

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
имени академика С.П. КОРОЛЁВА»

А.В. КУЗНЕЦОВ, Г.М. МАКАРЬЯНЦ

**Математическое моделирование
системы автоматического управления давления в ёмкости**

Методические указания к лабораторной работе

Самара 2016

УДК: 004.942

Авторы: А.В. Кузнецов, Г.М. Макарьянц

Математическое моделирование системы автоматического управления давлением в ёмкости: методические указания к лабораторной работе / [А.В. Кузнецов, Г.М. Макарьянц]. – Самара: Изд-во Самар. ун-та, 2016. – 29 с.: ил.

В учебном пособии описана работа в программном пакете MatLab/Simulink. Приведена схема лабораторной установки. Описаны математические модели элементов установки и его системы управления. Рассмотрены используемые при моделировании блоки стандартной библиотеки Simulink. Приведена настройка ПИ-регулятора и fuzzy-регулятора и проведено их тестирование при управляющих и возмущающих воздействиях на модель объекта управления. Учебное пособие предназначено для студентов технических специальностей и направлений.

УДК: 004.942

© Самарский национальный
исследовательский университет, 2016

Содержание

1. Математическое описание.....	4
2. Моделирование в MatLab	5
2.1. Создание объекта управления	8
2.2. Создание системы управления	17
3. Задание	28
ПРИЛОЖЕНИЕ 1	29
Список использованных источников	30

1. Математическое описание

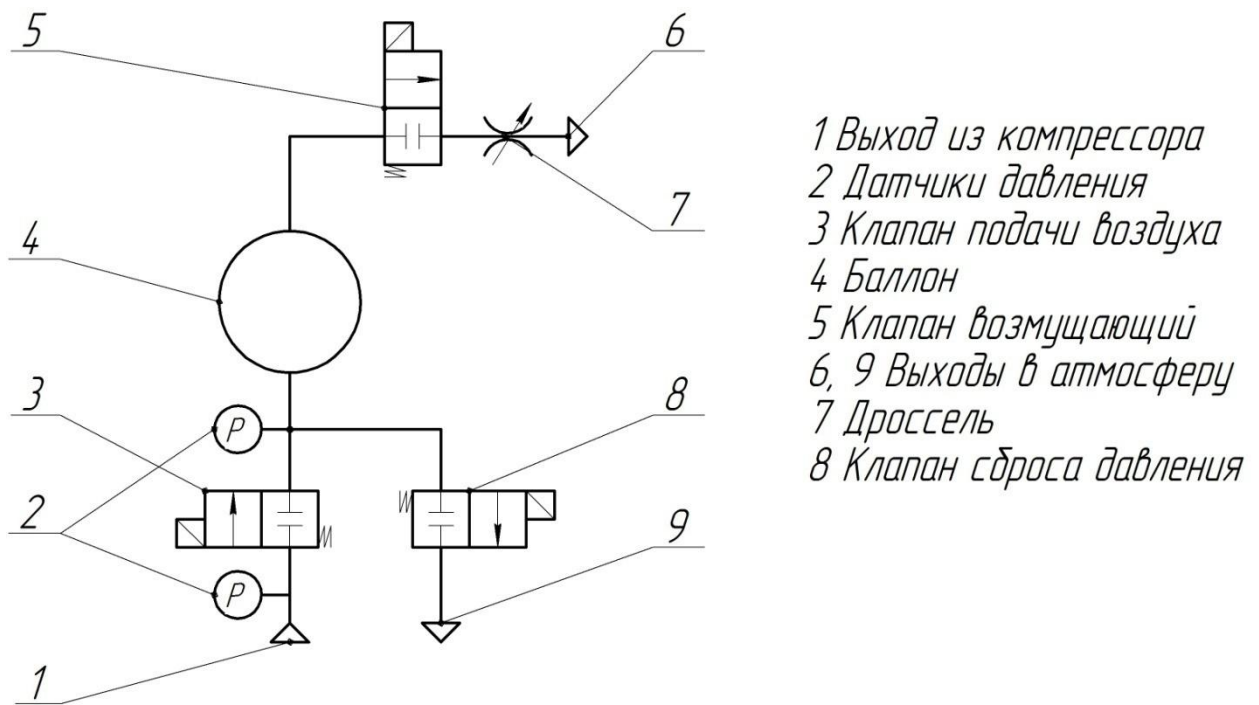


Рисунок 1 – Пневматическая схема установки

Входные и выходные условия:

- Показатель адиабаты, $k = 1,4$
- Удельная газовая постоянная, $R = 287 \text{ Дж}/(\text{кг}\cdot\text{К})$
- Температура воздуха, $T = 300 \text{ К}$
- Давление за компрессором, $p_k = 6 \text{ кПа}$
- Давление атмосферное, $p_a = 101325 \text{ Па}$

Объем баллона принимается равным $V = 0,0015 \text{ м}^3$, диаметр проходного сечения клапанов $d = 2,4 \text{ мм}$.

Центральным элементом лабораторной установки является баллон, в который компрессором нагнетается давление. Величина давления должна быть равной величине уставки и регулироваться системой управления при изменении уставки или при наличии возмущающего воздействия. Математическое описание баллона представляется собой дифференциальное уравнение первого порядка:

$$\frac{V}{kRT} \frac{dp}{dt} = G_{\text{вх}} - G_{\text{вых}} . \quad (1)$$

Под входным расходом понимается расход воздуха через клапан, стоящий после компрессора. Под выходным расходом понимается совокупность расходов воздуха через клапаны, соединяющие баллон с атмосферой. Расход воздуха через клапан рассчитывается по формуле Сен-Венана – Венцеля:

$$G = F p_{ex} \sqrt{\frac{2}{RT} \frac{k}{k-1} \left[\left(\frac{p_{вых}}{p_{ex}} \right)^{\frac{2}{k}} - \left(\frac{p_{вых}}{p_{ex}} \right)^{\frac{k+1}{k}} \right]}, \quad (2)$$

где p_{ex} – давление на входе в клапан, Па; $p_{вых}$ – давление на выходе из клапана.

Управление клапанами осуществляется по ШИМ (широотно-импульсная модуляция), частотой 400 Гц. Для моделирования этого процесса будем умножать площадь проходного сечения клапана на число от 0 до 1 в зависимости от степени открытия клапана. Степень открытия клапана определяется по передаточной функции, описывающей динамику клапана, в которой постоянная времени $T = 0,01$ с, а коэффициент усиления $K = 1/24$ (напряжение на клапане 24 В, а степень открытия клапана 1). Так же в клапане присутствует нелинейность, которую можно описать следующими соотношениями:




Входное значение	Выходное значение
0	0
0,2	0
0,8	1
1	1



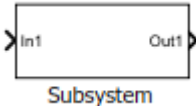
Входные значения ШИМ находятся в диапазоне от 0 до 600000. ШИМ управляет степенью открытия клапана, а выбор клапана (нагнетение давление от компрессора или сброс давления в атмосферу) осуществляется по знаку. Положительное число – клапан, соединяющий компрессор с баллоном, отрицательное число – клапан, соединяющий баллон с атмосферой.

2. Моделирование в MatLab

Для создания системы верхнего уровня использовались следующие элементы библиотеки Simulink [1], разделённые по разделам библиотеки:

Sources

 <p>Constant</p>	<p>Источник постоянного сигнала Constant</p> <p><i>Назначение:</i> Задаёт постоянный по уровню сигнал.</p> <p><i>Параметры:</i></p> <ol style="list-style-type: none"> 1. Constant value - Постоянная величина. 2. Interpret vector parameters as 1-D – Интерпретировать вектор параметров как одномерный (при установленном флажке). Данный параметр встречается у большинства блоков библиотеки Simulink. В дальнейшем он рассматриваться не будет. <p>Значение константы может быть действительным или комплексным числом, вычисляемым выражением, вектором или матрицей.</p>
 <p>Step</p>	<p>Генератор ступенчатого сигнала Step</p> <p><i>Назначение:</i> Формирует ступенчатый сигнал.</p> <p><i>Параметры:</i></p> <ol style="list-style-type: none"> 1. Step time - Время наступления перепада сигнала (с). 2. Initial value - Начальное значение сигнала. 3. Final value - Конечное значение сигнала. <p>Перепад может быть как в большую сторону (конечное значение больше чем начальное), так и в меньшую (конечное значение меньше чем начальное). Значения начального и конечного уровней могут быть не только положительными, но и отрицательными (например, изменение сигнала с уровня -5 до уровня -3).</p>
<p>Sinks</p>	
 <p>Display</p>	<p>Цифровой дисплей Display</p> <p><i>Назначение:</i> Отображает значение сигнала в виде числа.</p> <p><i>Параметры:</i></p> <ul style="list-style-type: none"> • Format – формат отображения данных. Параметр Format может принимать следующие значения: <ol style="list-style-type: none"> 1. short – 5 значащих десятичных цифр. 2. long – 15 значащих десятичных цифр. 3. short_e – 5 значащих десятичных цифр и 3 символа степени десяти. 4. long_e – 15 значащих десятичных цифр и 3 символа степени десяти. 5. bank – "денежный" формат. Формат с фиксированной точкой и двумя десятичными цифрами в дробной части числа. • Decimation – кратность отображения входного сигнала. При Decimation = 1 отображается каждое значение входного сигнала, при Decimation = 2 отображается каждое второе значение, при Decimation = 3 – каждое третье значение и т.д. • Sample time – шаг модельного времени. Определяет дискретность отображения данных. • Floating display (флажок) – перевод блока в "свободный" режим. В данном режиме входной порт блока отсутствует, а выбор сигнала для отображения

	<p>выполняется щелчком левой клавиши “мыши” на соответствующей линии связи. В этом режиме для параметра расчета Signal storage reuse должно быть установлено значение off (вкладка Advanced в окне диалога Simulation parameters...).</p>
 <p>Scope</p>	<p>Осциллограф Scope</p> <p><i>Назначение:</i></p> <p>Строит графики исследуемых сигналов в функции времени. Позволяет наблюдать за изменениями сигналов в процессе моделирования.</p>
<p>Math Operations</p>	
 <p>Sum</p>	<p>Блок вычисления суммы Sum</p> <p><i>Назначение:</i></p> <p>Выполняет вычисление суммы текущих значений сигналов.</p> <p><i>Параметры:</i></p> <ol style="list-style-type: none"> Icon shape – Форма блока. Выбирается из списка. <ul style="list-style-type: none"> - round – окружность, - rectangular – прямоугольник. List of sign – Список знаков. В списке можно использовать следующие знаки: + (плюс), - (минус) и (разделитель знаков). Saturate on integer overflow (флажок) – Подавлять переполнение целого. При установленном флажке ограничение сигналов целого типа выполняется корректно. <p>Количество входов и операция (сложение или вычитание) определяется списком знаков параметра List of sign, при этом метки входов обозначаются соответствующими знаками. В параметре List of sign можно также указать число входов блока. В этом случае все входы будут суммирующими.</p> <p>Если количество входов блока превышает 3, то удобнее использовать блок Sum прямоугольной формы.</p>
<p>Ports & Subsystems</p>	
 <p>Subsystem</p>	<p>Виртуальная и монолитная подсистемы Subsystem и Atomic Subsystem</p> <p>Доступ к окну параметров подсистемы осуществляется через меню Edit командой Block Parameters...</p> <p><i>Параметры:</i></p> <ol style="list-style-type: none"> Show port labels – Показать метки портов. Treat as atomic unit (флажок) – Считать подсистему монолитной. Таким образом, блоки виртуальной и монолитной подсистем – это один и тот же блок, отличающийся значением данного параметра. Access – Доступность подсистемы для изменений. Выбирается из списка: <ul style="list-style-type: none"> ○ ReadWrite – Пользователь может открывать и изменять подсистему. ○ ReadOnly – Пользователь может открывать подсистему только для

	<p>просмотра.</p> <ul style="list-style-type: none"> ○ NoReadOrWrite – Пользователь не может открывать и изменять подсистему. <p>4. Name of error callback function – Имя функции используемой для обработки ошибок возникающих в данной подсистеме.</p> <p>Остальные параметры подсистемы доступны при разработке приложений с использованием Real-Time Workshop и рассмотрены в документации на это приложение.</p> <p>Находящийся в библиотеке блок Subsystem (или Atomic Subsystem) содержит входной и выходной порты и линию связи между ними.</p> <p>После того как блок подсистемы скопирован из библиотеки в модель, он становится доступным для редактирования.</p>
--	--

Общий вид модели объекта управления и системы управления представлен на рисунке 2.

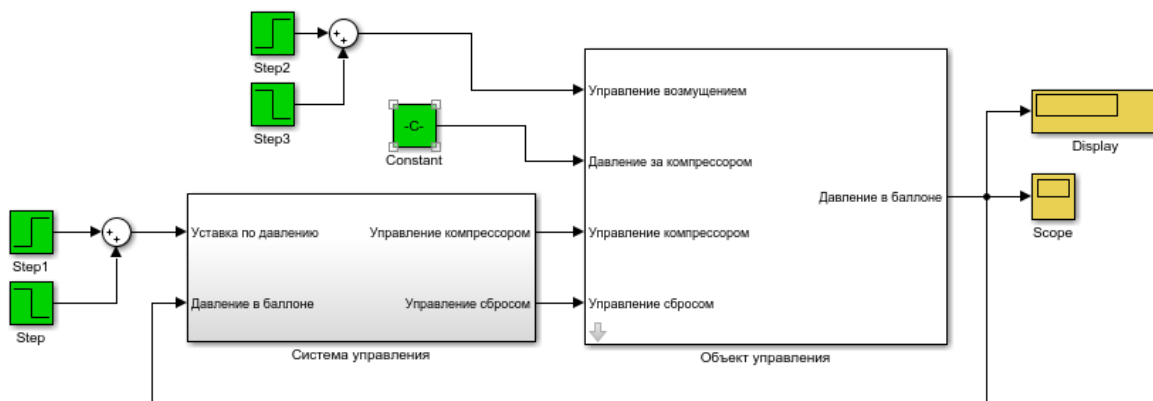


Рисунок 2 – Общий вид на модель

2.1. Создание объекта управления

Объект управления представляет собой подсистему (Subsystem) с включёнными в неё моделями баллона и клапанов (рисунок 3).

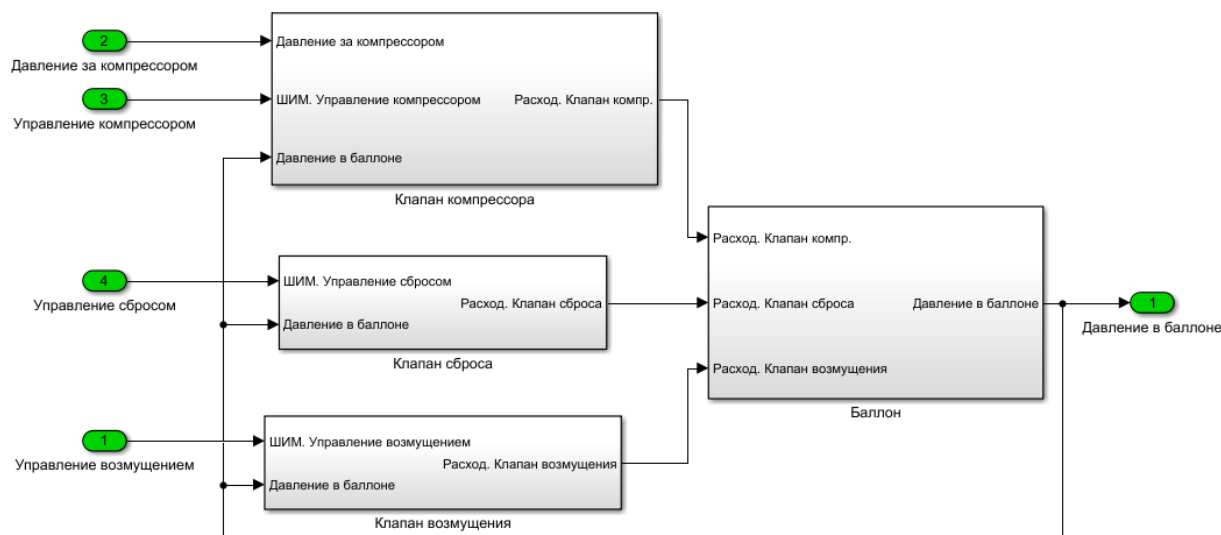




Рисунок 3 – Модель объекта управления

Sources	
	<p>Блок входного порта Inport</p> <p><i>Назначение:</i></p> <p>Создает входной порт для подсистемы или модели верхнего уровня иерархии.</p> <p><i>Параметры:</i></p> <ul style="list-style-type: none"> • Port number – Номер порта. • Port dimensions – Размерность входного сигнала. Если этот параметр равен – 1, то размерность входного сигнала будет определяться автоматически. • Sample time – Шаг модельного времени. • Data type – Тип данных входного сигнала: auto, double, single, int8, uint8, int16, uint16, int32, uint32 или boolean. • Signal type – Тип входного сигнала: <ol style="list-style-type: none"> 1. auto – Автоматическое определение типа. 2. real – Действительный сигнал. 3. complex – Комплексный сигнал. • Interpolate data (флажок) – Интерполировать входной сигнал. В случае, если временные отсчеты входного сигнала считываемого из рабочей области MATLAB не совпадают с модельным временем, то блок будет выполнять интерполяцию входного сигнала. При использовании блока Inport в подсистеме данный параметр не доступен.
Sinks	

	<h3 style="text-align: center;">Блок выходного порта Output</h3> <p><i>Назначение:</i></p> <p>Создает выходной порт для подсистемы или для модели верхнего уровня иерархии.</p> <p><i>Параметры:</i></p> <ul style="list-style-type: none"> • Port number – номер порта. • Output when disabled – вид сигнала на выходе подсистемы, в случае если подсистема выключена. Используется для управляемых подсистем. Может принимать значения (выбираются из списка): <ol style="list-style-type: none"> 1. held – выходной сигнал подсистемы равен последнему рассчитанному значению. 2. reset – выходной сигнал подсистемы равен значению задаваемому параметром Initial output. • Initial output - значение сигнала на выходе подсистемы до начала ее работы и в случае, если подсистема выключена. Используется для управляемых подсистем.
---	---

Для подсистем есть возможность накладывать «маску», в которой могут быть прописаны все необходимые константы. При этом будет осуществлена возможность удобного редактирования этих констант. Для этого на подсистеме нужно нажать правой кнопкой мыши и во всплывающем меню выбрать Mask -> Create Mask (либо, если она уже существует Edit Mask) (рисунок 4).

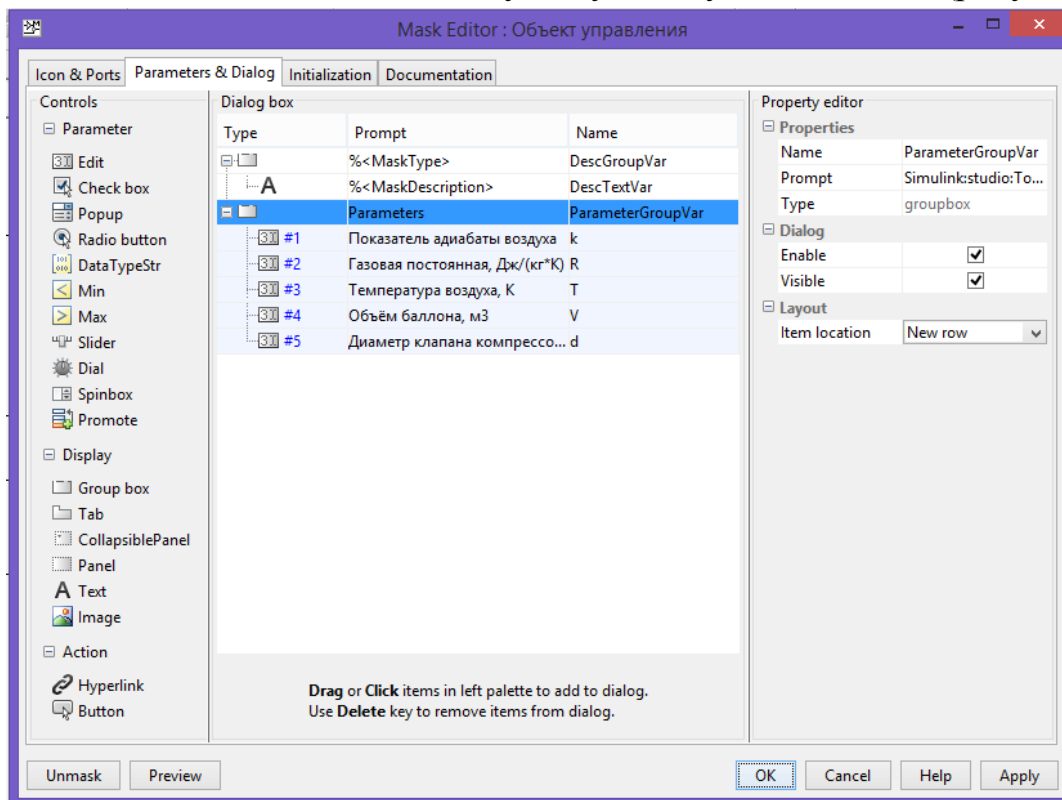


Рисунок 4 – Окно редактора «маски»

В открывшемся окне (рисунок 4) во вкладке Parameters & Dialog в левой колонке с названием Controls находятся элементы, выносимые в «маску». Для наших целей достаточно будет поля редактирования Edit (добавляется в список параметров щелчком левой кнопки мыши). В центральной части окна в колонке Prompt указывается название параметра или его описание, а в колонке Name – название переменной, которая будет использоваться в подсистеме. Значение переменной можно редактировать либо сразу (правая колонка, при нажатии на определённый элемент появляется поле для более тонкого редактирования), либо ввести нужное значение при открытии «маски» после окончания редактирования (рисунок 5).

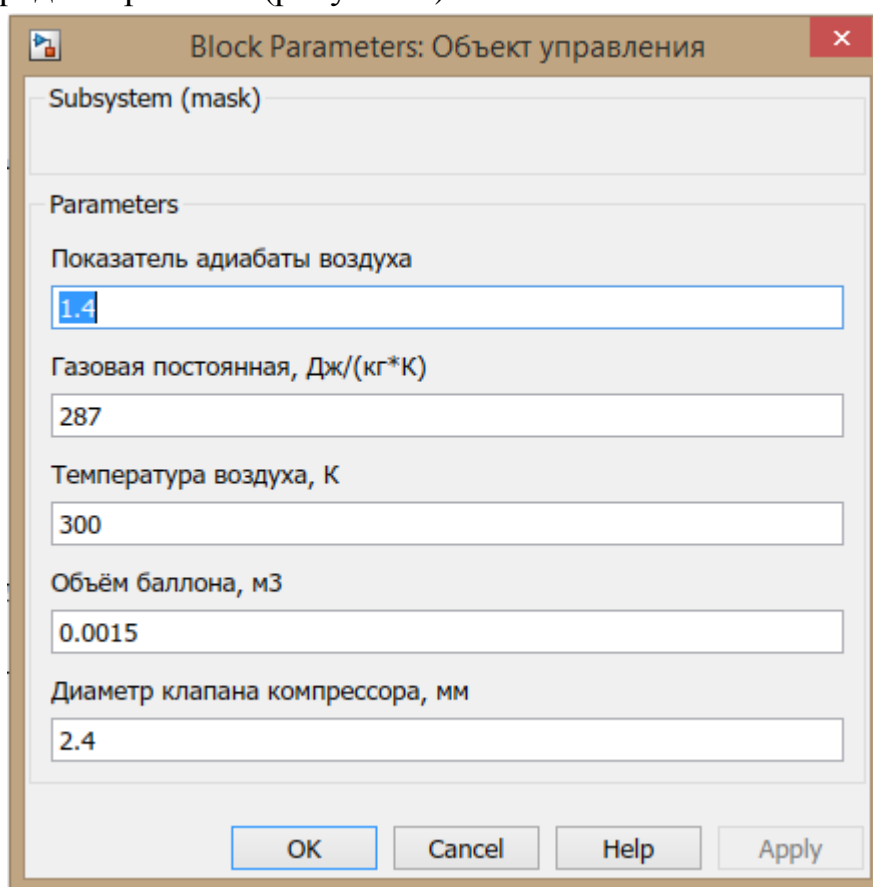


Рисунок 5 – Блок параметров «маски»

Во вкладке Initialization окна редактора «маски» можно ввести любые необходимые зависимости (рисунок 6). В данном случае реализован расчёт площади проточного сечения клапана.

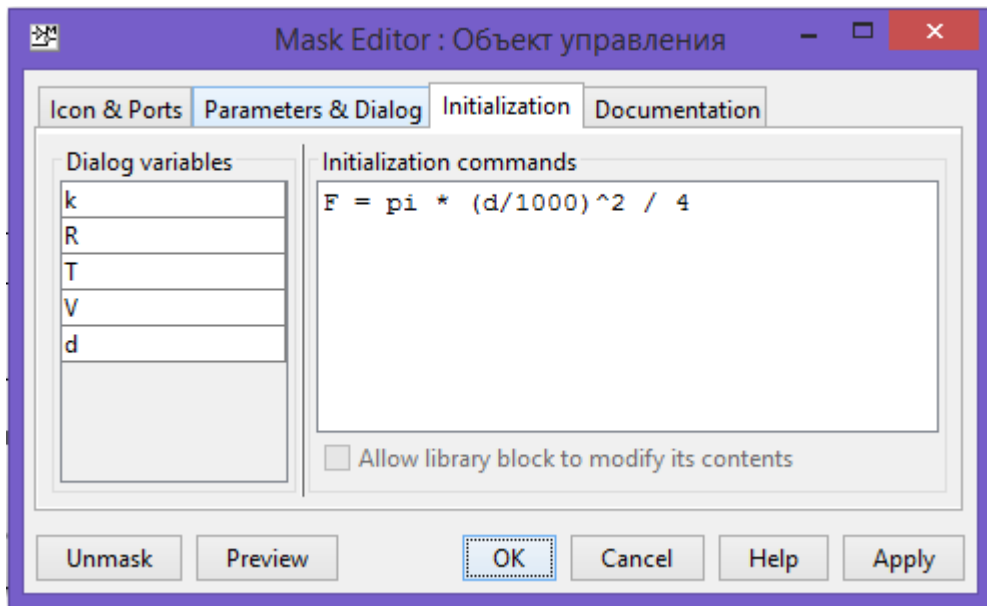


Рисунок 6 – Расчёт зависимостей в «маске»

Моделирование клапана. Согласно приведённым ранее уравнениям собирается схема, представленная на рисунке 7. Блок Fcn используется для вычисления выражения под корнем (формула (2)). Пример использования «маски» виден на блоке Gain, в котором используется площадь проточной части клапана F . В блоке 1-D Lookup Table присутствует описанная ранее нелинейность.

В моделируемой системе 3 клапана, которые моделируются одинаковым образом. Отличие между ними только в указании названий входных и выходных элементов для избегания путаницы при подключении подсистем друг с другом.

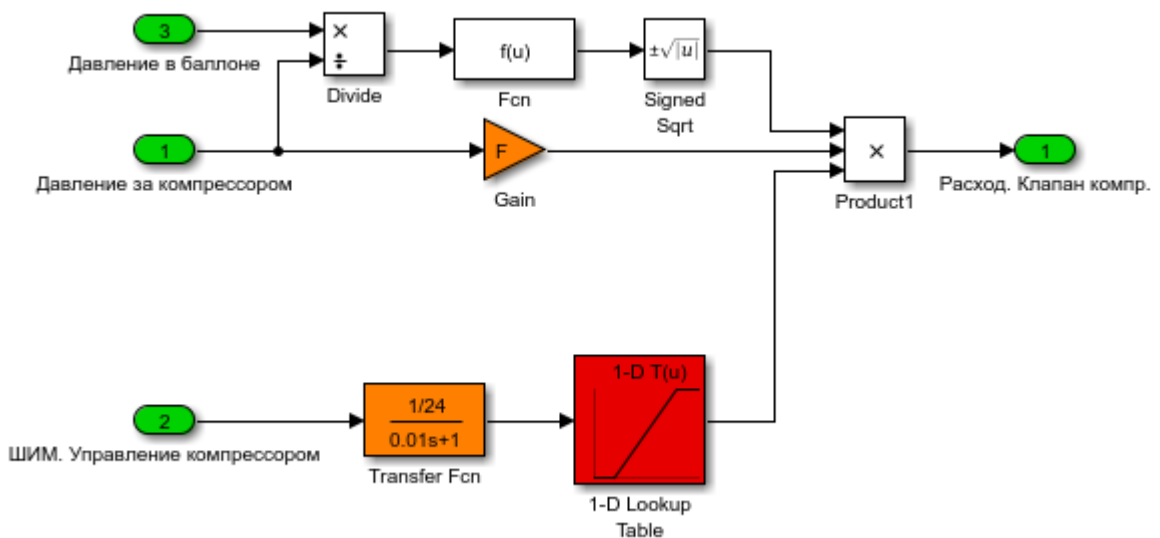


Рисунок 7 – Создание модели клапана

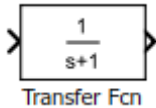
Continuous

Блок передаточной функции Transfer Fcn

Назначение:

Блок передаточной характеристики **Transfer Fcn** задает передаточную функцию в виде отношения полиномов:

$$H(s) = \frac{y(s)}{u(s)} = \frac{num(s)}{den(s)} = \frac{num(1)s^{nn-1} + num(2)s^{nn-2} + \dots + num(nn)}{den(1)s^{nd-1} + den(2)s^{nd-2} + \dots + den(nd)}$$



где

nn и **nd** – порядок числителя и знаменателя передаточной функции,
num – вектор или матрица коэффициентов числителя,
den – вектор коэффициентов знаменателя.

Параметры:

1. **Numerator** — вектор или матрица коэффициентов полинома числителя
2. **Denominator** -вектор коэффициентов полинома знаменателя
3. **Absolute tolerance** — Абсолютная погрешность.

Порядок числителя не должен превышать порядок знаменателя.

Math Operations

Усилители Gain и Matrix Gain

Назначение:


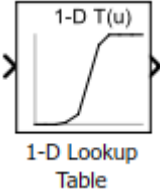
Выполняют умножение входного сигнала на постоянный коэффициент.

Параметры:

1. **Gain** – Коэффициент усиления.
2. **Multiplication** – Способ выполнения операции. Может принимать значения (из списка):
 - **Element-wise K*u**– Поэлементный.
 - **Matrix K*u** – Матричный. Коэффициент усиления является левосторонним операндом.
 - **Matrix u*K** – Матричный. Коэффициент усиления является правосторонним операндом.
3. **Saturate on integer overflow** (флажок) – Подавлять переполнение целого. При установленном флажке ограничение сигналов целого типа выполняется корректно.



Блоки усилителей **Gain** и **Matrix Gain** есть один и тот же блок, но с разными начальными установками параметра **Multiplication**.

	<p>Блок умножения Product</p> <p><i>Назначение:</i></p> <p>Выполняет вычисление произведения текущих значений сигналов.</p> <p><i>Параметры:</i></p> <ol style="list-style-type: none"> Number of inputs – Количество входов. Может задаваться как число или как список знаков. В списке знаков можно использовать знаки * (умножить) и / (разделить). Multiplication – Способ выполнения операции. Может принимать значения (из списка): <ul style="list-style-type: none"> - Element-wise – Поэлементный. - Matrix – Матричный. Saturate on integer overflow (флажок) – Подавлять переполнение целого. При установленном флажке ограничение сигналов целого типа выполняется корректно. <p>Если параметр Number of inputs задан списком, включающим кроме знаков умножения также знаки деления, то метки входов будут обозначены символами соответствующих операций.</p>
<h2>Lookup Tables</h2>	
	<p>Блок одномерной таблицы Look-Up Table</p> <p><i>Назначение:</i></p> <p>Задаёт в табличной форме функцию одной переменной.</p> <p><i>Параметры:</i></p> <ol style="list-style-type: none"> Vector of input values – Вектор значений входного сигнала. Может быть задан в виде дискретных значений (например, [1 2 7 9]), либо в виде непрерывного диапазона (например, [0:10]). Элементы вектора или граница диапазона могут быть заданы в виде вычисляемого выражения, например [tan(5) sin(3)]. Vector of output values – Вектор выходных значений, соответствующий вектору входных значений. <p>Блок работает в соответствии со следующими правилами:</p> <ol style="list-style-type: none"> Если входной сигнал равен одному из элементов вектора входных значений (Vector of input values), то выходное значение блока будет равно соответствующему элементу вектора выходных значений (Vector of output values). Например, пусть вектор входных значений равен [0 1 2 5], а вектор выходных значений [-5 -10 3 100], тогда при входном сигнале равном 1 выходной сигнал будет равен -10. Если входной сигнал не совпадает ни с одним из элементов вектора входных значений, то блок выполняет линейную интерполяцию между двумя ближайшими к нему элементами. Если входной сигнал выходит за границы вектора входных значений, то блок выполняет линейную экстраполяцию по двум крайним элементам. <p>График функции, заданный с помощью настроек блока отображается на его пикто-</p>

грамме.

Входной сигнал блока может быть векторным. В этом случае блок выполняет поэлементную операцию.

User-Defined Functions

Блок задания функции Fcn

Назначение:

Задаёт выражение в стиле языка программирования C .

Параметры:

Expression – Выражение, используемое блоком для вычисления выходного сигнала на основании входного. Это выражение составляется по правилам, принятым для описания функций на языке C.

В выражении можно использовать следующие компоненты:

1. Входной сигнал. Входной сигнал в выражении обозначается **u**, если он является скаляром. Если входной сигнал – вектор, необходимо указывать номер элемента вектора в круглых скобках. Например, **u(1)** и **u(3)** – первый и третий элементы входного вектора.
2. Константы.
3. Арифметические операторы (+ – * /).
4. Операторы отношения (= != > < >= <=).
5. Логические операторы (&& || !).
6. Круглые скобки.
7. Математические функции: **abs, acos, asin, atan, atan2, ceil, cos, cosh, exp, fabs, floor, hypot, ln, log, log10, pow, power, rem, sgn, sin, sinh, sqrt, tan, и tanh.**
8. Переменные из рабочей области. Если переменная рабочей области является массивом, то ее элементы должны указываться с помощью индексов в круглых скобках. Например, **A(1,1)** - первый элемент матрицы **A**.



Операторы отношения и логические операторы возвращают значения в виде логического нуля (**FALSE**) или логической единицы (**TRUE**).

Операторы, допускаемые к использованию в выражении, имеют следующий приоритет (в порядке убывания):

1. ()
2. + – (унарные)
3. Возведение в степень
4. !
5. /
6. + – (бинарные)
7. > < <= >=
8. = !=
9. &&
10. ||

Блок не поддерживает матричные и векторные операции. Выходной сигнал блока всегда – скаляр.

Моделирование баллона. Модель баллона представлена на рисунке 8. Так же, как и при создании клапана в этой подсистеме использовались переменные из «маски».

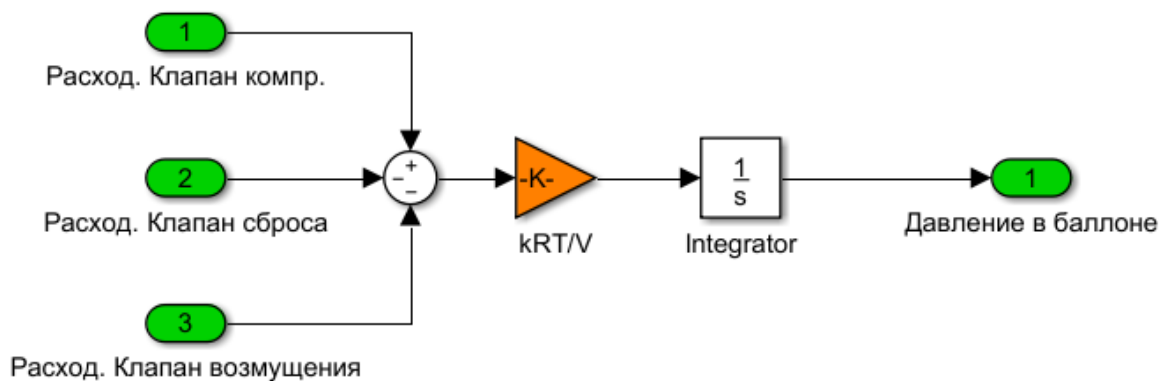



Рисунок 8 – Создание модели баллона

Continuous	
 Integrator	<p>Интегрирующий блок Integrator</p> <p><i>Назначение:</i></p> <p>Выполняет интегрирование входного сигнала.</p> <p><i>Параметры:</i></p> <ul style="list-style-type: none"> • External reset – Внешний сброс. Тип внешнего управляющего сигнала, обеспечивающего сброс интегратора к начальному состоянию. Выбирается из списка: <ol style="list-style-type: none"> 1. none – нет (сброс не выполняется), 2. rising - нарастающий сигнал (передний фронт сигнала), 3. falling - спадающий сигнал (задний фронт сигнала), 4. either – нарастающий либо спадающий сигнал, 5. level – не нулевой сигнал (сброс выполняется если сигнал на управляющем входе становится не равным нулю); <p>В том случае, если выбран какой-либо (но не none), тип управляющего сигнала, то на изображении блока появляется дополнительный управляющий вход. Рядом с дополнительным входом будет показано условное обозначение управляющего сигнала.</p> <ul style="list-style-type: none"> • Initial condition source — Источник начального значения выходного сигнала. Выбирается из списка: <ol style="list-style-type: none"> 1. internal – внутренний 2. external – внешний. В этом случае на изображении блока появляется дополнительный вход, обозначенный x_0, на который необходимо подать сигнал задающий начальное значение выходного сигнала интегратора.

- **Initial condition** — Начальное условие. Установка начального значения выходного сигнала интегратора. Параметр доступен, если выбран внутренний источник начального значения выходного сигнала.
- **Limit output** (флажок) — Использование ограничения выходного сигнала.
- **Upper saturation limit** — Верхний уровень ограничения выходного сигнала. Может быть задан как числом, так и символьной последовательностью **inf**, то есть $+\infty$.
- **Lower saturation limit** — Нижний уровень ограничения выходного сигнала. Может быть задан как числом, так и символьной последовательностью **inf**, то есть $-\infty$.
- **Show saturation port** — управляет отображением порта, выводящего сигнал, свидетельствующий о выходе интегратора на ограничение. Выходной сигнал данного порта может принимать следующие значения:
 1. **Ноль**, если интегратор не находится на ограничении.
 2. **+1**, если выходной сигнал интегратора достиг верхнего ограничивающего предела.
 3. **-1**, если выходной сигнал интегратора достиг нижнего ограничивающего предела.
- **Show state port** (флажок) — Отобразить/скрыть порт состояния блока. Данный порт используется в том случае, если выходной сигнал интегратора требуется подать в качестве сигнала обратной связи этого же интегратора. На пример, при установке начальных условий через внешний порт или при сбросе интегратора через порт сброса. Выходной сигнал с этого порта может использоваться также для организации взаимодействия с управляемой подсистемой.
- **Absolute tolerance** — Абсолютная погрешность.

2.2. Создание системы управления

Общая структура блока системы управления представлена на рисунке 9.

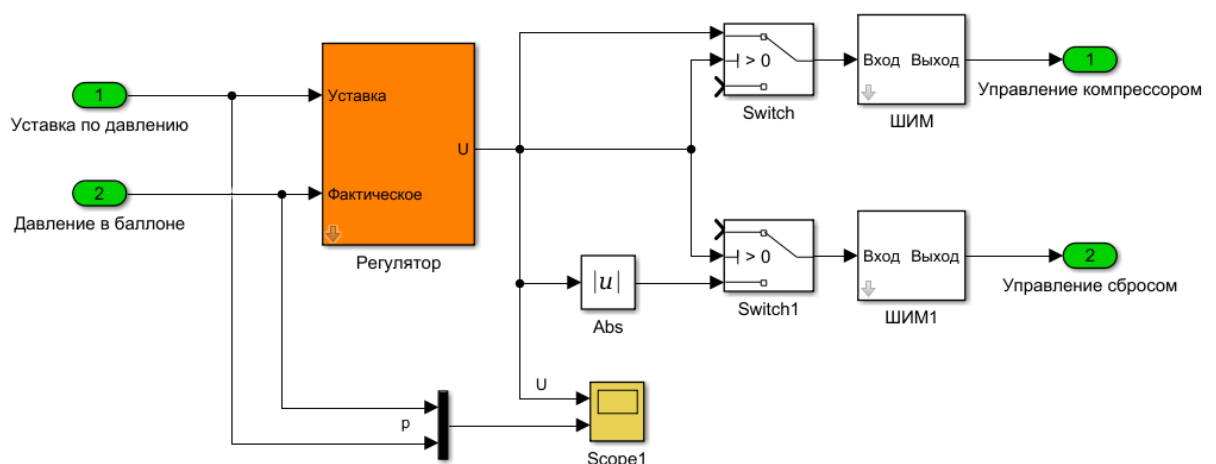

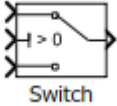
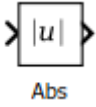


Рисунок 9 – Структура блока системы управления

	<p>Мультиплексор (смеситель) Mux</p> <p><i>Назначение:</i></p> <p>Объединяет входные сигналы в вектор.</p> <p><i>Параметры:</i></p> <ol style="list-style-type: none"> Number of Inputs - Количество входов. Display option - Способ отображения. Выбирается из списка: <ul style="list-style-type: none"> bar - Вертикальный узкий прямоугольник черного цвета. signals - Прямоугольник с белым фоном и отображением меток входных сигналов. none - Прямоугольник с белым фоном без отображения меток входных сигналов. <p>Входные сигналы блока могут быть скалярными и (или) векторными.</p>
	<p>Блок переключателя Switch</p> <p><i>Назначение:</i></p> <p>Выполняет переключение входных сигналов по сигналу управления.</p> <p><i>Параметры:</i></p> <p>Threshold – Порог управляющего сигнала.</p> <p>Блок работает следующим образом: Если сигнал управления, подаваемый на средний вход больше, чем величина порогового значения Threshold, то на выход блока проходит сигнал с первого (верхнего) входа. Если сигнал управления станет меньше, чем пороговое значение, то на выход блока будет поступать сигнал со второго (нижнего) входа.</p>
<p>Math Operations</p>	
	<p>Блок вычисления модуля Abs</p> <p><i>Назначение:</i></p> <p>Выполняет вычисление абсолютного значения величины сигнала.</p> <p><i>Параметры:</i></p> <ul style="list-style-type: none"> Saturate on integer overflow (флажок) – Подавлять переполнение целого. При установленном флажке ограничение сигналов целого типа выполняется корректно.

Моделирование ШИМ. Модель блока ШИМ была собрана по схеме, показанной на рисунке 10. Она работает следующим образом: входной постоянный сигнал «Constant2» преобразуется интегратором в линейно-изменяющийся. По достижении выходным сигналом значения равного

«Constant3», блок «Relational Operator» вырабатывает логический сигнал, по которому происходит сброс выходного сигнала интегратора до начального значения равного нулю. В результате на выходе интегратора формируется пилообразный сигнал, изменяющийся от 0 до $X_{max} = 600000$ с частотой $freq = 400$ Гц. Этот сигнал вычитается из непрерывного входного сигнала. Если значение отрицательно (входной сигнал больше пилообразного), то на выход блока ШИМ идёт значение $U_{max} = 24$, иначе – 0.

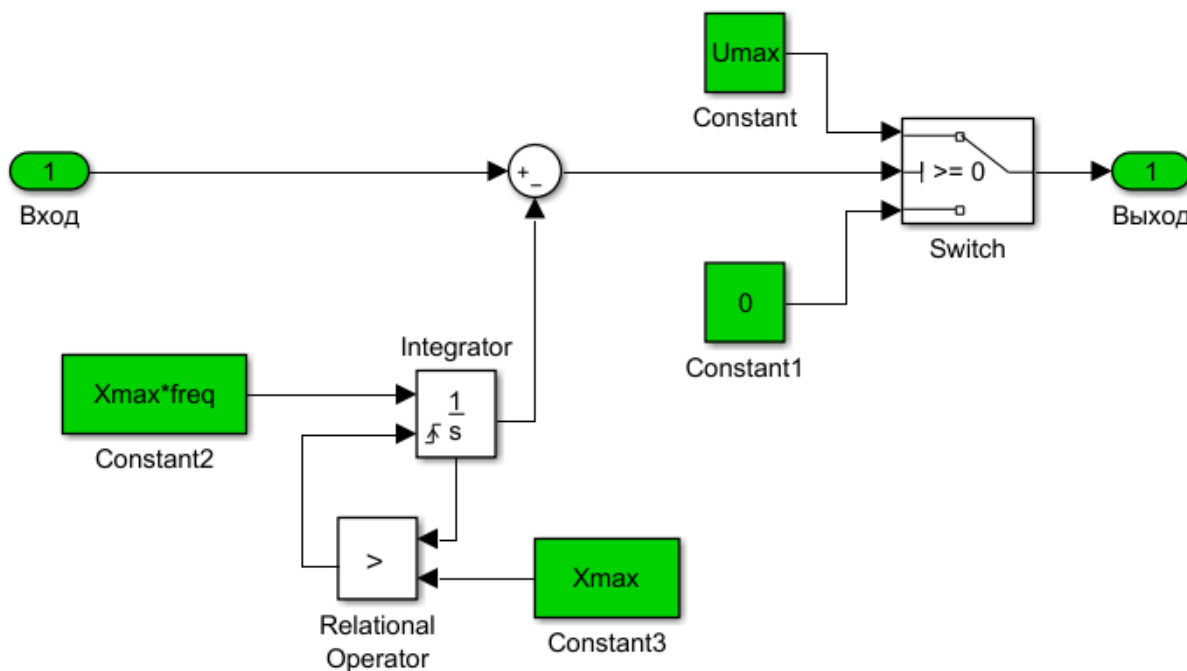



Рисунок 10 – Внутренняя структура блока ШИМ

Logic and Bit Operations	
 Relational Operator	<p>Блок вычисления операции отношения Relational Operator</p> <p><i>Назначение:</i></p> <p>Блок сравнивает текущие значения входных сигналов.</p> <p><i>Параметры:</i></p> <p>Relational Operator – Тип операции отношения (выбирается из списка):</p> <ul style="list-style-type: none"> • == - Точно равно. • ~= - Не равно. • < - Меньше. • <= - Меньше или равно. • >= - Больше или равно. • > - Больше. <p>В операции отношения первым операндом является сигнал, подаваемый на первый</p>

	<p>(верхний) вход блока, а вторым операндом – сигнал, подаваемый на второй (нижний) вход. Выходным сигналом блока является 1, если результат вычисления операции отношения есть “ИСТИНА” и 0, если результат – “ЛОЖЬ”.</p> <p>Входные сигналы блока могут быть скалярными, векторными или матричными. Если оба входных сигнала – векторы или матрицы, то блок выполняет поэлементную операцию сравнения, при этом размерность входных сигналов должна совпадать. Если один из входных сигналов – вектор или матрица, а другой входной сигнал – скаляр, то блок выполняет сравнение скалярного входного сигнала с каждым элементом массива. Размерность выходного сигнала, в этом случае, будет определяться размерностью векторного или матричного сигнала, подаваемого на один из входов.</p> <p>Для операций $=$ (тождественно равно) и \sim (не равно) допускается использовать комплексные входные сигналы.</p> <p>Входные сигналы также могут быть логического типа (boolean).</p>
--	--

Для удобства все переменные этого блока так же вынесены в «маску» (рисунок 11).

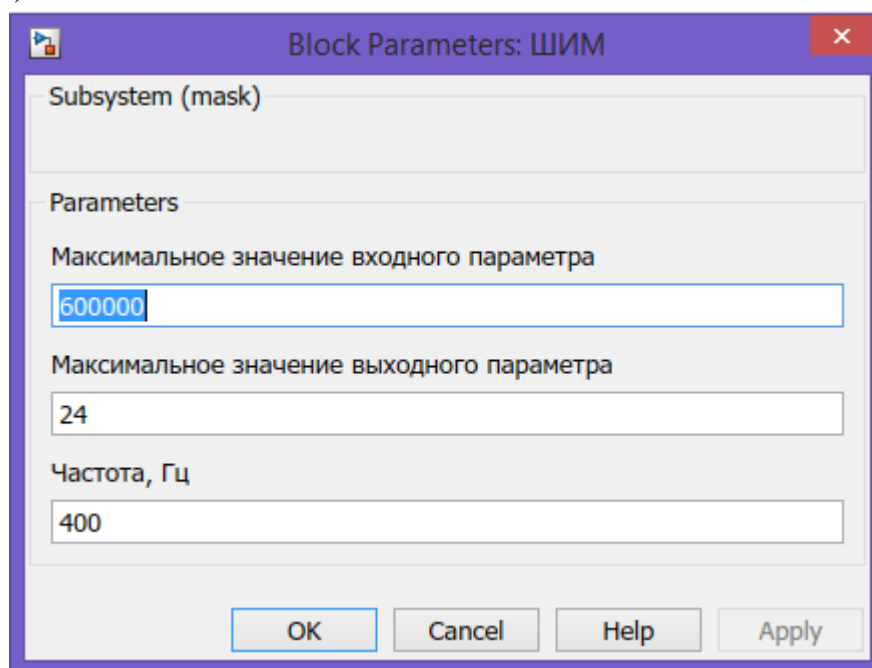


Рисунок 11 – «Маска» блока ШИМ

Моделирование регулятора. В данной работе мы будем моделировать два варианта системы управления: ПИ-регулятор и нечёткий регулятор (рисунок 11).

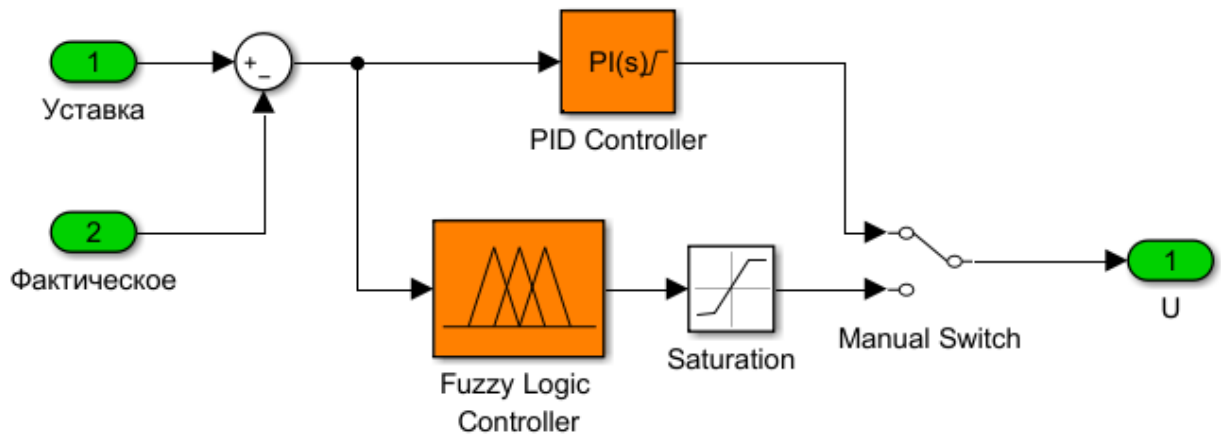

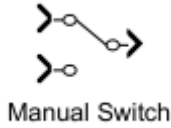


Рисунок 11 – Внутренняя структура блока регулятора

Discontinuities	
 Saturation	<p>Блок ограничения Saturation</p> <p><i>Назначение:</i></p> <p>Выполняет ограничение величины сигнала.</p> <p><i>Параметры:</i></p> <ol style="list-style-type: none"> Upper limit - Верхний порог ограничения. Lower limit - Нижний порог ограничения. Treat as gain when linearizing (флажок) - Трактовать как усилитель с коэффициентом передачи равным 1 при линеаризации. <p>Выходной сигнал блока равен входному если его величина не выходит за порог ограничения. По достижении входным сигналом уровня ограничения выходной сигнал блока перестает изменяться и остается равным порогу.</p>
Signal Routing	
 Manual Switch	<p>Блок ручного переключателя Manual Switch</p> <p><i>Назначение:</i></p> <p>Выполняет переключение входных сигналов по команде пользователя.</p> <p><i>Параметры:</i></p> <p>Нет.</p> <p>Командой на переключение является двойной щелчок левой клавишей “мыши” на изображении блока. При этом изображение блока изменяется, показывая, какой входной сигнал в данный момент проходит на выход блока. Переключение блока можно выполнять как до начала моделирования, так и в процессе расчета.</p>

ПИ-регулятор. ПИ регулятор в этой модели представлен в виде стандартного блока Simulink (поиск блоков библиотеки осуществляется в поле ввода поисковика, находящегося в левом верхнем углу окна Simulink Library

Browser). В общей теории автоматического управления структура регулятора выбирается исходя из модели объекта управления. При этом более сложным объектам управления соответствуют более сложные регуляторы.

Выходное значение ПИ-регулятора должно по размерности совпадать со входными значениями, поэтому ограничим верхний порог в 600000, а нижний в -600000 (знак влияет на выбор клапана, а значение – максимальное возможное в этой системе и ограниченное давлением за компрессором).

Переходный процесс с созданной системой управления приведён на рисунке 12, где сверху показано значение выхода регулятора, а снизу – значения уставки (пунктир) фактическое значение давления (сплошная линия) и открытие клапана возмущения (синий).

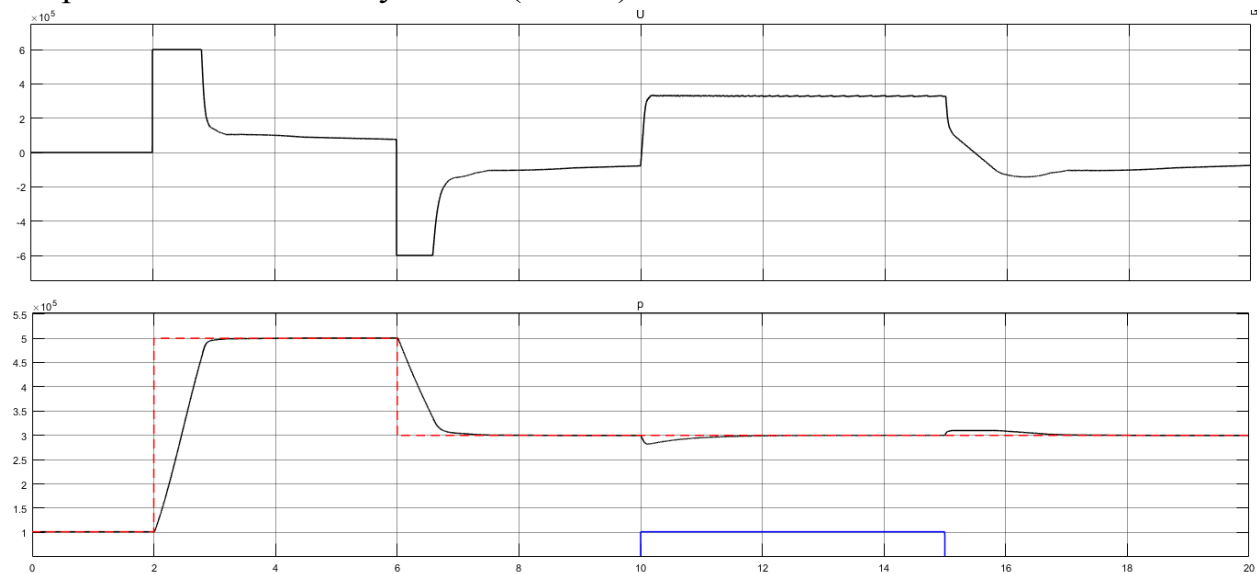


Рисунок 12 – Переходный процесс с ПИ-регулятором

Регулятор на нечёткой логике (Fuzzy logic controller). Нечеткое управление [2] (управление на основе методов теории нечетких множеств) [3 – 5] используется при недостаточном знании объекта управления, но наличии опыта управления им, в нелинейных системах, идентификация которых слишком трудоемка, а также в случаях, когда по условию задачи необходимо использовать знания эксперта. Примером может быть доменная печь или ректификационная колонна, математическая модель которых содержит много эмпирических коэффициентов, изменяющихся в широком диапазоне и вызывающих большие затруднения при идентификации [3]. В то же время квалифицированный оператор достаточно хорошо управляет такими объектами, пользуясь показаниями приборов и накопленным опытом.

Поскольку информация, полученная от оператора, выражена словесно, для ее использования в ПИД-регуляторах применяют лингвистические переменные и аппарат теории нечетких множеств, который был разработан Л. За-

де в 1965 году [6]. Основная идея этой теории состоит в следующем. Если в теории четких множеств некоторый элемент (например, температура 50 град.) может принадлежать множеству (например, множеству "температура горячей воды $T_{гор}$) или не принадлежать ему, то в теории нечетких множеств вводится понятие функции принадлежности, которая характеризует степень принадлежности элемента множеству. При этом говорят, например, температура 50 град. принадлежит множеству $T_{гор}$ со степенью принадлежности 0,264. Функцию принадлежности можно приближенно трактовать как вероятность того, что данный элемент принадлежит множеству [7], однако такая интерпретация, хотя и является для инженеров более понятной, не является математически строгой, поскольку существующая теория нечетких множеств не оперирует понятием вероятности.

Нечеткая логика в ПИД-регуляторах используется преимущественно двумя путями: для построения самого регулятора (рисунок 13) и для организации подстройки коэффициентов ПИД-регулятора. Оба пути могут использоваться в ПИД-контроллере одновременно.

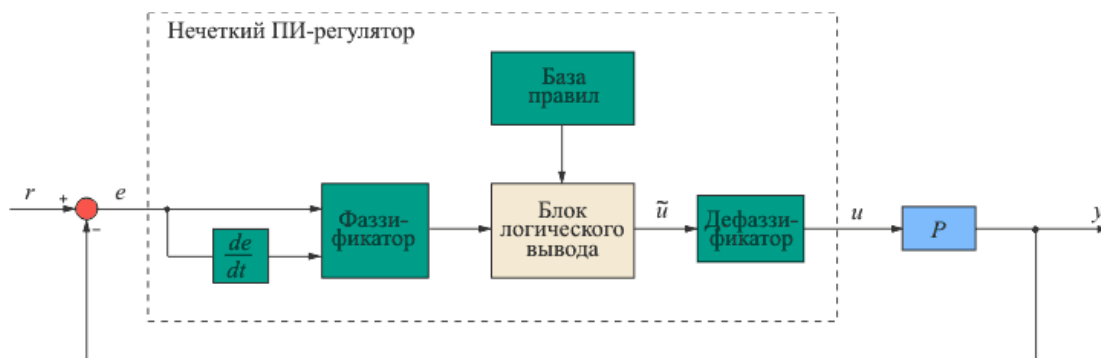


Рисунок 13 – Структура нечеткого регулятора

Для применения методов нечеткой логики прежде всего необходимо преобразовать обычные четкие переменные в нечеткие. Процесс такого преобразования называется фаззификацией (от английского "fuzzy" – "нечеткий"). Он иллюстрируется на рисунке 14.

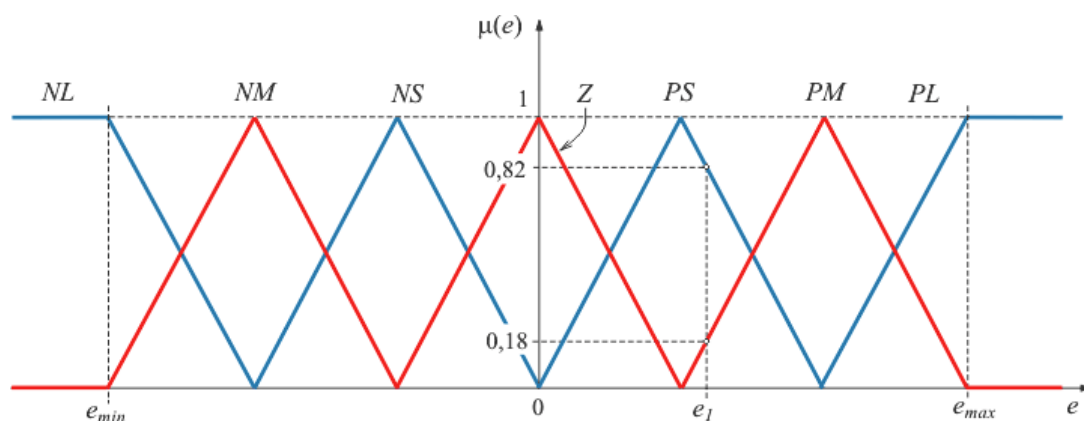


Рисунок 14 – Деление области изменения переменной e на множества NL, NM, NS и т.д. с функциями принадлежности $\mu(e)$ треугольной формы

Диапазон изменения переменной e разбивается на множества (подмножества) NL, NM, NS, Z, PS, PM, PL, в пределах каждого из которых строится функция принадлежности переменной e каждому из множеств. На рисунке 14 функции принадлежности имеют треугольную (наиболее распространенную) форму, хотя в общем случае они могут быть любыми, исходя из смысла решаемой задачи [5]. Количество множеств также может быть произвольным.

Для нечетких множеств существует общепринятая система обозначений: N – отрицательный (Negative); Z – нулевой (Zero); P – положительный (Positive); к этим обозначениям добавляют буквы S (малый, Small), M (средний, Medium), L (большой, Large). Например, NL – отрицательный большой; NM – отрицательный средний (Negative Medium); PL – положительный большой. Количество таких переменных (термов) может быть любым, однако с увеличением их количества существенно возрастают требования к опыту эксперта, который должен сформулировать правила для всех комбинаций входных переменных.

Если величина ошибки e на входе нечеткого регулятора (рисунок 13) равна e_1 (рисунок 14), то соответствующее значение нечеткой переменной будет равно PS со степенью принадлежности подмножеству PS, равной $\mu(e_1) = 0,82$, или равно PM со степенью принадлежности $\mu(e_1) = 0,18$. Степень принадлежности ошибки e_1 другим множествам (Z, PL, NS и др.) равна нулю. Таким образом, величина ошибки e_1 оказалась преобразованной в нечеткие переменные.

Для выполнения функции регулирования над нечеткими переменными должны быть выполнены операции, построенные на основании высказываний оператора, сформулированных в виде нечетких правил. Совокупность нечетких правил и нечетких переменных используется для осуществления нечеткого логического вывода, результатом которого является управляющее воздействие на объект управления (рисунок 13).

Нечеткий вывод выполняется следующим образом. Предположим, что область изменения ошибки e разделена на множества N, Z, P , область изменения управляющего воздействия – на множества NL, NM, Z, PM, PL и что с помощью эксперта удалось сформулировать следующие правила работы регулятора:

Правило 1: если $e = N$ и $de/dt = P$, то $\tilde{u} = Z$

Правило 2: если $e = N$ и $de/dt = Z$, то $\tilde{u} = NM$

Правило 3: если $e = N$ и $de/dt = N$, то $\tilde{u} = NL$

Правило 4: если $e = Z$ и $de/dt = P$, то $\tilde{u} = PM$

Правило 5: если $e = Z$ и $de/dt = Z$, то $\tilde{u} = Z$

Правило 6: если $e = Z$ и $de/dt = N$, то $\tilde{u} = NM$

Правило 7: если $e = P$ и $de/dt = P$, то $\tilde{u} = PL$

Правило 8: если $e = P$ и $de/dt = Z$, то $\tilde{u} = PM$ *PM*

Правило 9: если $e = P$ и $de/dt = N$, то $\tilde{u} = Z$.

Операция "И" в приведённых выше правилах соответствует пересечению множеств, а результат применения всех правил соответствует операции объединения множеств [5]. Функция принадлежности для пересечения двух множеств, например, N и P (см. Правило 1) находится как [5]:

$$\mu_{e \cap de/dt} = \min(\mu_e, \mu_{de/dt}),$$

т.е. каждое значение функции принадлежности пересечения множеств равно наименьшему значению из двух, стоящих в круглых скобках.

Функция принадлежности для объединения тех же множеств имеет вид [5]:

$$\mu_{e \cup de/dt} = \max(\mu_e, \mu_{de/dt}).$$

Функции принадлежности, полученные при пересечении или объединении множеств, могут быть определены различными способами, в зависимости от смысла решаемой задачи. В этом смысле сама теория нечетких множеств тоже является нечеткой. В [5] приводится 10 различных определений функции принадлежности для пересечения множеств, но не сказано, какое из них нужно выбрать для решения конкретной задачи. Используют, в частности, более понятную операцию нахождения функций принадлежности в случае пересечения и объединения множеств, имеющую аналогию с правилами умножения и сложения вероятностей:

$$\mu_{e \cap de/dt} = \mu_e \cdot \mu_{de/dt}$$

$$\mu_{e \cup de/dt} = \mu_e + \mu_{de/dt} - \mu_e \cdot \mu_{de/dt}.$$

Однако применение первых двух способов нахождения функции принадлежности обычно более предпочтительно, т.к. при этом сохраняется большинство правил, разработанных для обычных множеств [4].

Функции принадлежности для каждого из множеств NL, NM, Z, PM, PL, входящих в нечеткую переменную \tilde{u} в правилах, получаются в виде [5]:

$$\begin{aligned} \mu_{\pi_1}(\tilde{u}) &= \min\{\mu_{u_1}(\tilde{u}), \min(\mu_{e_1}(e), \mu_{de/dt_1}(de/dt))\}, \\ \mu_{\pi_2}(\tilde{u}) &= \min\{\mu_{u_2}(\tilde{u}), \min(\mu_{e_2}(e), \mu_{de/dt_2}(de/dt))\}, \\ &\dots \\ \mu_{\pi_9}(\tilde{u}) &= \min\{\mu_{u_9}(\tilde{u}), \min(\mu_{e_9}(e), \mu_{de/dt_9}(de/dt))\}. \end{aligned}$$

Здесь каждое из 9-ти уравнений соответствует одному из правил. Результирующая функция принадлежности управляющего воздействия \tilde{u} , полученная после применения всех 9-ти правил, находится как объединение функций принадлежности всех правил:

$$\mu(\tilde{u}) = \max\{\mu_{\pi_1}(\tilde{u}), \mu_{\pi_2}(\tilde{u}), \dots, \mu_{\pi_9}(\tilde{u})\}.$$

Теперь, когда получена результирующая функция принадлежности управляющего воздействия u , возникает вопрос, какое конкретно значение управляющего воздействия нужно выбрать. Если использовать вероятностную интерпретацию теории нечетких множеств, то становится понятно, что такое значение можно получить по аналогии с математическим ожиданием управляющего воздействия в виде:

$$u = \frac{\int_{u \min}^{u \max} \tilde{u} \mu(\tilde{u}) d\tilde{u}}{\int_{u \min}^{u \max} \mu(\tilde{u}) d\tilde{u}}.$$

Такой способ дефаззификации является наиболее распространенным, но не единственным.

Для построения нечетких регуляторов обычно используют П, И, ПИ и ПД ПД+И, ПИ+Д и ПИД-законы регулирования. В качестве входных сигналов для системы нечеткого вывода используют сигнал ошибки, приращение ошибки, квадрат ошибки и интеграл от ошибки. Реализация нечеткого ПИД регулятора вызывает проблемы, поскольку он должен иметь трехмерную таблицу правил в соответствии с тремя слагаемыми в уравнении ПИД-регулятора, которую чрезвычайно сложно заполнить, пользуясь ответами эксперта.

В соответствии с вышесказанным создадим нечёткую систему управления на основе П-закона регулирования (нечёткий регулятор обрабатывает только само значение ошибки рассогласования). Создавать регулятор будем

не через графический режим Fuzzy Logic Toolbox, а с помощью скриптового файла MatLab, используя функции из этого тулбокса.

Процесс создания следующий:

1. Создание нового скриптового файла (главное окно MatLab → HOME → FILE → New Script)
2. Написание самого текста скрипта (см. ПРИЛОЖЕНИЕ 1)
3. Сохранение скрипта в рабочей директории
4. Запуск скрипта
5. Перемещение блока Fuzzy Logic Controller из библиотеки Simulink в модель (аналогично блоку PID Controller) в модель
6. Указание названия файла со структурой нечёткого регулятора (двойной щелчок мыши на нечётком контроллере и ввод названия созданного скриптом .fis-файла в одинарных кавычках)

Подробнее узнать о синтаксисе используемых функций можно введя команду `help *название_функции*` в командном окне MatLab.

Строгих рекомендаций к созданию термов нечётких логических переменных нет. Наиболее часто используемые термы – треугольный и трапециевидный. Так же, определение того, что считается большой ошибкой рассогласования, а что – маленькой зависит от человека, создающего регулятор. Стоит, однако, остановиться на том, почему диапазон выхода регулятора принят в два раза большим, чем в ПИ-регуляторе. В качестве метода дефазификации принят метод центра тяжести, а значит максимальное значение выходной величины в нашем случае будет примерно в центре между нулём и требуемым максимальным значением. Увеличение максимального выходного значения позволяет получить на выходе необходимые 600000. Так же на выходе из нечёткого регулятора установлен ограничитель значения в том же диапазоне, что и для П-регулятора.

Переходный процесс с нечётким регулятором представлен на рисунке 15.

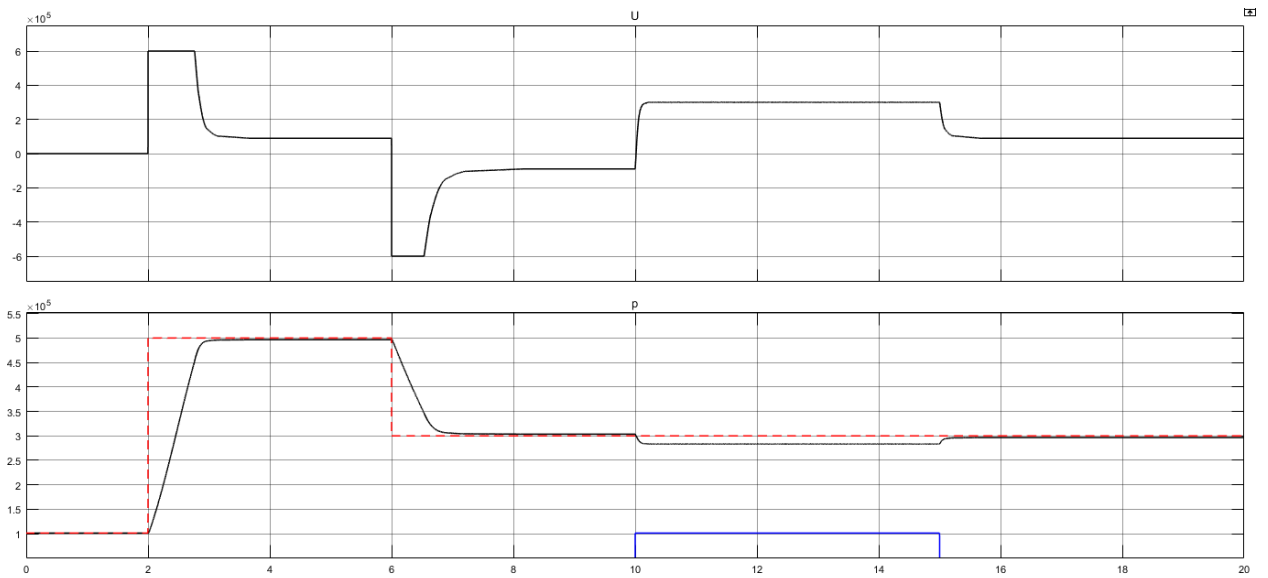


Рисунок 15 – Переходный процесс fuzzy-регулятора

3. Задание

1. Создать математическую модель объекта управления с теми же параметрами, что были указаны в разделе 1.
2. Создать систему управления с ПИ-регулятором и нечётким регулятором.
3. Получить графики переходных процессов для каждого из регуляторов при наличии управляющего и возмущающего воздействий.

ПРИЛОЖЕНИЕ 1

```
% Создание объекта - нечёткий логический регулятор
fuzzy = newfis('fuzzy_ballon');

% Создание нечётких логических переменных
E = 0.1e5;
fuzzy = addvar(fuzzy, 'input', 'E', [-500000 500000]);
fuzzy = addmf (fuzzy, 'input', 1, 'N', 'trapmf', [-500000 -500000 -E/6 -E/12]); %1
fuzzy = addmf (fuzzy, 'input', 1, 'Z', 'trimf', [-E/6 0 E/6]); %2
fuzzy = addmf (fuzzy, 'input', 1, 'P', 'trapmf', [ E/12 E/6 500000 500000]); %3

U = 1200000;
fuzzy = addvar(fuzzy, 'output', 'U', [-U U]);
fuzzy = addmf (fuzzy, 'output', 1, 'N', 'trimf', [-U -U 0]); % 1
fuzzy = addmf (fuzzy, 'output', 1, 'Z', 'trimf', [-U 0 U]); % 2
fuzzy = addmf (fuzzy, 'output', 1, 'P', 'trimf', [ 0 U U]); % 3

% Создание базы правил
%           E U w &
ruleList = [1 1 1 1;
            2 2 1 1;
            3 3 1 1];

fuzzy = addrule(fuzzy, ruleList);

% Запись регулятора в файл
writefis(fuzzy, 'fuzzy_ballon');
```

Список использованных источников

1. <http://matlab.exponenta.ru/simulink/book1/>
2. Методы робастного, нейро-нечеткого и адаптивного управления. Ученик/ Под ред. Н. Д. Егупова, изд. 2-е. М.: Изд-во МГТУ им. Бауман, 2002, 744 с.
3. http://bookasutp.ru/Chapter5_7.aspx
4. Усков А.А., Кузьмин А.В. Интеллектуальные технологии управления. Искусственные нейронные сети и нечеткая логика. М.: "Горячая линия-Телеком", 2004, 143 с.
5. Рутковская Д, Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. М.: Горячая линия-Телеком, 1006, 383 с.
6. Zadeh L.A. Fuzzy sets. - Information and Control. 1965, №8, p.338-353
7. Ротач В.Я. Теория автоматического управления. М.: МЭИ, 2004, 400 с.