в работе системы:

- а) режим нормальной работы системы;
- б) переполнение первого бака;
- в) переполнение второго бака;

г) периодическое открытие и закрытие выходного крана. Для этого случая отметить время первого и десятого переключения, а также значения уровня h_2 в эти моменты.

7.2 По результатам экспериментов заполнить следующую таблицу:

Ситуация	Time1 [с]	Time2 [с]	L plus [м]	L min [м]
а				
б				
в				
г				

8 Варианты заданий

№ варианта	h	H	D ₁	D ₂	V _{input}
1	0,7	0,27	8,4	3,5	280
2	0,5	0,2	6	2,5	200
3	0,625	0,25	7,5	3,125	250
4	0,8	0,48	10,6	4	320
5	0,875	0,34	10,5	4,375	350
6	0,75	0,210	10	3,75	300
7	1,25	0,48	15	6,25	500
8	1,125	0,44	1,5	5,625	450
10	1,5	0,58	18	7,5	600
10	1,375	0,54	16,5	6,875	550
11	1,1	0,585	13,2	5,5	440
12	1,2	0,42	14,4	6	480
13	0,6	0,27	8,0	4,5	270
14	0,5	0,3	5,6	2,8	200
15	0,65	0,35	7,0	3,5	250
16	0,8	0,5	10,5	4,2	300
17	0,85	0,35	10,0	4,75	320
18	0,75	0,25	10,5	3,5	310
110	1,5	0,5	10	6,5	400
20	1,15	0,45	8,0	5,5	250
21	1,5	0,5	15	7,0	200
22	1,35	0,5	14,5	6,75	50
23	1,3	0,55	12,2	5,8	420
24	1,4	0,4	14,0	6,2	430
25	1,2	0,3	7,0	5,0	230
26	1,8	0,5	17	6,0	210
27	1,25	0,7	14,0	6,5	70
28	1,35	0,50	12,0	5,10	400
210	1,5	0,6	14,6	6,3	470
30	0,75	0,4	10,0	5,0	315

9 Требования к отчету по работе

Отчет по работе должен содержать:

9.1 Описание объекта исследования курсовой работы; уравнения модели с объяснениями.

9.2 Перечень входных и выходных переменных модели.

9.3 Алгоритм работы системы.

9.4 Подробное описание реализации модели в среде Simulink программы MatLab:

- создание каждой диаграммы п.5 должно быть подробно объяснено (какое уравнение описывает рассматриваемый процесс, какие блоки используются и т.д.); необходимо в отчете разместить эту диаграмму;

- создание подсистем (рисунки 2-9) с подробными комментариями;

- подробная реализация диаграммы, находящейся в подсистеме рисунка 10 и объяснение создания самой этой подсистемы;

- объяснение создания подсистем 11,2

9.5 Описание создания блок-диаграммы контроллера.

9.6 Обоснование выбора параметров модели.

9.7 Результаты и анализ проведенных экспериментов.

10 Контрольные вопросы

10.1 В чем отличие аналитических методов моделирования от экспериментальных?

10.2 Какие законы сохранения используются для вывода уравнений модели?

10.3 К каким классам принадлежит рассматриваемая модель движения жидкости в каскаде резервуаров?

10.4 Какие переменные являются входами модели; выходами модели.

10.5 По истечении какого времени открывается кран между резервуарами?

10.6 В течение какого времени заполняется второй резервуар?

10.7 Какой инструментарий системы MatLab используется для моделирования управления объектом?

10.8 Какие переменные управляют поведением кранов в системе?

10.9 Какие блоки Stateflow используются для выполнения переходов?

10.10 Какой константой моделируется сигнал на закрытие кранов в системе?

10.11 Какой константой моделируется сигнал на открытие кранов в системе?

10.12 В течение какого времени открывается кран V_1 ?

10.13 Сколько входов и выходов у подсистемы $V_{in_Control}$?

10.13 В течение какого времени открывается кран \bar{V}_{input} ?

10.15 Сколько состояний установлено в модели Stateflow?

10.16 Сколько времени система находится в состоянии $state1$?

Список литературы

1. Ибраева Л.К. Моделирование объектов систем управления. Уч.пособие – Алматы:АУЭС,2018.
2. Ибраева Л.К., Хисаров Б.Д. Моделирование и идентификация объектов управления. Уч.пособие – Алматы:АУЭС,20010.
3. Кудинов Ю.И., Пащенко Ф.Ф.. Simulink : учеб.пособие.- СПб.: Лань – 2016.
4. Hunt Brian R., / R. Hunt Brian , L. Lipsman Roland. A Guide to Matlab : for Beginners and Experienced Users: updated for Matlab 8 and Simulink 8. - United Kingdom : Cambridge University Pres – 2014.
5. Васильев А.Н. Matlab. Практический подход. СПб. : Наука и Техника - 2012.