

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Оренбургский государственный университет»

Кафедра управления и информатики в технических системах

ЛАБОРАТОРНЫЕ РАБОТЫ ПО ДИСЦИПЛИНЕ «ОСНОВЫ МЕХАТРОНИКИ»

Методические указания

Составитель

С.С. Акимов

Рекомендовано к изданию редакционно-издательским советом федерального государственного бюджетного образовательного учреждения высшего образования «Оренбургский государственный университет» для обучающихся по образовательным программам высшего образования по направлениям подготовки 27.03.04 Управление в технических системах, 27.03.03 Системный анализ и управление

Оренбург
2021

УДК 681.5(076.5)
ББК 32.966я7
Л12

Рецензент – профессор, доктор экономических наук В.Н. Шепель

Л12 **Лабораторные работы по дисциплине «Основы мехатроники»:**
методические указания / составитель С.С. Акимов; Оренбургский
гос. ун-т. – Оренбург : ОГУ, 2021. – 49 с.

В методических указаниях приведены основные положения по лабораторным работам, формируемые компетенции, а также указания к выполнению лабораторных работ.

Методические указания предназначены для выполнения лабораторных работ по дисциплине «Основы мехатроники» для обучающихся очной и заочной форм обучения по образовательным программам высшего образования по направлениям подготовки 27.03.04 Управление в технических системах и 27.03.03 Системный анализ и управление.

УДК 681.5(076.5)
ББК 32.966я7

© Акимов С.С.,
составление, 2021
© ОГУ, 2021

Содержание

Введение.....	4
1 Лабораторная работа № 1. Математическое моделирование мехатронной системы.....	6
2 Лабораторная работа № 2. Имитационное моделирование мехатронной системы.....	13
3 Лабораторная работа № 3. Составление циклограмм работы мехатронной системы.....	17
4 Лабораторная работа № 4. Подбор устройств и составление кинематической схемы работы мехатронной системы	20
5 Лабораторная работа № 5. Составление структурной схемы мехатронной системы.....	24
6 Лабораторная работа № 6. Составление электрической, пневматической и функциональной схем мехатронной системы	27
7 Лабораторная работа № 7. Составление схемы алгоритма программы контроллера мехатронной системы.....	36
8 Лабораторная работа № 8. Программирование контроллера мехатронной системы.....	39
9 Лабораторная работа № 9. Интеграция мехатронной системы в общую систему управления.....	45
Список использованных источников	48

Введение

Методические указания посвящены изучению теоретических основ мехатроники и практических инструментов по их созданию в рамках формирования профессиональных компетенций бакалавров по направлениям 27.03.03 Системный анализ и управление и 27.03.04 Управление в технических системах при изучении дисциплины «Основы мехатроники».

Лабораторная работа – одна из ключевых форм учебного занятия, направленная на развитие самостоятельности студентов, приобретение умений и формирования общих и профессиональных компетенций. Выполнение лабораторных работ способствует углубленному пониманию, усвоению и закреплению пройденного материала профессионального модуля, развитию мышления и творческого потенциала.

Целью лабораторных работ по учебным дисциплинам является закрепление студентами пройденного теоретического материала по профессиям и выработка навыков самостоятельной профессиональной и научно-исследовательской деятельности.

Задачи лабораторных работ обусловлены необходимостью получения обучающимися знаний, умений, навыков согласно требованиям ФГОС ВО, на основе которых формируются соответствующие профессиональные компетенции. Ведущей дидактической целью лабораторных работ является формирование умений, необходимых в последующей учебной деятельности по общепрофессиональным, профессиональным дисциплинам.

Методические указания по выполнению заданий на лабораторных работах разработаны по разделам рабочей программы. Подготовка к лабораторным работам заключается в изучении теории по рекомендуемой литературе, предусмотренной рабочей программой.

Для эффективного выполнения заданий студенты должны знать теоретические материалы и уметь применять эти знания для приобретения навыков при выполнении лабораторных работ.

В конце занятия преподаватель выставляет оценку, которая складывается из результатов наблюдения за выполнением практической части работы, проверки отчета, беседы в ходе работы или после нее. Оценки за выполнение лабораторных работ выставляются по пятибалльной системе.

Условия и порядок выполнения работы:

1 Изучить методические рекомендации по выполнению лабораторной работы.

2 Ответить на перечень вопросов, необходимых для выполнения заданий.

3 Изучить содержание заданий и начать выполнение.

4 Консультацию по выполнению работы получить у преподавателя.

5 Работа оценивается в целом, по итогам выполнения работы выставляется оценка.

Защита проводится путем индивидуальной беседы или выполнения зачетного задания.

Процесс выполнения лабораторных работ направлен на формирование следующих компетенций:

– ОПК-9 способность использовать навыки работы с компьютером, владеть методами информационных технологий, соблюдать основные требования информационной безопасности;

– ПК-2 способность проводить вычислительные эксперименты с использованием стандартных программных средств с целью получения математических моделей процессов и объектов автоматизации и управления.

1 Лабораторная работа №1. Математическое моделирование мехатронной системы

1.1 Цель работы

Цель работы заключается в построении математической модели разрабатываемой мехатронной системы.

1.2 Краткие теоретические сведения

Математическое моделирование – все методы, основанные на построении и использовании различных форм математических моделей проектируемых объектов, независимо от того, как они реализуются. При математическом моделировании описание системы производится в терминах некоторой математической теории, например, теории матриц, теории дифференциальных уравнений и т. д.

Математическое моделирование основано на ограниченности числа фундаментальных законов природы и принципе подобия, означающем, что явления различной физической природы могут описываться одинаковыми математическими закономерностями.

Как и всякие модели, математические модели основаны на некотором упрощении, идеализации, отбрасывании факторов, которые для данной задачи или на данном этапе исследований представляются несущественными.

В зависимости от формы представления математические модели можно разделить на аналитические, структурные и алгоритмические.

Аналитические модели представляют собой отображение взаимосвязей между переменными объекта в виде дифференциальных, алгебраических или любых других систем математических уравнений. Такие модели обычно получают на основе физических законов. Использование аналитических моделей

позволяет исследовать фундаментальные свойства объекта, часто без использования ЭВМ.

Структурная модель представляет систему в виде совокупности элементов, а также совокупности необходимых и достаточных отношений между этими элементами и связей между системой и окружающей средой.

В простейшем случае с помощью структурной математической модели воспроизводится структура уравнений, описывающих поведение исследуемого объекта. Вариантами структурных моделей являются графы, структурные и функциональные схемы, диаграммы и т. д.

Алгоритмические модели воспроизводят пошаговый процесс численного решения уравнений, представляющих математическую модель исследуемого объекта, и обычно реализуются в форме программы для ЭВМ. Результаты исследования на таких моделях всегда являются приближенными. Применение компьютеров делает алгоритмические модели наиболее универсальными.

Математические модели технических объектов должны быть по сложности согласованы с возможностями восприятия человеком и с возможностями ЭВМ оперировать этими моделями. Как правило, решить все задачи в рамках некоторого единого описания невозможно. Обычно требуется структурирование математических моделей на несколько иерархических уровней, отличающихся детальностью описания технического объекта [1].

1.3 Задание к лабораторной работе

Рассматривается следующий участок цеха. Конвейер перемещает разноцветные детали. В непосредственной близости от конвейера находится манипулятор, который представляет собой два звена и захватное устройство. Начальное звено крепится к муфте, способной поворачиваться вокруг вертикальной оси. Помимо этого, начальное звено способно двигаться и в горизонтальной плоскости. Конечное звено сцеплено с начальным и также способно двигаться в горизонтальной плоскости. Захватное устройство механическое, жестко сцеп-

лено с конечным звеном и оборудовано пневмоприводом. Сам манипулятор осуществляет движения посредством серводвигателей.

Стартовое положение манипулятора выверено так, что захватное устройство направлено на ленту (вращения по вертикальной оси не требуется).

Манипулятор соединен с датчиком распознавания цвета. При появлении детали определенного цвета (например, красной), манипулятор осуществляет ее захват (движение в горизонтальной плоскости), затем перемещение в сторону (движение в вертикальной плоскости) и установку ее в специальную кассету. Движение по каждой оси производится в течении 1 с.

Требуется: определить текущее положение манипулятора и угловую скорость, отдельно для горизонтальной и вертикальной осей. Варианты заданий приведены в таблице 1.

Таблица 1 – Варианты заданий

Вариант	Горизонтальная ось						Вертикальная ось
	Нач. звено a , м	Кон. звено b , м	Угол основания φ_0 , град.	Угол м/у звеньями ψ_0 , град.	Смещение по x , м	Смещение по y , м	Смещение по z , м
1	0,5	1,2	60	15	1,409	$0,743 - 0,1t$	$0,580 - 0,2t$
2	0,4	0,7	70	25	0,771	$0,080 + 0,2t$	$1,603 - 0,3t$
3	0,4	1,2	115	25	$0,646 + 0,2t$	0,742	$0,121 + 0,3t$
4	0,4	0,7	45	10	0,9722	$0,404 - 0,2t$	$1,634 - 0,3t$
5	0,3	0,8	115	25	0,4715	$0,881 + 0,3t$	$0,679 - 0,2t$
6	0,8	0,6	30	25	1,236	$0,146 + 0,2t$	$0,596 - 0,2t$
7	0,9	0,24	40	25	$1,559 - 0,2t$	0,172	$0,641 - 0,2t$
8	0,3	0,66	38	25	1,07	$0,090 + 0,3t$	$1,371 - 0,3t$
9	0,74	0,8	35	15	$1,378 - 0,3t$	0,631	$1,281 - 0,2t$
10	0,3	0,36	80	5	1,128	$0,389 - 0,3t$	$0,733 - 0,2t$
11	0,28	0,34	50	40	1,3213	$-0,012 + 0,3t$	$1,226 - 0,2t$
12	0,6	0,8	35	15	$1,264 - 0,2t$	0,1371	$1,405 - 0,2t$
13	0,8	0,6	35	15	1,234	$0,303 + 0,2t$	$-0,201 + 0,2t$
14	0,46	0,82	70	15	$0,949 + 0,2t$	0,220	$0,544 + 0,3t$
15	0,5	0,3	125	25	0,5289	$0,789 - 0,4t$	$0,451 + 0,2t$

1.4 Методические указания по выполнению работы

На первом этапе определяется текущее положение манипулятора, для чего решается система уравнений по теореме косинусов. Система уравнений определяется следующим образом:

$$\begin{cases} x = a \cdot \cos\varphi + b \cdot \cos\psi \\ y = a \cdot \sin\varphi + b \cdot \sin\psi \end{cases} \quad (1)$$

где x, y – расстояния относительно нулевой точки по соответствующим осям;

a – длина начального звена;

b – длина конечного звена;

φ_0 – угол основания манипулятора;

ψ_0 – угол между звеньями.

Для нахождения скоростей необходимо провести дифференцирование полученных систем уравнений по времени.

Расчет необходимо выполнить в MS Word, продублировав его в MS Excel с целью проверки. Результаты привести в таблице.

1.5 Пример

На рисунке 1 изображен в начальном положении механизм манипулятора и показаны положительные направления отсчета углов φ, ψ, θ и расстояние S . Размеры звеньев: $a = 1,30$ м, $b = 1,10$ м, $c = 0,55$ м. Захват (точка A) движется в течении 1 с согласно уравнениям:

$$\begin{cases} x_A(t) = 1,5329 - 0,2t, \\ y_A(t) = 0,5487 - 0,089t \end{cases} .$$

В начальный момент времени ($t = 0$), $\varphi = \varphi_0 = 62^\circ$, $\psi = \psi_0 = 33^\circ$. Определить значения неизвестного угла θ и расстояние S . Вычислить также угловые скорости $\dot{\varphi}_0$, $\dot{\psi}_0$, $\dot{\theta}_0$ и относительную скорость \dot{S}_0 .

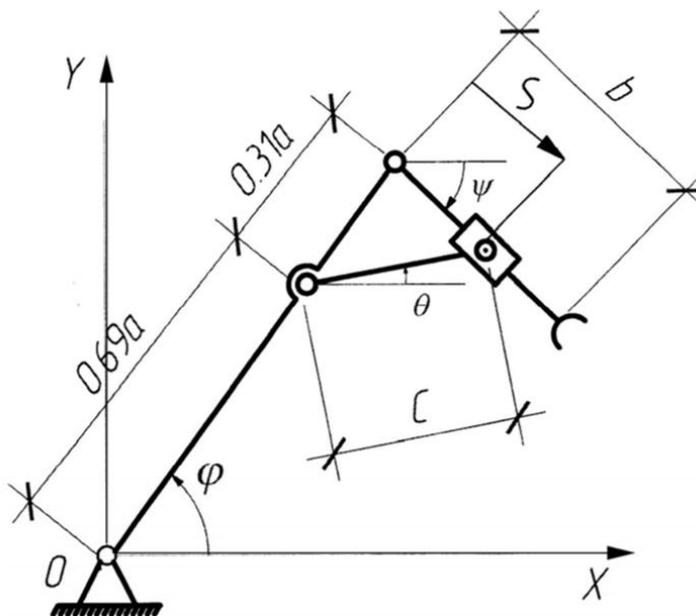


Рисунок 1 – Кинетическая схема манипулятора

Для любого положения механизма справедливы следующие векторные соотношения (рисунок 2):

$$\overline{OA} = \overline{OC} + \overline{CA},$$

$$\overline{DB} = \overline{DC} + \overline{CB},$$

где

$$|\overline{OC}| = a = \text{const},$$

$$|\overline{CA}| = b = \text{const},$$

$$|\overline{DC}| = 0,31a = \text{const},$$

$$|\overline{DB}| = c = \text{const},$$

$$|\overline{CB}| = S.$$

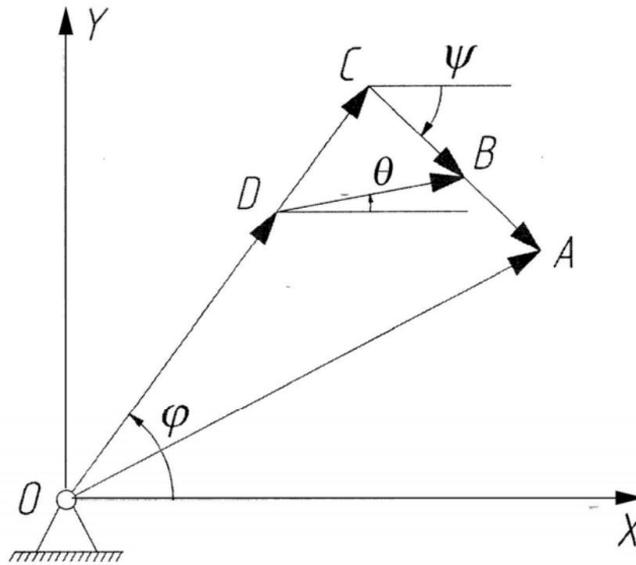


Рисунок 2 – Векторная схема манипулятора

Спроецируем полученные равенства на оси координат:

$$\begin{cases} -a\dot{\varphi}_0 \sin \varphi_0 - b\dot{\psi}_0 \sin \psi_0 = -0,2 \\ a\dot{\varphi}_0 \cos \varphi_0 - b\dot{\psi}_0 \cos \psi_0 = -0,089 \end{cases}$$

$$\begin{cases} (-0,31a\dot{\varphi}_0 \sin \varphi_0 - S_0\dot{\psi}_0 \sin \psi_0) + \dot{S}_0 \cos \varphi_0 + c\dot{\theta}_0 \sin \theta_0 = 0 \\ (0,31a\dot{\varphi}_0 \cos \varphi_0 - S_0\dot{\psi}_0 \cos \psi_0) - \dot{S}_0 \sin \varphi_0 + c\dot{\theta}_0 \cos \theta_0 = 0 \end{cases}$$

Для определения расстояния используем следующую систему уравнений:

$$\begin{cases} c \cos \theta_0 = 0,31a \cos \varphi_0 + S_0 \cos \psi_0 \\ c \sin \theta_0 = 0,31a \sin \varphi_0 - S_0 \sin \psi_0 \end{cases}$$

Здесь неизвестными являются S_0 и θ_0 . Возведём в квадрат оба выражения и сложим соответственно их левые и правые части:

$$c^2 = (0,31a)^2 + 0,62S_0a \cos \varphi_0 \cos \psi_0 - 0,62S_0a \sin \varphi_0 \sin \psi_0 + S_0^2;$$

$$c^2 = (0,31a)^2 + 0,62S_0a \cos(\varphi_0 + \psi_0) + S_0^2.$$

В результате получено квадратное уравнение для определения начальной длины S . Для большей наглядности представим его в привычном виде:

$$S_0^2 + [0,62a \cos(\varphi_0 + \psi_0)]S_0 + [(0,31a)^2 - c^2] = 0,$$

$$S_0^2 = -[0,31a \cos(\varphi_0 + \psi_0)] + \sqrt{[0,31a \cos(\varphi_0 + \psi_0)]^2 - [(0,31a)^2 - c^2]}.$$

Решив квадратное уравнение при заданных параметрах, получим: $S_0 = 0,4102$ м.

Из предыдущей системы уравнений получим:

$$\operatorname{tg} \theta_0 = \frac{0,31a \sin \varphi_0 - S_0 \sin \psi_0}{0,31a \cos \varphi_0 + S_0 \cos \psi_0}.$$

Откуда $\operatorname{tg} \theta_0 = 0,2483$; $\theta_0 = 13,946^\circ$.

Подставляя значения a , b , c , φ_0 , ψ_0 , θ_0 , получим следующую систему уравнений:

$$\begin{cases} -1,1478\dot{\varphi}_0 - 0,5991\dot{\psi}_0 = -0,5 \\ 0,6301\dot{\varphi}_0 - 0,9225\dot{\psi}_0 = 0,089 \end{cases}$$
$$\begin{cases} 0,8337\dot{S}_0 + 0,1326\dot{\theta}_0 = 0,3558\dot{\varphi}_0 + 0,2234\dot{\psi}_0 \\ -0,5446\dot{S}_0 - 0,5338\dot{\theta}_0 = -0,1892\dot{\varphi}_0 + 0,344\dot{\psi}_0 \end{cases}$$

Решая эту систему уравнений, получим итоговые значения скоростей:

$\dot{\varphi}_0 = 0,0921$ рад/с, $\dot{\psi}_0 = 0,1574$ рад/с, $\dot{\theta}_0 = -0,1806$ рад/с, $\dot{S}_0 = 0,1095$ м/с.

2 Лабораторная работа №2. Имитационное моделирование мехатронной системы

2.1 Цель работы

Цель работы заключается в построении имитационной модели разрабатываемой мехатронной системы.

2.2 Краткие теоретические сведения

Имитационное моделирование (ИМ) – это метод исследования, заключающийся в имитации на ЭВМ (с помощью комплекса программ) процесса функционирования системы или отдельных ее частей и элементов. Сущность метода ИМ заключается в разработке таких алгоритмов и программ, которые имитируют поведение системы, ее свойства и характеристики в необходимом для исследования системы составе, объеме и области изменения ее параметров. При ИМ реализующий модель алгоритм воспроизводит процесс функционирования системы во времени, причем имитируются явления, составляющие процесс, с сохранением их логической структуры и последовательности протекания во времени, что позволяет по исходным данным получить сведения о состояниях процесса в определенные моменты времени, дающие возможность оценить характеристики системы.

ИМ позволяет осуществлять многократные испытания модели с нужными входными данными, чтобы определить их влияние на выходные критерии оценки работы системы. При таком моделировании компьютер используется для численной оценки модели, а с помощью полученных данных рассчитываются ее реальные характеристики. Имитационная модель отображает стохастический процесс смены дискретных состояний системы. При реализации модели

на компьютере производится накопление статистических данных по показателям модели, которые являются предметом исследований.

По окончании моделирования накопленная статистика обрабатывается, и результаты моделирования получаются в виде выборочных распределений исследуемых величин. Таким образом, математическая статистика и теория вероятностей являются математическими основами имитационного моделирования.

Имитационная модель может представить объект практически любой сложности. Ограничениями могут служить лишь недостаточная квалификация исполнителя, а также требование адекватности модели и достижения очень большой точности результата. Это связано с получением статистических выборок большого объема, что ведет к необходимости получения большого числа реализаций модели и наличия высокопроизводительных компьютеров [2].

2.3 Задание к лабораторной работе

Для своего варианта (таблица 1) выполнить имитационное моделирование мехатронной системы. Дополнительные данные для выполнения задания приведены в таблице 2.

Таблица 2 – Варианты заданий

Вариант	Продолжительность поворота муфты, с	Перегрузка приемного устройства, с	Цикл захвата детали, с
1	2	100	1
2	2	120	1
3	2	140	1
4	2	160	2
5	2	180	2
6	3	100	2
7	3	120	3
8	3	140	3
9	3	160	3
10	3	180	4
11	4	100	4
12	4	120	4
13	4	140	5
14	4	160	5
15	4	180	5

2.4 Методические указания по выполнению работы

Имитационная модель отображается при помощи системы имитационного моделирования GPSS World Student Version.

Для выполнения поставленного задания необходимо:

- описать все необходимые операторы при имитационном моделировании мехатронной системы;
- отобразить все значения, применяемые при моделировании;
- написать код программы моделирования на языке GPSS.

2.5 Пример

Примем продолжительность поворота муфты 2 с, перегрузка приемного устройства после его заполнения – 180 с, цикла захвата детали – 1 с.

Режим работы пневмозахвата обозначаем ZAHVAT, загрузочное устройство (конвейер) ZNU, приемное устройство PRU, промежуточный приемный стол PRS, манипулятор PRM. При составлении алгоритма моделирования участков введем следующие обозначения: К – количество деталей, n – счетчик тары под готовые детали (так как емкость тары под детали – 2100, а кассеты – 300 деталей, то $n=1, 2, \dots, 7$). С учетом коэффициента использования участка и сменности его работы модельное время будет $=28800$ с.

В связи с тем, что продолжительность цикла захвата и отпускания одной детали составляет 1 с, для увеличения точности принимаем за единицу модельного времени 0,01 с. При разработке программы для организации цикла используем оператор TRANSFER в режиме безусловной передачи. При моделировании счетчиков используются операторы SAVEVALUE в режимах замещения и приращения, а также оператор TEST.

```
SIMULATE  
VAR VARIABLE 100
```

	GENERATE	, , , 1
	ASSIGN	1, 2880000
MET1	SEIZE	PRM
	ADVANCE	700
	RELEASE	PRM
MET2	SAVEVALUE	1+, 1
	TEST	X1, 300, MET 3
	SEIZE	ZNU
	ADVANCE	200
	RELEASE	ZNU
	SAVEVALUE	1, 0
	SAVEVALUE	2+, 1
	TEST E	X2, 7, MET 1
	SEIZE	PRU
	ADVANCE	18000
	RELEASE	PRU
	SAVEVALUE	2, 0
	TRANSFER	, MET 1
MET3	SEIZE	PRM
	ADVANCE	700
	RELEASE	PRM
	SEIZE	ZAHVAT
	ADVANCE	V\$VAR
	RELEASE	ZAHVAT
	TEST LE	M1, P1, MET 4
	TRANSFER	, MET 2
MET4	TERMINATE	1
	START	1
	END	

3 Лабораторная работа №3. Составление циклограмм работы мехатронной системы

3.1 Цель работы

Цель работы заключается в построении циклограммы работы разрабатываемой мехатронной системы.

3.2 Краткие теоретические сведения

Циклограмма – это диаграмма, показывающая время и последовательность выполнения операций в процессе.

Построение временной циклограммы работы проектируемого комплекса выполняется с целью выявления несогласованности работы его элементов и определения коэффициентов загрузки оборудования. Циклограмма должна отражать работу каждой единицы оборудования: станков, роботов, накопителей, транспортных устройств, склада, контрольно-измерительных машин и т.д. за полный цикл работы комплекса.

Коэффициенты загрузки основного и вспомогательного оборудования определяются как отношение времени работы данной единицы оборудования к общему времени цикла работы оборудования.

Циклограмма последовательности работы механизмов и узлов оборудования, входящего в состав комплекса, является практически алгоритмом его работы и служит исходной информацией для создания системы управления всем участком.

Следует отметить отсутствие необходимости включать в циклограмму все механизмы комплекса, т.к. многие группы механизмов управляются самостоятельно от своих систем управления.

Главная особенность циклограмм последовательности состоит в том, что она не временная, а потактовая: каждый такт циклограммы отражает новое положение механизмов автоматической системы. При этом соблюдается строгая последовательность их срабатывания от такта к такту. Для удобства составления циклограммы, весь комплекс следует разбить на модули, а последние в свою очередь на отдельные виды оборудования.

В результате анализа построенной циклограммы даются рекомендации для разработки системы управления автоматическим комплексом и проектируется ее структурная схема [3].

3.3 Задание к лабораторной работе

Для своего варианта (таблицы 1 и 2) построить циклограмму работы оборудования.

3.4 Методические указания по выполнению работы

Для выполнения задания необходимо все операции, описанные в имитационной модели, разбить на конкретные этапы с указанием временных интервалов работы каждой единицы оборудования. Интервалы работы помечаются посредством окрашивания соответствующих ячеек в определенный цвет.

Циклограмма может быть построена в любом табличном редакторе (например, MS Excel).

3.5 Пример

Используя данные из прошлой лабораторной работы, получаем следующую циклограмму (таблица 3).

Таблица 3 – Циклограмма работы манипулятора мехатронной системы

№	Операция	1 с	2 с	3 с	4 с	5 с	6 с	7 с
1	Движение манипулятора в точку захвата							
2	Захват детали							
3	Поворот муфты							
4	Движение манипулятора в точку перемещения							
5	Отпускание детали							
6	Возврат муфты							
7	Возврат манипулятора в исходное положение							

4 Лабораторная работа №4. Подбор устройств и составление кинематической схемы работы мехатронной системы

4.1 Цель работы

Цель работы заключается в подборе необходимых устройств для реализации мехатронной системы, а также построении кинематической схемы манипулятора.

4.2 Краткие теоретические сведения

Принципиальная кинематическая схема – это такая схема, на которой показана последовательность передачи движения от двигателя через передаточный механизм к рабочим органам машины и их взаимосвязь.

В состав мехатронной машины входят пять основных частей:

- механическое устройство, конечным звеном которого является рабочий орган;
- исполнительные двигатели;
- силовые преобразователи;
- устройство компьютерного управления, на вход которого поступают команды человека-оператора либо ЭВМ верхнего уровня управления;
- информационное устройство, предназначенное для получения и передачи в устройство компьютерного управления данных о реальном движении машины и о фактическом состоянии ее подсистем.

Механическое устройство и двигатели объединены в группу исполнительных устройств. В состав группы интеллектуальных устройств включены электронная (силовые преобразователи), управляющая и информационная части машины.

Устройством компьютерного управления будем называть комплекс аппаратных и программных средств, вырабатывающий сигналы управления для блока приводов машины. В состав этого комплекса обычно входят задающие устройства (например, джойстики и рукоятки), пульт управления оператора, вычислительные и преобразующие устройства, периферийные устройства ввода-вывода информации.

Устройство компьютерного управления выполняет следующие основные функции:

- управление функциональными движениями мехатронной машины в реальном масштабе времени;
- координация управления механическим движением с сопутствующими внешними процессами;
- взаимодействие с человеком-оператором через человеко-машинный интерфейс в режимах программирования (режим off-line) и непосредственно в процессе движения (режим on-line);
- обмен данными с внешними устройствами (информационным устройством, блоком приводов, компьютером верхнего уровня, периферийными устройствами).

Информационное устройство предназначено для сбора и передачи в устройство управления информации о фактическом состоянии внешней среды и движущейся мехатронной машины.

Механическое устройство мехатронной машины представляет собой многозвенный механизм, кинематическую цепь которого образуют движущиеся звенья, составляющие кинематические пары. Конечным звеном кинематической цепи является рабочий орган [4].

4.3 Задание к лабораторной работе

Для своего варианта (таблица 1) подобрать оборудование и построить кинематическую схему манипулятора.

4.4 Методические указания по выполнению работы

Исходя из данных своего варианта в сети интернет подобрать:

- захватное устройство для манипулятора (усилие захвата не менее 100 N);
- сервомоторы и редукторы для механических звеньев (средняя или низкая инерция);
- датчики положения манипулятора (любой датчик, применимый к промышленной автоматике);

Кинематическая схема строится при помощи средств рисования MS Word или же в специализированных программных продуктах.

4.5 Пример

Согласно представленному описанию подобрано подходящее оборудование (таблица 4).

Таблица 4 – Оборудование для мехатронной системы

№	Наименование оборудования	Основные характеристики
1	Пневматический параллельный захват, серия P5GB	P5GB - это пневматический параллельный захват двойного действия, доступный с диаметром отверстий от 12 до 32 мм. Усилие захвата от 10 до 170 Н.
2	Сервомотор Fukuta серия SF-80	Конструкция сервомотора - средней и низкой инерции. Постоянное управление крутящим моментом может отвечать требованиям высокой точности движения.
3	Датчик положения Festo SMT-C1-PS-24V-K-0,3-M8D	Бесконтактный датчик положения с магнитоиндуктивным принципом измерения. Имеет блочную конструкцию. Подходит для манипуляторов промышленной автоматики в любой сфере производства.

Кинематическая схема построена при помощи средств рисования MS Word (рисунок 3).

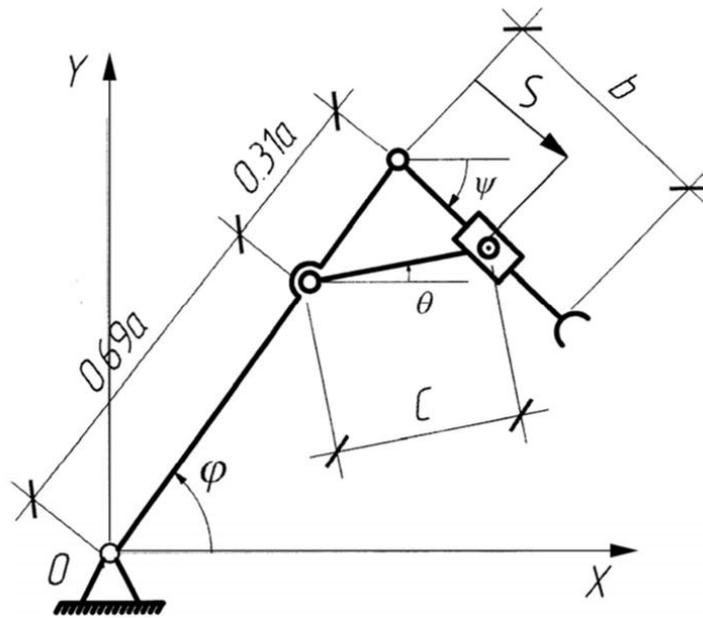


Рисунок 3 – Кинематическая схема

5 Лабораторная работа №5. Составление структурной схемы мехатронной системы

5.1 Цель работы

Цель работы заключается в построении структурной схемы разрабатываемой мехатронной системы.

5.2 Краткие теоретические сведения

Структурная схема представляет собой совокупность элементарных звеньев объекта и связей между ними, формируя, таким образом, один из видов графической модели.

Структурная схема разрабатывается на начальных стадиях проектирования и предшествует разработке других схем. Структурная схема определяет основные функциональные части изделия, их назначение и взаимосвязи между ними. Схема отображает принцип действия изделия в самом общем виде.

Действительное расположение составных частей на структурной схеме не учитывают и способ связи не раскрывают. Построение схемы должно давать наглядное представление о составе изделия и последовательности взаимодействия функциональных частей в изделии. Функциональные части на схеме изображают в виде прямоугольников или условных графических обозначений. При изображении функциональных частей в виде прямоугольников их наименования, типы и обозначения вписывают внутрь прямоугольников.

Направление хода процесса, происходящего в изделии, обозначают стрелками, соединяющими функциональные части. На схемах простых изделий функциональные части располагают в виде цепочки в соответствии с ходом рабочего процесса в направлении слева направо. Схемы, содержащие несколько

основных рабочих каналов, рекомендуется вычерчивать в виде параллельных горизонтальных строк.

При большом количестве функциональных частей в изделии элементы структурной схемы могут обозначаться порядковыми номерами. При этом должен быть составлен перечень этих функциональных частей. В этом случае наглядность структурной схемы ухудшается, т.к. роль каждой функциональной части выясняется не только по изображению, но и с помощью перечня.

Для сложных изделий, состоящих из нескольких функциональных частей, для каждой части также могут быть разработаны их структурные схемы.

На структурной схеме допускается указывать характеристики функциональных частей, поясняющие надписи и диаграммы, определяющие последовательность процессов во времени, а также параметры в характерных точках (величины токов, напряжений, формы и величины импульсов и др.). Данные помещаются рядом с графическим обозначением или на свободном поле схемы. После окончания проектирования структурная схема изделия включается в эксплуатационную документацию для общего ознакомления обслуживающего персонала с изделием [5].

5.3 Задание к лабораторной работе

Для своего варианта (таблица 1) разработать структурную схему мехатронной системы.

5.4 Методические указания по выполнению работы

Все основные элементы структурной схемы разработаны при имитационном моделировании и в процессе построения кинематической схемы. При построении четко указать состав блоков и направление связей.

Структурная схема строится при помощи средств рисования MS Word.

5.5 Пример

На рисунке 4 показана структурная схема системы управления движением звеньев двухзвенного манипулятора, в котором выделены только переносные координаты.

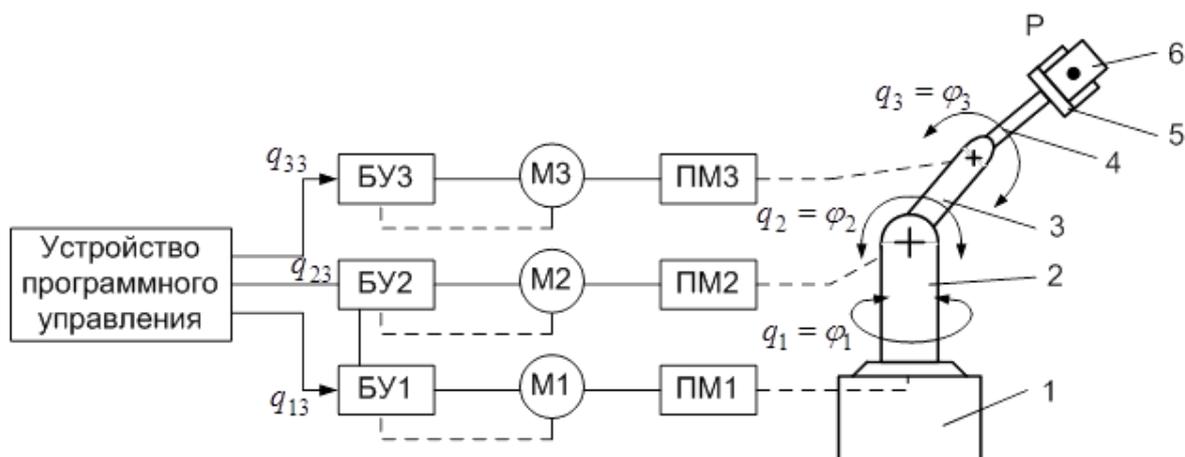


Рисунок 4 – Структурная схема манипулятора

Задающие воздействия q_{13} , q_{23} , q_{33} поступают на блоки управления БУ1 ... БУ3 электроприводами М1 ... М3.

Передаточные механизмы ПМ1 ... ПМ3 обеспечивают реализацию требуемого вида движения (линейного или углового) звеньев. Это движение определяется обобщенными координатами q_1 , q_2 , q_3 . Датчики положения звеньев можно устанавливать на электродвигателях и звеньях манипулятора.

В манипуляторе на неподвижном основании 1 установлена муфта 2, совершающая угловые движения φ_1 вокруг вертикальной оси, а звенья 3 и 4 совершают угловые движения φ_2 и φ_3 вокруг осей, перпендикулярных плоскости рисунка. Цифрами 5 и 6 обозначены, соответственно, захват и груз.

6 Лабораторная работа №6. Составление электрической и пневматической схем автоматизации системы

6.1 Цель работы

Цель работы заключается в построении электрической и пневматической схем автоматизации разрабатываемой мехатронной системы.

6.2 Краткие теоретические сведения

Электрическая схема – графическое изображение электрической цепи, содержащее условные обозначения ее элементов и способы их соединения.

По количеству источников ЭДС схемы могут быть с одной или несколькими ЭДС.

По конфигурации (топологии) цепи могут быть неразветвленными (последовательными, одноконтурными) и разветвленными (параллельными, смешанными, многоконтурными).

Электрические цепи и их элементы могут работать в режимах: номинальном, для которого они предназначены; холостого хода (ток в источнике электроэнергии или ее приемнике равен нулю); короткого замыкания (выводы источника электроэнергии или ее приемника замкнуты между собой); переходном (при переходе цепи из одного установившегося состояния в другое, например, при подключении участка цепи к источнику электропитания или отключении его от нее) и согласованном (при котором на нагрузке (приемнике) цепи выделяется максимальная мощность).

Пневматическая схема – это технический документ, содержащий в виде условных графических изображений или обозначений информацию о строении изделия, его составных частях и взаимосвязи между ними, действие которого основывается на использовании энергии сжатой жидкости (газа).

Разрабатываемая документация принципиальной пневматической схемы питания сжатым воздухом включает в себя всех потребителей этого вида энергии и служит основным документом для разработки соответствующих рабочих чертежей и осуществления нормальной эксплуатации этих технических средств.

Воздухопроводы на принципиальной пневматической схеме питания позиционного обозначения или маркировки не имеют. Однако на всех коллекторах и воздухопроводах от источника питания до последнего запорного органа на ответвлениях к пневмоприемникам должен быть указан размер условного прохода трубы. На коллекторах дополнительно указывается также давление сжатого воздуха.

Для улучшения чтения принципиальных пневматических схем и для более быстрого нахождения возникающих неисправностей эти схемы оформляются аналогично электрическим [6].

6.3 Задание к лабораторной работе

Для своего варианта (таблица 1) разработать электрическую и пневматическую схемы работы мехатронной системы.

6.4 Методические указания по выполнению работы

Электрическая и пневматическая схемы строятся в соответствии ГОСТ [7, 8]. Схемы строятся при помощи программного пакета RastrWin.

Программа RastrWin разработана коллективом авторов Уральского политехнического института (г. Екатеринбург, Россия) и предназначена для расчета и анализа установившихся режимов электроэнергетических систем на персональных компьютерах с установленными операционными системами Windows XP, Vista, 7, 8, 10. Предельный объем схемы полной версии программы 32000 узлов. Студенческая версия RastrWin, которая используется

для выполнения лабораторных работ, имеет ограничения по числу узлов, максимальное количество которых не может превышать 60.

Начало работы с RastrWin. После открытия программы RastrWin (ярлык с соответствующим названием на *Рабочем столе* компьютера) открывается окно *Рабочей области*, на котором в отдельных окнах отображается содержимое загруженных файлов (таблицы режимов, графика и т.д.).

Для загрузки в *Рабочую область* существующего на диске режима сети необходимо выполнить команду *Загрузить...* в пункте *Файлы* главного меню программы и выбрать нужный файл с расширением *.rg2*.

Перед вводом новой схемы нужно выполнить команду *Новый* в пункте *Файлы* главного меню и отметить галочкой тип файла *режим.rg2* (*ФайлыНовый режим.rg2*). После этого целесообразно сразу сохранить вновь созданный файл режима (*Файлы – Сохранить как...*) в рабочем каталоге студента под именем, удобном для восприятия и идентификации, и содержащем информацию о группе, фамилии студента, номере лабораторной работы, например, *Ts-30612_Ivanov_A.rg2*.

После сохранения файла режима нужно открыть два окна, содержащие пустые таблицы для ввода информации об узлах и ветвях сети (*Открыть – Узлы – Узлы* и *Открыть – Ветви – Ветви*), которые с помощью пункта *Окна* главного меню можно расположить на экране подходящим образом – каскадом, горизонтальной либо вертикальной мозаикой.

Открытые таблицы *Узлы* и *Ветви* содержат столбцы и строки, в которые заносится информация об узлах и ветвях. Каждый столбец (поле) соответствует определенному виду данных (название, номинальное напряжение, сопротивления и т.д.), а каждая строка является записью (набором данных) для каждого узла или ветви.

Для добавления в таблицу строк, их удаления и дублирования, необходимо использовать команды *Вставить*, *Добавить*, *Удалить*, *Дублировать* в пункте *Таблица* главного меню (добавление происходит в конец таблицы, а вставка – перед выделенной строкой).

Перемещаться по столбцам и строкам таблиц можно с помощью мыши и соответствующих клавиш клавиатуры, а переключаться между режимами просмотра и редактирования таблицы – щелчком мыши на выделенной ячейке, клавишами *Enter* и *F2*.

Заголовки столбцов таблиц содержат названия данных, которые необходимо занести в соответствующее поле (исходные данные), либо которые определены в процессе расчета режима сети (расчетные данные). Некоторые из нужных/ненужных столбцов данных могут отсутствовать/присутствовать в таблицах, что зависит от настроек программы. Сделать видимыми или скрыть столбцы с необходимой информацией (исходной либо расчетной) на время текущего сеанса работы с данным файлом можно с помощью команд *Скрыть*, *Заменить на*, *Вставить за*, *Вставить до* в меню, вызываемого нажатием правой кнопки мыши на заголовке соответствующего столбца.

Ширину столбцов в таблицах и точность вывода числовой информации в полях можно менять с помощью полей *Ширина* и *Точность* диалогового окна, появляющегося после нажатия правой кнопкой мыши на заголовке нужного столбца и последующего выбора пункта меню с названием столбца (первый пункт меню). При этом менять *Заголовок* и *Описание* столбца данных не желательно.

Изменять ширину столбцов на время данного сеанса работы с загруженным файлом можно также с помощью мыши, изменяя границы столбца в области его заголовка. Однако эта операция не позволяет сохранить проведенные изменения после выхода из программы или открытия нового файла.

Быстрым двойным нажатием левой кнопки мыши на заголовке столбца информацию в столбцах можно сортировать по возрастанию или убыванию. Переместить столбец таблицы позволяет повторное нажатие левой кнопкой мыши на его заголовке с последующим ее удержанием.

Чтобы изменения в настройках не исчезли после открытия другого файла или выхода из программы, необходимо их сохранить.

Для сохранения добавленных или удаленных из таблиц столбцов нужно после нажатия правой кнопки мыши на заголовке любого столбца выбрать пункт меню *Заполнить* и нажать *ОК*, затем выполнить команду *Файлы – Настройки программы – Формы*, в диалоговом окне *Формы* из списка *Формы* выбрать название таблицы (*Узлы* или *Ветви*), последовательно нажать *Применить* и *Сохранить*, и при всех выбранных галочках в появившемся окне нажать *ОК*, после чего нажать *Заккрыть*.

Если необходимо сохранить ширину столбцов и точность вывода данных, нужно выполнить команду *Сохранить шаблон* в пункте *Файлы* главного меню.

Ввод схемы сети рекомендуется начинать с данных по узлам. Минимально необходимым набором информации для каждого узла сети является:

O – отметка узла (используется для сортировки, выборки, эквивалентирования и т.д.); в данном поле пусто, если узел не отмечен;

S – состояние узла (включен/отключен); во включенном состоянии узла поле является пустым;

Тип – тип узла (*База*, *Нагр*, *Ген+*, *Ген-*); определяется программой автоматически за исключением базисного (балансирующего) узла, в качестве которого необходимо задать шину энергосистемы, выбрав в данной ячейке вариант *База*;

Номер – номер узла; задается в соответствии со схемой замещения сети;

Название – задается название узла;

U_ном – задается номинальное напряжение узла, [кВ];

P_н, Q_н – активная и реактивная мощности нагрузки в узле; заносятся величины P_к, Q_к [МВт] и [Мвар];

V_зд – модуль фиксированного напряжения в узле; задается и фиксируется напряжение U_к, [кВ];

Delta – расчетный угол напряжения в узле; для базисного (балансирующего) узла задается и фиксируется величина δ_k , для остальных узлов рассчитывается программой, [град].

Основными полями, в которые заносятся исходные данные по ветвям сети, являются:

O – отметка ветви (используется для сортировки, выборки, эквивалентирования и т.д.); в данном поле пусто, если ветвь не отмечена;

S – состояние ветви (включена/отключена); во включенном состоянии ветви поле является пустым; ветвь может быть отключена в начале, в конце или с обеих сторон;

Тип – тип ветви (*ЛЭП, Tr-p*); определяется программой автоматически по значению ячейки, в которую заносится коэффициент трансформации ветви;

N_нач, N_кон – номера узлов, которыми ограничена ветвь; для трансформатора начало ветви N_нач – это обязательно тот узел, к напряжению которого приведены его параметры (как правило, это напряжение обмотки высшего напряжения);

Название – название ветви; задается программой автоматически по известным названиям узлов, ограничивающих данную ветвь;

N_п – номер параллельной ветви; задается, если несколько линий или трансформаторов работают параллельно;

R, X – активное и реактивное сопротивления ветви; задаются величины $R_{ij}, X_{ij}, R_{jm}, X_{jm}, R_{jn}, X_{jn}$ [Ом].

После занесения исходной информации в таблицы *Узлы* и *Ветви* созданный файл режима необходимо сохранить.

Перед расчетом установившегося режима сети при необходимости нужно настроить параметры расчета, для чего через главное меню программы должна быть открыта таблица *Режим (Расчеты – Параметры... – Режим)*. Значение параметра *Точность расчета (dP)* (задается в МВт) необходимо корректировать в зависимости от мощностей в узлах сети, поскольку при

точности, соизмеримой с нагрузками, результаты расчета могут оказаться некорректными. Значения остальных параметров из таблицы *Режим* можно оставить принятыми по умолчанию, и изменять, только если возникли проблемы при расчете режима (некорректно заданы исходные данные, режим расходится и т.п.).

Расчет режима сети производится после выполнения команды *Расчеты – Режим* или нажатия клавиши *F5*.

При аварийном завершении расчета (режим разошелся) в появившемся окне *Протокол* с поэтапным описанием итерационного процесса будет содержаться краткое описание ошибки, отмеченное красным знаком. Кроме того, в данном случае программа автоматически может произвести изменения в таблицах *Узлы* и *Ветви* (отключить некоторые узлы или ветви), которые необходимо устроить после исправления ошибок, приведших к аварийному завершению расчета. После коррекции исходных данных и/или настроек программы расчет режима нужно повторить.

Если расчет завершился успешно (режим сошелся), в окне *Протокол* не будет сообщений об ошибках. Значение поля *Max. неб.* в последней строке таблицы сходимости окна *Протокол* не должно быть выше значения параметра *Точность расчета (dP)* в таблице *Режим*.

Чтобы распечатать результаты расчета режима сети, отображенные в таблицах *Узлы* и *Ветви*, необходимо сделать активным соответствующее окно, в диалоговом окне *Параметры страницы (Файлы – Настройки принтера...)* выбрать размер бумаги А4 и альбомную ориентацию страницы, после чего выполнить команду *Файлы – Печать...*, в открывшемся окне выбрать нужный принтер и нажать *ОК*.

6.5 Пример

В рассматриваемом варианте манипулятора электрическая схема будет выглядеть, согласно рисунку 5.

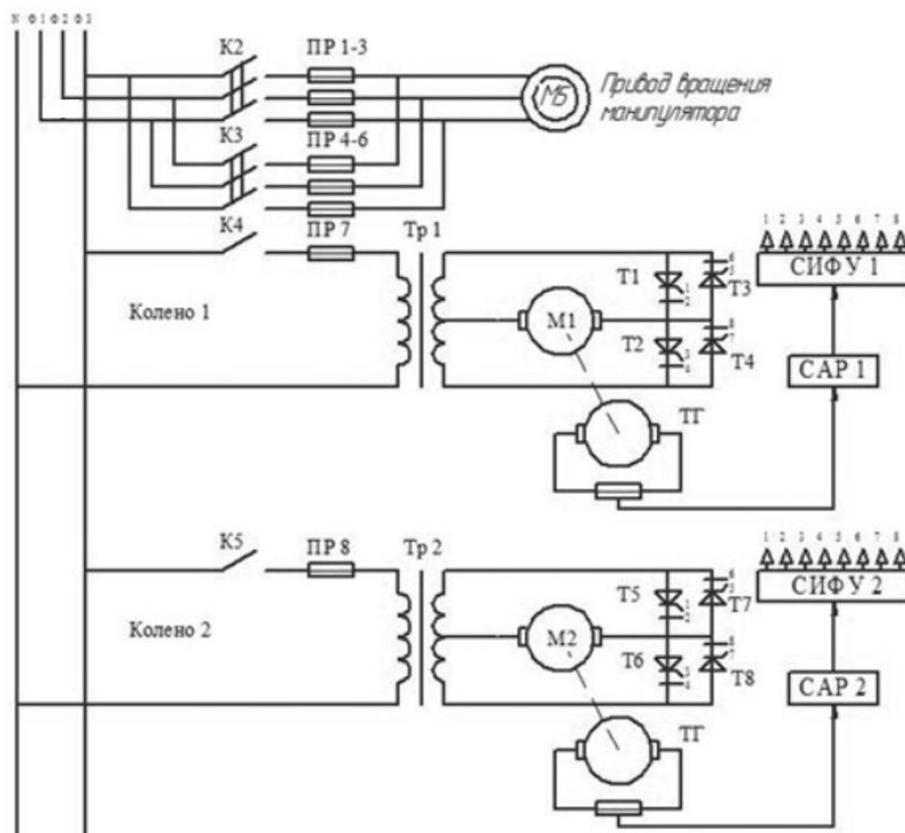


Рисунок 5 – Электрическая схема манипулятора

Условные обозначения:

K1...K5 – электромеханические контакторы;

Пр1...Пр8 – предохранители;

M1 ...M2 – электродвигатель постоянного тока;

M5 – асинхронный электродвигатель;

Tr1...Tr2 – трансформатор однофазный;

T1...T8 – тиристоры;

СИФУ – система импульсно-фазового управления тиристорами;

САР – система автоматического регулирования.

Пневматическая схема захвата отображена на рисунке 6.

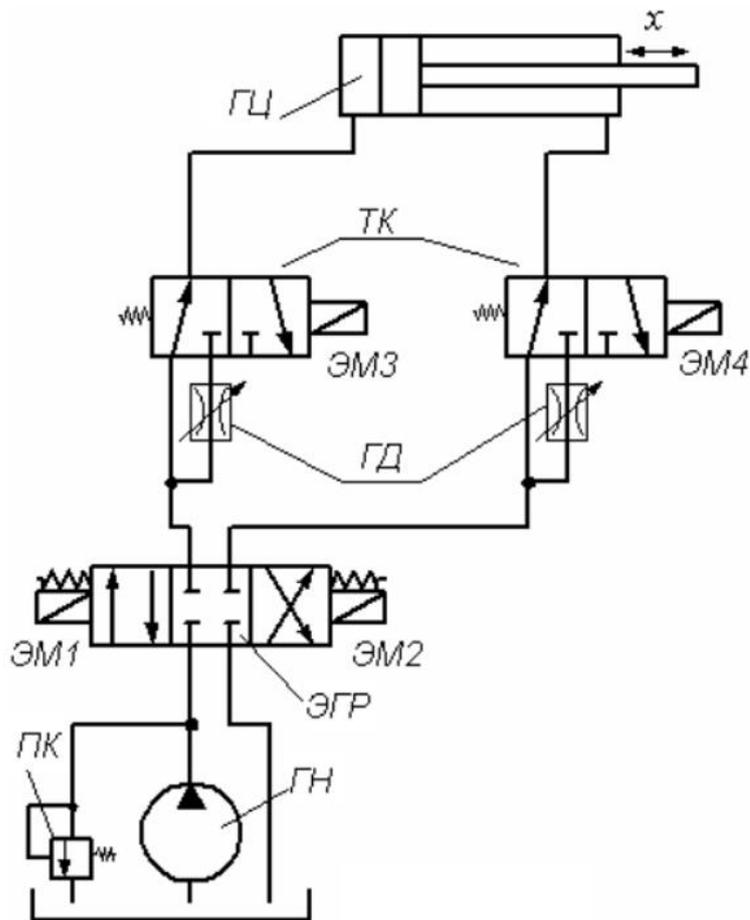


Рисунок 6 – Пневматическая схема захвата

Условные обозначения:

ГЦ – гидродвигатель;

ТК – распределители;

ГД – дроссель;

ЭГР – электрогидрораспределитель;

ПК – предохранительный клапан;

ГН – компрессор.

7 Лабораторная работа №7. Составление схемы алгоритма программы контроллера мехатронной системы

7.1 Цель работы

Цель работы заключается в построении схемы алгоритма программы контроллера разрабатываемой мехатронной системы.

7.2 Краткие теоретические сведения

Контроллер – специализированное техническое устройство, предназначенное для управления другими устройствами путем получения информации в виде цифровых данных или аналого-дискретного сигнала от внешнего устройства (ЭВМ, датчики или иное устройство), преобразования этой информации по специальному алгоритму и выдачи управляющих воздействий в виде цифрового или аналого-дискретного сигнала. Чаще всего контроллеры представляют собой программируемые устройства, имеющие в своем составе программируемые логические интегральные схемы (ПЛИС) или специализированные процессоры.

Программируемые контроллеры предназначены для циклового программного управления станками и различным технологическим оборудованием, оснащенным датчиками и исполнительными устройствами, которые работают по двухпозиционному принципу: «включен-выключен».

В зависимости от исполнения этот контроллер позволяет управлять 16, 32 или 48 устройствами. Количество входных цепей для подключения датчиков соответствует числу выходов. Каждый вход и выход имеет свой адрес.

Контроллер обеспечивает управление исполнительными устройствами, прием информации от датчиков состояния оборудования, формирование вы-

держек времени, организацию условных и безусловных переходов по программе управления, а также реализует другие функции.

Проектирование управляющего устройства, как правило, сводится к двум этапам: 1 – составление схемы подключения датчиков и исполнительных устройств к контроллеру, 2 – составление управляющей программы по схеме алгоритма.

Для удобства и наглядности отдельные действия алгоритма принято записывать в различные геометрические фигуры.

Последовательность выполнения шагов (направление обхода алгоритма) принято считать сверху – вниз и слева – направо. Если эта последовательность нарушается, то направление принято указывать стрелками [9].

7.3 Задание к лабораторной работе

Для своего варианта (таблица 1) разработать схему алгоритма работы контроллера (микроконтроллера) для мехатронной системы.

7.4 Методические указания по выполнению работы

Последовательность принятия решений программой определяется структурной и функциональной схемами оборудования. Общие принципы построения схем применимы и для схем алгоритмов.

Контроллер должен обрабатывать информацию датчика распознавания цвета и датчика положения манипулятора.

Для безошибочной работы программы необходимо, также, встроить счетчик перемещаемых деталей. Окончание программы достигается при перемещении 10 деталей на другую ленту.

Схема алгоритма строится при помощи средств рисования MS Word или же в специализированных программных продуктах.

7.5 Пример

Разработаем схему алгоритма для манипулятора (рисунок 7).

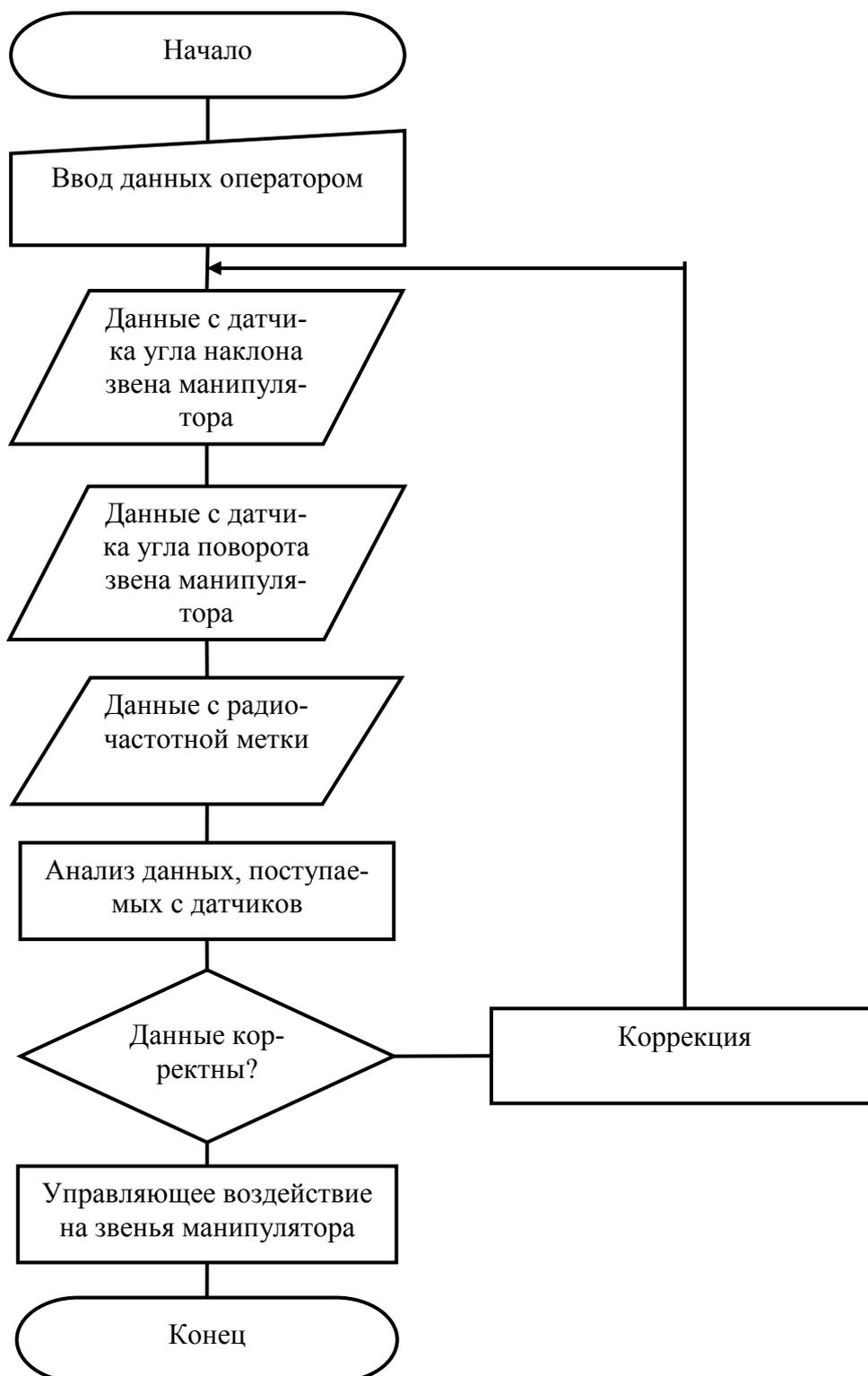


Рисунок 7 – Схема алгоритма работы контроллера

8 Лабораторная работа №8. Программирование контроллера мехатронной системы

8.1 Цель работы

Цель работы заключается в написании программы для контроллера разрабатываемой мехатронной системы.

8.2 Краткие теоретические сведения

Язык C++ представляет собой набор команд, которые говорят компьютеру, что необходимо сделать. Этот набор команд, обычно называется исходный код или просто код. Командами являются или «функции» или «ключевые слова». Ключевые слова (зарезервированные слова C/C++) являются основными строительными блоками языка. Функции являются сложными строительными блоками.

Такая структура функций напоминает содержание книги. Содержание может показывать главы книги, каждая глава в книге может иметь своё собственное содержание, состоящее из пунктов, каждый пункт может иметь свои подпункты. Хотя C++ предоставляет много общих функций и зарезервированных слов, которые можно использовать, все-таки возникает потребность в написании своих собственных функций.

Каждая программа в C++ имеет одну функцию, её называют главной или main-функция, выполнение программы начинается именно с этой функции. Из главной функции, также можно вызывать любые другие функции, неважно, являются ли они написанными вручную, или предоставляются компилятором [10].

8.3 Задание к лабораторной работе

Для своего варианта (таблица 1) написать код программы работы контроллера на C++. Ознакомиться с основными функциями и командами данного языка. Выполнить компилирование программы, осуществить проверку на наличие ошибок.

8.4 Методические указания по выполнению работы

В качестве среды разработки под Windows необходимо использовать среду разработки программного обеспечения RAD Studio XE6 Enterprise.

RAD Studio XE6 представляет собой полномасштабное решение для разработки действительно «нативных» приложений для Windows, Mac, Android и iOS на основе единой базы исходных кодов.

Редакция RAD Studio XE6 Enterprise предназначена для разработчиков и групп разработки программного обеспечения (ПО), занимающихся созданием клиент-серверных, многоуровневых, облачных, веб-приложений для платформ Windows, Mac OS X, .NET, iOS и веб-решений. RAD Studio Enterprise обеспечивает удобство подключения к различным серверам баз данных и корпоративным источникам данных, а также предоставляет многоуровневую технологию DataSnap. Текущая версия Embarcadero RAD Studio XE6 объединяет Delphi XE и C++ Builder XE6 в единую интегрированную среду разработки.

В отличие от задач, работающих под управлением MS-DOS, почти каждая (за редким исключением) задача под Windows, во время выполнения, представлена на рабочем столе в виде окна и имеет хотя бы минимальные средства ведения диалога с пользователем. Оконное представление программ отвечает условиям, накладываемым на систему многозадачностью, так как несколько приложений одновременно не могут выводить свою информацию в одно и то же место.

В любой момент времени одно из окон «находится в фокусе» (является «активным») и потоки вводимой информации направляются операционной системой в программу, которой принадлежит это окно. И хотя подавляющая часть ресурсов встроена в саму операционную систему Windows, создание программ для Windows требует от разработчика написания огромного кода, описывающего окна программы, их элементы и взаимодействие этих окон и элементов. Избавиться от огромного труда при создании программ возможно, если использовать одну из визуальных систем программирования, например, Builder.

Все пользовательские программы в среде C++Builder называются приложениями (прилагаются к самой среде).

Полноценная работа в среде программирования C++Builder возможна, если пользователь уже имеет практические навыки в написании программ, причем, не простых программ, а программ, выполненных в стиле объектно-ориентированного и модульного программирования.

Такие программы по своим характеристикам сходны с программами MS-DOS и представляют собой утилиты — узкоспециализированные программы, нацеленные на решение, как правило, одной конкретной задачи. При запуске таких программ они отображаются в окнах, подобных окнам программ MS-DOS. Да и разработка подобных программ выполняется аналогично программам для ОС MS-DOS.

Запуск среды Embarcadero RAD Studio XE6 для разработки программы в C++ Builder XE осуществляется из главного меню системы Windows, которое вызывается по кнопке ПУСК на панели задач:

Пуск > Программы > Программирование > Embarcadero RAD Studio XE2 > C++ Builder XE.

После старта программы, система Embarcadero RAD Studio XE2 предоставляет пользователю рабочий стол, на котором расположена группа из нескольких окон.

После старта проекта центральное окно (окно 6) превращается в окно с закладками. И при работе с несколькими модулями (файлами и формами) про-

граммист открывает необходимые дизайнеры для каждой из форм и редакторы для каждого текстового модуля. То есть центральное окно содержит закладки для одной страницы приглашения, для одного или нескольких файлов проекта. Оно является многооконным, но переходить от окна к окну (в том числе и к странице приглашения) можно по закладкам.

Главное меню располагается, как и в любом окна Windows, под строкой заголовка окна. Назначение многих команд стандартно для приложений Windows.

Приведем технологию создания приложения в данной программе.

1 Создать новую отдельную папку для разрабатываемого проекта.

2 Запустить Embarcadero RAD Studio XE2 для работы в C++ Builder XE.

3 Создать новый проект: Выполнить команду File главного меню системы Embarcadero RAD Studio XE2;, перейти по вкладкам File >New > Other..., где необходимо выбрать иконку с именем Console Application и кликнуть Ok.

4 Сохранить новый проект: выполнить команду File>Save Project As...

5 Вписать текст программы в окно редактора кода File1.cpp. Выполнить компиляцию файла File1.cpp. Выполнить команду Project>Build File1.cpp или с помощью горячих клавиш Alt-F9.

6 Исправить ошибки компиляции, если они есть.

7 Запустить программу на выполнение: выполнить команду Run>Run Without Debugging (кнопка) или с помощью горячих клавиш Shift-Ctrl-F9 В случае «зависания» — команда Run/Program Reset!!

8 Если выявятся ошибки алгоритма или ошибки выполнения, внести исправления и повторить запуск программы.

8.5 Пример

Приведем пример программы работы контроллера для манипулятора

```
#include<iostream>  
using namespace std;
```

```

int main() {

    bool color = 0; // переменная отвечает за цвет детали
    int counter; // для подсчета количества операций
    bool starting_position; // стартовое положение манипулятора
    bool near_the_tape; // положение манипулятора возле ленты
    bool grab_detail; // захватить деталь
    bool near_the_table; // положение манипулятора возле стола
    bool let_go_detail; // отпустить деталь

    while (color == 1) {
        for (int count = 0; count < 10; count++) {
            if (color == 1 && counter < 10) {
                near_the_tape = 1; // движение манипулятора к
ленте
            }
            if (near_the_tape == 1) {
                grab_detail = 1; // манипулятор выполняет захват
детали
            }
            if (grab_detail == 1) {
                near_the_table = 1; // движение манипулятора с де-
талью, к столу
            }
            if (near_the_table == 1) {
                let_go_detail = 1; // манипулятор отпускает деталь
            }
            if (let_go_detail == 1) {

```

```
starting_position = 1; // манипулятор возвращается
```

в стартовое положение

```
    }  
    counter = count;  
  }  
}  
}
```

9 Лабораторная работа №9. Интеграция мехатронной системы в общую систему управления

9.1 Цель работы

Цель работы заключается в интеграции мехатронной системы в общую систему управления посредством разработки управляющей программы.

9.2 Краткие теоретические сведения

Для интеграции необходимо использовать систему SCADA.

SCADA-системы (Supervisory Control and Data Acquisition) - это системы диспетчерского контроля, управления и сбора данных о технологических процессах и работе участков, цехов и предприятия в целом.

SCADA-система состоит из компьютерной (виртуальной части) и физической. Компьютерная часть – это программы, мнемосхемы, базы данных, позволяющие оператору (диспетчеру), а также в автоматическом режиме контролировать технологический процесс, управлять его ходом и документировать его параметры, осуществлять обмен информацией с внешними элементами системы управления и другими контроллерами, участвующими в управлении.

Физическая часть SCADA-системы – это оконечные устройства управления: датчики, преобразователи, исполнительные устройства, непосредственно воздействующие на объекты управления, которые, например, составляют технологическую цепочку, а также промежуточные управляющие устройства (встроенные в управление подсистемами компьютеры, программируемые логические контроллеры) и линии связи с этими устройствами.

Проектирование систем автоматики (АСУТП и АСУ) предполагает и создание проекта построения физической части системы SCADA, и ее виртуальной части. Проектирование физической части осуществляется в соответствии с

требованиями технологии, и состоит в выборе и определении предполагаемых мест размещения физических элементов автоматики, моделировании и оптимизации параметров.

Проектирование виртуальной части SCADA системы проводится в специальной программе, редакторе проекта SCADA и состоит в выборе и задании свойств компонентов проекта SCADA -системы (мнемосхем, программ обработки информации и выработки сигналов и команд, протоколов и драйверов устройств и модулей связи, форм и порядка документирования хода процесса управления и др.) и соединении их посредством некоторых переменных (аргументов) друг с другом, а также с внешними устройствами [11].

Такого рода проекты создаются в общей среде как для АРМ, так и для подчиненных им встроенных в управление компьютеров и контроллеров.

9.3 Задание к лабораторной работе

Для своего варианта (таблица 1) разработать базовый интерфейс посредством создания статического и динамического варианта мехатронной системы.

9.4 Методические указания по выполнению работы

Проектирование лучше организовать по последовательному принципу: создавать и отлаживать одну ветвь или один контур управления, к которым можно поочередно добавлять другие ветви и контура, последовательно отлаживая их работу.

TRACE MODE сравнительно медленно работающая система с опросом внешних устройств, на который могут потребоваться доли секунд, секунды и десятки секунд.

При выполнении заданий важно не просто автоматически выполнять предлагаемые действия, благо они подробно проиллюстрированы, а запомнить и, главное, понять и почувствовать принципы и логику построения проекта.

Отметим, что равноценные проекты, приводящие к правильно работающим виртуальным системам контроля и управления (SCADA) могут быть построены несколько отличающимися способами, например, привязкой переменных.

9.5 Пример

Приведем главный экран управления манипулятором, разработанный в SCADA TRACE MODE (рисунок 8).

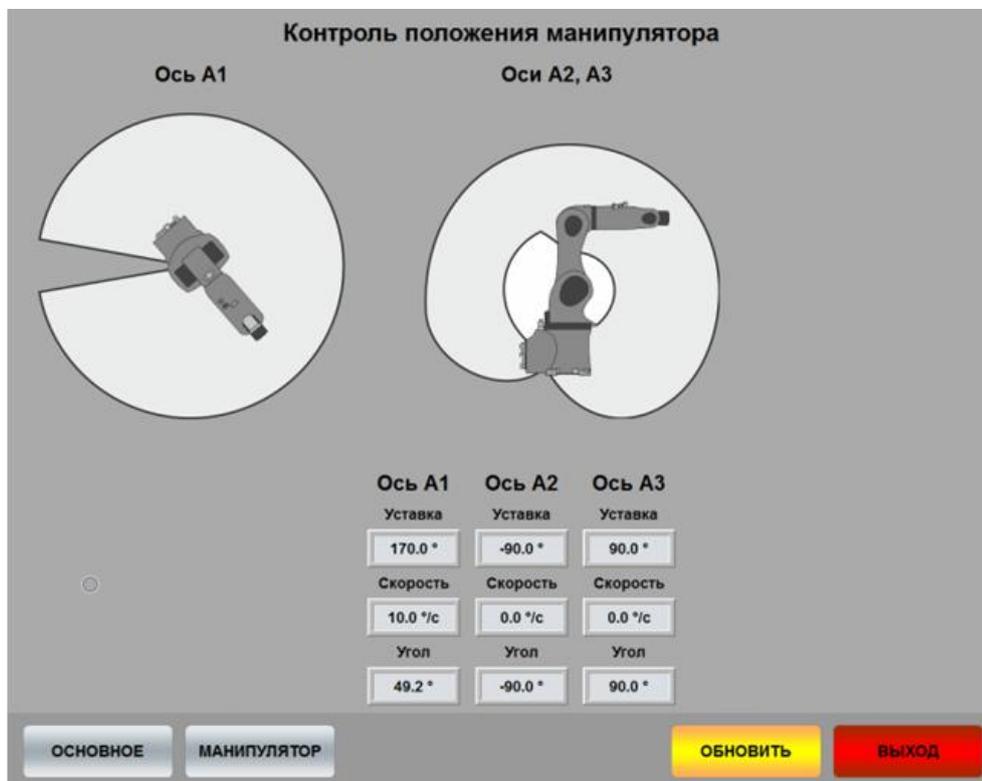


Рисунок 8 – Экран управления манипулятором в SCADA TRACE MODE

Список использованных источников

1 Воронин, А.А. Моделирование мехатронных систем: учебное пособие / А.А. Воронин. – Томск: Изд-во Томского политехнического университета, 2008. – 127 с.

2 Боев, В.Д. Имитационное моделирование систем : учеб. пособие для прикладного бакалавриата / В.Д. Боев. – М. : Издательство Юрайт, 2017. – 253 с.

3 Сергушичева, А.П. Устройства автоматики и вычислительная техника в производственных системах: учебное пособие в 2-х частях/ А.П. Сергушичева. – Вологда: ВоГТУ, 2007. – 110 с.

4 Жавнер, В.Л. Мехатронные системы: учеб. пособие / В. Л. Жавнер, А. Б. Смирнов. – СПб.: Изд-во Политехн. ун-та, 2011. – 131 с.

5 Едемский, С.Н. Электрические схемы систем автоматики. Часть 1.: Учебное пособие / С.Н. Едемский. – Северодвинск: РИО Севмашвтуза, 2007. – 121 с.

6 Волжанова, О.А. Схемы электрические принципиальные: учеб.- метод. пособие / О.А. Волжанова. – Ижевск: Изд-во «Удмуртский университет», 2012. – 43 с.

7 ГОСТ 2.702-2011 Единая система конструкторской документации (ЕСКД). Правила выполнения электрических схем : утвержден Межгосударственным советом по стандартизации, метрологии и сертификации (протокол от 12 мая 2011 г. N 39). – URL: <http://docs.cntd.ru/document/1200086241> (дата обращения: 18.11.2019). – Текст: электронный.

8 ГОСТ 2.704-2011 Единая система конструкторской документации (ЕСКД). Правила выполнения гидравлических и пневматических схем : утвержден Межгосударственным советом по стандартизации, метрологии и сертификации (протокол от 12 мая 2011 г. N 39). –

<http://docs.cntd.ru/document/1200086243> (дата обращения: 18.11.2019). – Текст: электронный.

9 Белов, А.В. Микроконтроллеры AVR. От азов программирования до создания практических устройств / А.В. Белов. – СПб.: Наука и техника, 2016. – 544 с.

10 Шилдт. Г. С++ для начинающих / Г. Шилдт. – М.: Знание, 2013. – 640 с.

11 Интегрированные системы проектирования и управления: SCADA-системы : учебное пособие / И.А. Елизаров, А.А. Третьяков, А.Н. Пчелинцев и др. – Тамбов : Изд-во ФГБОУ ВПО «ТГТУ», 2015. – 160 с.