

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Технология машиностроения»

# ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

*Методические рекомендации к лабораторным работам  
для студентов направления подготовки  
15.03.06 «Мехатроника и робототехника»  
дневной формы обучения*



Могилев 2019

УДК 621.8  
ББК 34.5  
И 38

Рекомендовано к изданию  
учебно-методическим отделом  
Белорусско-Российского университета

Одобрено кафедрой «Технология машиностроения» «19» ноября 2018 г.,  
протокол № 6

Составители: д-р техн. наук, проф. Г. Ф. Шатуров;  
канд. техн. наук, доц. И. Д. Камчицкая

Рецензент канд. техн. наук, доц. В. В. Кутузов

Представлены методические рекомендации к лабораторным работам по  
дисциплине «Информационные технологии».

Учебно-методическое издание

## ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Ответственный за выпуск	В. М. Шеменков
Технический редактор	А. Т. Червинская
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.  
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 46 экз. Заказ №

Издатель и полиграфическое исполнение:  
Межгосударственное образовательное учреждение высшего образования  
«Белорусско-Российский университет».  
Свидетельство о государственной регистрации издателя,  
изготовителя, распространителя печатных изданий  
№ 1/156 от 24.01.2014.  
Пр. Мира, 43, 212000, Могилев.

© Белорусско-Российский  
университет, 2019



## Содержание

Инструкция по охране труда при проведении лабораторных работ.....	4
1 Лабораторная работа № 1. Изучение лабораторной установки на основе манипулятора Kawasaki FS003N, видеокамеры 2D с программным обеспечением обработки изображения RFA-Vision v4.1.....	6
2 Лабораторная работа № 2. Определение параметров плоских объектов при помощи СТЗ манипулятора Kawasaki FS003N.....	17
3 Лабораторная работа № 3. Исследование индуктивных датчиков перемещений подвижных органов робота .....	27
4 Лабораторная работа № 4. Изучение и исследование датчиков скорости тахогенераторов .....	34
5 Лабораторная работа № 5. Изучение и исследование фотоэлектрических датчиков положения .....	37
Список литературы.....	41



# Инструкция по охране труда при проведении лабораторных работ

## *Общие требования безопасности*

1 Допуск студентов к лабораторным занятиям производится только после инструктажа по технике безопасности, о чем делается соответствующая запись в специальном журнале. Получивший инструктаж подтверждает его прохождение подписью.

2 При проведении лабораторных работ и лабораторного практикума необходимо соблюдать правила пожарной безопасности, знать места расположения первичных средств пожаротушения.

3 Студентам следует быть внимательными и дисциплинированными, точно выполнять указания преподавателя.

4 Пребывание студентов в лаборатории разрешается только в присутствии преподавателя.

## *Требования безопасности перед началом работы*

1 Внимательно изучить содержание и порядок проведения лабораторной работы или лабораторного практикума, а также безопасные приемы его выполнения.

2 В случае неисправности оборудования немедленно сообщить об этом преподавателю и до ее устранения к работе не приступать (работать на неисправном и на не имеющем необходимых защитных ограждений оборудовании запрещается).

3 Перед каждым включением оборудования предварительно убедиться, что его пуск безопасен.

## *Требования безопасности во время работы*

1 Точно выполнять все указания преподавателя при проведении лабораторной работы, без его разрешения не производить самостоятельно никаких работ.

2 Не прикасаться к находящимся под напряжением элементам электрической цепи, к корпусам стационарного электрооборудования, не производить переключений в цепях до отключения источника тока.

3 Постоянно наблюдать за работой робота в процессе работы.

4 Запрещается во время работы робота снимать ограждения и предохранительные устройства, а также держать их открытыми. Во время работы не касаться руками вращающихся частей робота, не вводить руки в зону движения манипулятора.

5 Все подготовительные работы на оборудовании проводить в их обесточенном состоянии или в режиме «Наладка».

6 Работать только с теми деталями, масса и габариты которых соответствуют паспортным данным робота. Детали должны устанавливаться правиль-



но и надежно закрепляться.

7 При обнаружении неисправности в работе электрических устройств, находящихся под напряжением, повышенном их нагревании, появлении искрения, запаха горелой изоляции и т. д. немедленно отключить источник электропитания и сообщить об этом преподавателю.

### ***Требования безопасности по окончании работы***

- 1 Полностью выключить оборудование.
- 2 Привести в порядок рабочее место.
- 3 Предупредить преподавателя обо всех, даже малейших и незначительных, неисправностях оборудования.



# 1 Лабораторная работа № 1. Изучение лабораторной установки на основе манипулятора Kawasaki FS003N, видеокамеры 2D с программным обеспечением обработки изображения RFA-Vision v4.1

**Цель работы:** изучение структуры и основных компонентов учебно-исследовательского комплекса на базе промышленного робота Kawasaki; приобретение навыков в управлении и программировании промышленного робота.

## 1.1 Схема учебно-исследовательского комплекса на базе промышленного робота Kawasaki

В состав комплекса (рисунок 1) входят: манипулятор Kawasaki FS003N; контроллер Kawasaki FD70; пульт оператора; вакуумный схват (установлен на манипулятор); монтажная панель; программа компьютерной симуляции Kawasaki PC ROSET.

Дополнительное оборудование – система технического зрения: видеокамера 2D; программное обеспечение обработки изображения RFA-Vision v4.1; ноутбук.

## 1.2 Описание контроллера

### *Панель контроллера.*

Внешняя передняя панель контроллера или система управления (СУ) робота осуществляет следующие функции (рисунки 2 и 3, таблица 2).

### *Включение силового питания контроллера.*

Нужно переместить вправо переключатель силового питания (номер позиции 1 в таблице 1), расположенный в нижней правой части панели контроллера.



Рисунок 1 – Общий вид учебно-исследовательского комплекса

### *Включение и выключение силового питания привода робота.*

Для включения силового питания двигателя робота нужно нажать клавиши A + Motor ON на пульте ручного управления. Затем нажать клавиши A + RUN для разрешения движения. При этом окна Motor и RUN будут включены на экране пульта управления. Выключение силового питания привода осуществляется в обратном порядке: A + Motor ON, затем A + RUN или HOLD – останавливает движение.

### *Методы для останова робота.*

Чтобы остановить робот, различаются методы останова: для режима обучения, автоматического режима и аварийного останова.

Для режима обучения:

- отпустите переключатель TRIGGER на пульте ручного управления;
- убедитесь, что робот полностью остановился, и нажмите клавишу HOLD на пульте ручного управления. В правом верхнем углу экрана пульта управления появится окно HOLD вместо окна RUN.

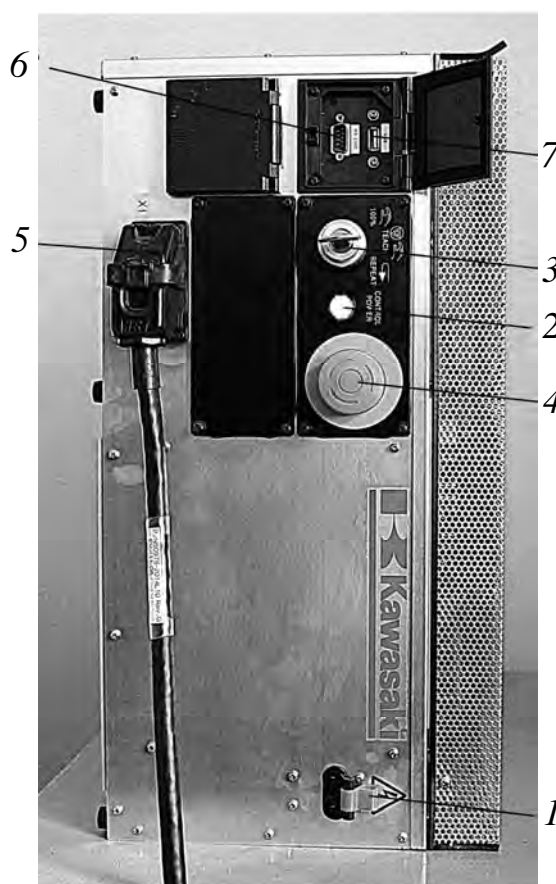


Рисунок 2 – Панель контроллера (СУ)

Для автоматического режима:

- чтобы остановить движение робота при выполнении программы в автоматическом режиме, нужно нажать клавишу HOLD на пульте ручного управления;
- чтобы закончить выполнение шага по программе и затем остановить движение робота, нужно выбрать функцию STEP ONCE нажатием в окне экрана пульта управления;



– чтобы закончить выполнение программы и затем остановить движение робота, нужно выбрать функцию REPEAT ONCE нажатием в окне экрана пульта управления.



Рисунок 3 – Панель контроллера (верхняя часть)

Таблица 1 – Переключатели и лампочки на передней панели контроллера. Внешние входы/выходы на передней панели контроллера

Номер позиции на рисунке 2	Переключатель и лампочка	Назначение
1	Переключатель	Включает и выключает общее силовое питание
2	Индикатор источника силового питания	Загорается при включении общего силового питания
3	ТЕАСН/РЕРЕАТ переключатель	Переключает режимы «обучение» или «автоматический режим». Обеспечивает выбор режимов управления (режим обучения, когда переключатель находится в горизонтальном положении ТЕАСН; режим выполнения программ, когда переключатель повернут по часовой стрелке и находится в положении РЕРЕАТ)
4	Кнопка аварийного останова	Обрывает силовое питание привода и останавливает робот, когда эта кнопка нажата при непредвиденной ситуации. Силовое питание не отключается
5	Подключение первичного питания	Подключение кабеля первичного питания в контроллере
6	СОМ-порт	Подключение вспомогательных устройств и технологического оборудования
7	USB-порт	Подключение вспомогательных устройств

При аварийном останове:

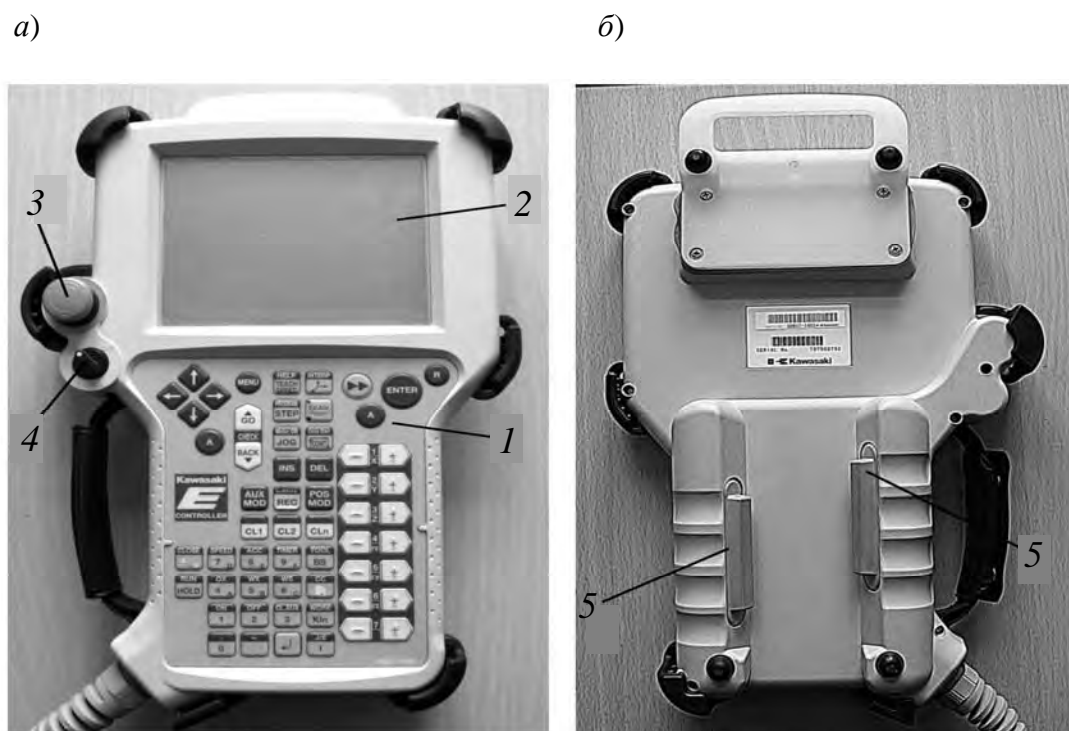
– когда робот работает неправильно, существует возможная опасность столкновения робота с предметами, получения травмы. Нужно немедленно нажать любую кнопку аварийного останова на пульте ручного управления или на операционной панели контроллера, чтобы отключить силовое питание привода.



### 1.3 Пульт ручного управления робота

*Описание пульта ручного управления.*

На рисунке 4 показан вид и дано краткое описание пульта ручного управления. Пульт ручного управления обеспечен аппаратными клавишами и переключателями, которые являются необходимыми для ручного управления роботом и редактирования данных, и экраном для того, чтобы отображать различные виды данных.



*а* – со стороны дисплея; *б* – задняя сторона пульта со стороны триггера; *1* – панель с клавишами; *2* – экран; *3* – кнопка аварийного останова; *4* – переключатель TEACH LOCK; *5* – переключатель TRIGGER

Рисунок 4 – Вид пульта ручного управления

Рисунок 5 показывает расположение аппаратных клавиш.

*Аппаратные клавиши на пульте ручного управления.*

Аппаратные клавиши на пульте ручного управления представлены в таблице 2.

*Экран пульта ручного управления.*

Пульт ручного управления обеспечен операционным дисплеем. Он состоит из трех областей: А, В, С, как показано на рисунке 6.

**Область экрана А.**

Эта область состоит из пяти экранов. Их функции приведены далее.



Рисунок 5 – Расположение аппаратных клавиш на пульте управления

Таблица 2 – Назначение основных аппаратных клавиш на пульте ручного управления и переключателей

Тип клавиши, переключателя	Назначение клавиши, переключателя
1	2
MENU	Отображает пошаговое меню выбранной области дисплея. Нажимая А + MENU, активизируем пошаговое меню выбранной области дисплея
Стрелки вверх-вниз, влево-вправо	Передвигает курсор в вертикальном и горизонтальном направлениях между шагами, пунктами, строками. Одновременное нажатие «А + стрелка вверх или вниз» переводит курсор на предыдущий шаг или последующий шаг в режиме обучения или редактирования
ENTER	Подтверждает выбор функции или пункта. Завершает ввод данных в активизированную строку выбранной области
R	Отмена последней операции. Закрывает пошаговое меню. Возврат в первоначальное состояние
A	«А» используется при выборе кнопок синего цвета (одновременное нажатие А +)
GO	Осуществляет пошаговое выполнение программных инструкций, переходя с выбранного шага к последующему
BACK	Осуществляет пошаговое выполнение программных инструкций, переходя с выбранного шага к предыдущему в проверочном режиме
TEACH SPEED	Изменяет скорость движения манипулятора в режиме обучения
Оси координат	Задаёт систему координат: базовую (декартовую); относительную (движение по осям, связанным со схватом); угловую (движение отдельных суставов)
PROGRAM STEP	Нажимая А + PROGRAM, активизируем меню выбора программ. Нажимая STEP, активизируем меню пошаговой области
Ext. Axis (Robot)	Выбирает дополнительную ось JT7 или еще один робот, связанный с данным роботом

Окончание таблицы 2

1	2
JOG	Медленно перемещает робот в режиме обучения или проверочном (контрольном режиме). Эффективно только в течение нажатия кнопки
CONT	Включает непрерывное или пошаговое выполнение программы
INS	Включает режим вставки программных шагов
DEL	Включает режим удаления программных шагов
AUX MOD	Включает режим редактирования дополнительной информации
POS MOD	Включает режим редактирования позиций
REC	Добавляет новый шаг следом за текущим шагом Нажимая A + REC, записываем новый шаг в режиме обучения
CL1	Переключает зажим 1 в положение ON/OFF
CL2	Переключает зажим 2 в положение ON/OFF
CLn	Переключает n-й зажим в положение ON/OFF
Клавиши «+» и «-»	Двигает каждый сустав от JT1 до JT6
ON	Нажимая A + ON, устанавливаем выбранный зажим в положение ON
OFF	Нажимая A + OFF, устанавливаем выбранный зажим в положение OFF
CLAUX	Нажимая A + CLAUX в режиме обучения, вызываем дополнительные пункты для зажимов

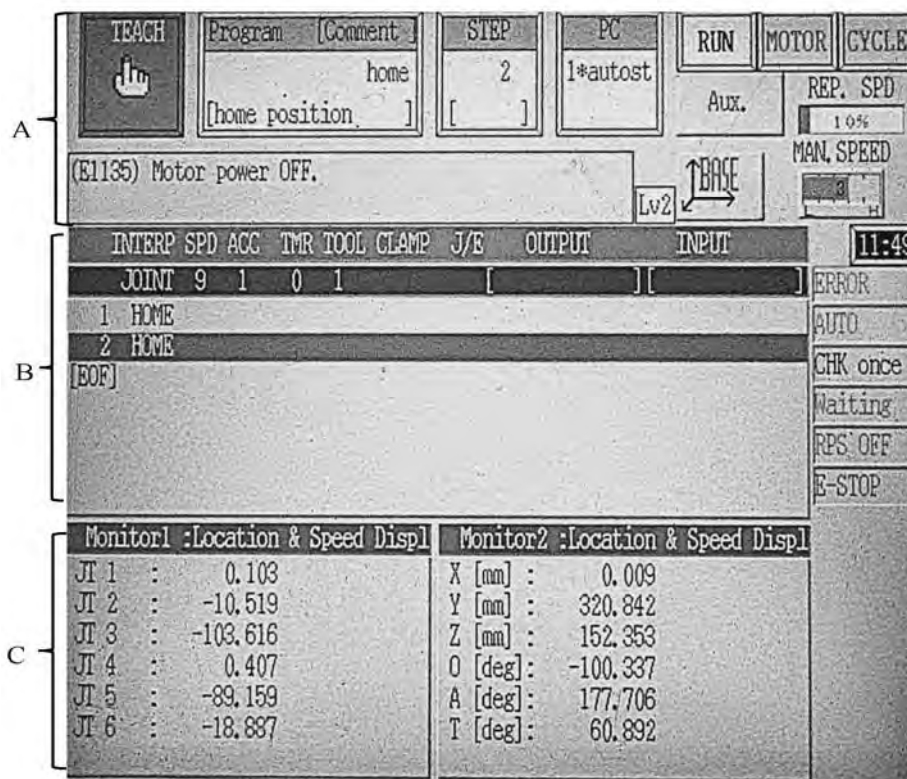


Рисунок 6 – Области дисплея



### Функции в Program Comment области.

Эта область показывает зарегистрированное имя программы и его комментарии. Пошаговое меню в данной области имеет шесть функций (рисунок 7). Чтобы вызвать пошаговое меню, нажмите A + Program на пульте ручного управления или область **Program Comment** непосредственно на экране.

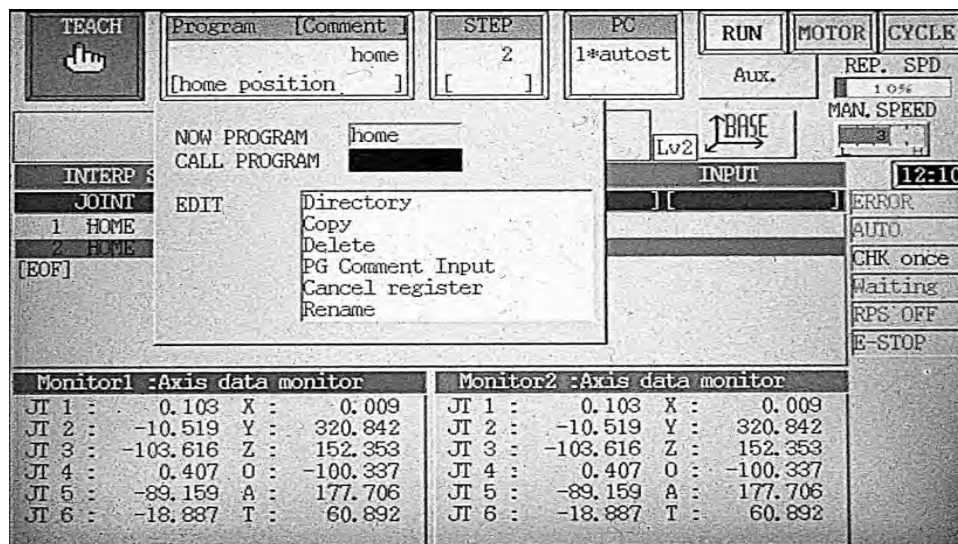


Рисунок 7 – Область Program Comment

В окно CALL PROGRAM можно ввести номер программы и выбрать программу, если ее имя состоит их чисел. Программы могут иметь имя до пяти символов.

Требуемая программа может быть выбрана из списка, если ее имя зарегистрировано.

Нужно выбрать из пошагового меню функцию **Directory** (см. рисунок 7), после этого появится экран выбора программы (рисунок 8) и из списка нужно выбрать программу.

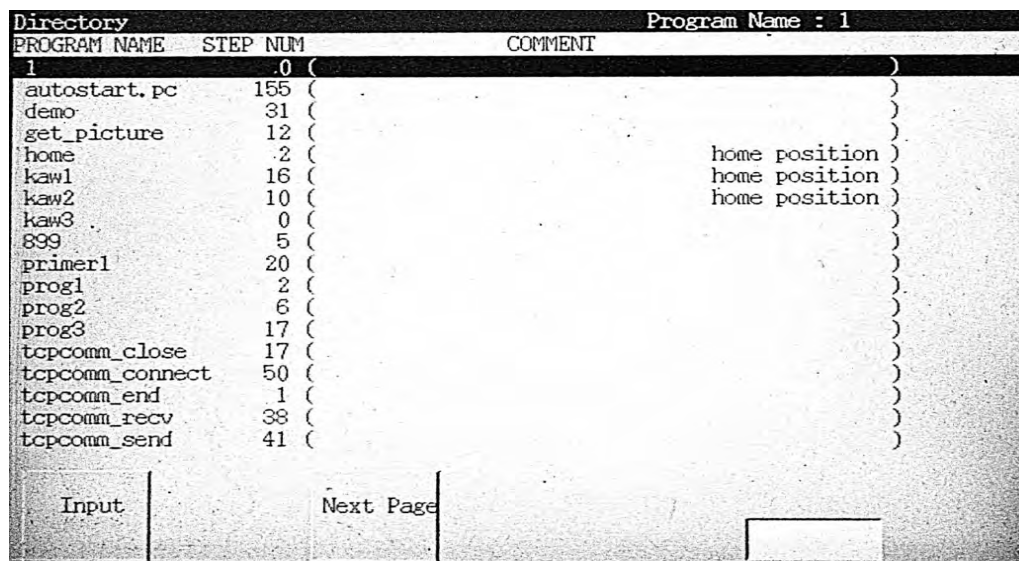


Рисунок 8 – Экран Directory выбора программы

Выбранная программа может быть удалена с помощью функции **Delete** (см. рисунок 7).

Комментарии могут быть введены в область «комментарий» выбранной программы с помощью функции **PG Comment Input**.

С помощью функции **Copy** можно скопировать содержание выбранной программы, изменив ее имя. Эту функцию можно использовать для создания новой программы.

С помощью функции **Cancel Register** программа может быть удалена из **Program Comment** области.

### **Функции в STEP области.**

Эта область показывает номер шага программы. Пошаговое меню в данной области имеет пять функций (рисунок 9). Чтобы показать пошаговое меню, нужно нажать напрямую на экране пошаговую область.

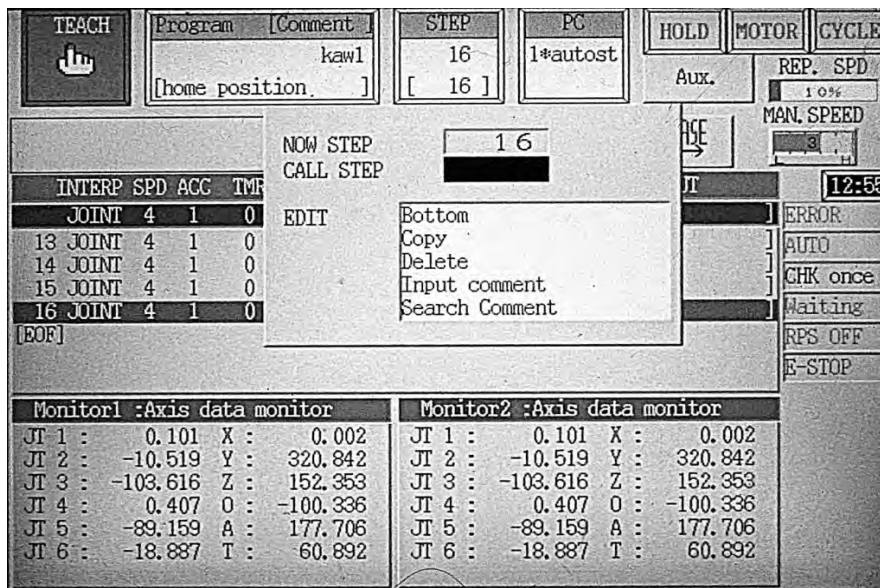


Рисунок 9 – STEP область экрана

На рисунке 9 в открытом пошаговом меню показан текущий шаг загруженной программы в поле **NOW STEP** – 16. Имя программы «kaw1» отображается в окне **Program Comment**. В поле вызова шага **CALL STEP** можно ввести номер шага программы, нажать кнопку **ENTER** на пульте и этот шаг будет выбран (на рисунке 10 выбран шаг 4).

Последний шаг в программе можно выбрать с помощью функции **Bottom** (см. рисунок 9, выбран последний шаг – 16).

### **Область системных сообщений.**

Эта область показывает сообщения об ошибках и различные виды сообщений, которые возникают в течение работы робота. На рисунке 6 в окне системных сообщений указаны код ошибки (E1135) и сообщение «Motor power OFF» (электропитание двигателя выключено).

### **Область состояния.**

Эта область расположена справа от области **PC** вверху рабочего экрана на пульте ручного управления и состоит из окон (см. рисунок 10).



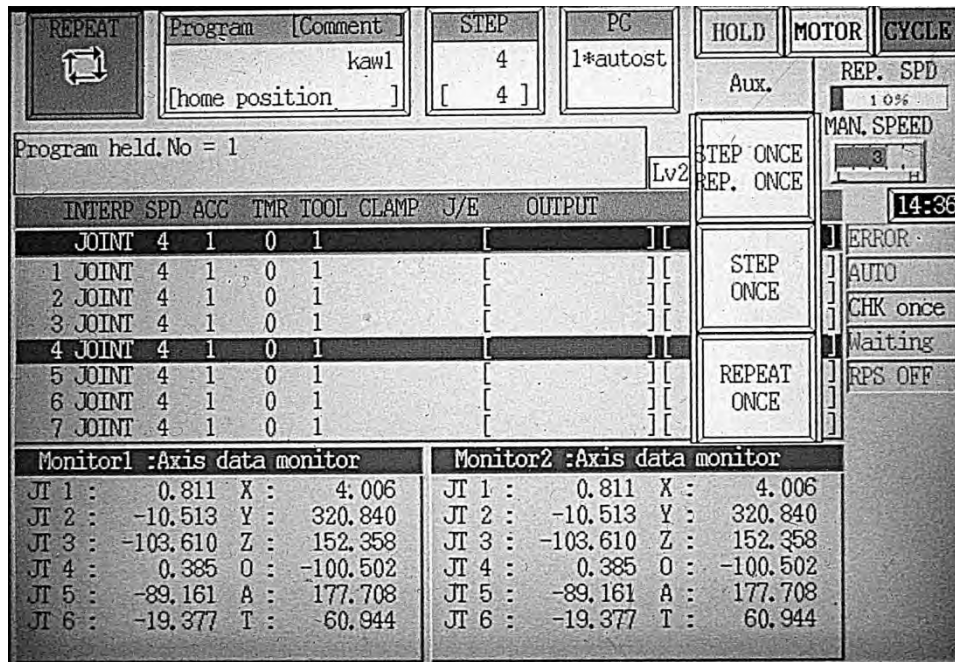


Рисунок 10 – Окна отработки программы в автоматическом режиме (пошаговая отработка)

Окно MAN. SPEED (manual speed) устанавливает скорость движения робота в режиме обучения (ручном режиме). Переключать скорость можно, нажав клавишу TEACH SPEED на пульте управления или касанием на экране в области окна.

Окно REP. SPD (Repeat speed) отображает скорость, когда робот перемещается в автоматическом режиме.

Три режима движения робота в режиме обучения:

- 1) BASE – базовая система координат;
- 2) TOOL – инструментальная система координат;
- 3) JOINT – режим суставного движения.

Нажимая клавишу с осями координат на пульте управления или соответствующее окно в области состояния, устанавливаем необходимую систему координат.

Окно MOTOR активно, когда силовое питание включено и неактивно, когда силовое питание выключено.

Окно HOLD появляется, когда происходит останов робота, окно RUN – когда робот в состоянии выполнения.

Пошаговая отработка программы в режиме обучения. Нажав клавишу CONT на пульте управления, на экране справа в окне отобразится CHK once. Выбирая шаг программы, нажав клавиши «А + стрелки вверх и вниз» на пульте управления, затем нажав и удерживая триггер и клавишу GO, выполняем программу в пределах одного шага. Затем выбираем и выполняем второй шаг и т. д.

Отработка программы в автоматическом режиме. Выбирая касанием на экране в окнах функции STEP CONT и REPEAT ONCE, выполняется загруженная программа только один раз. Выбирая функции в этих окнах STEP CONT и REPEAT CONT, выполняется загруженная программа непрерывно.

Пошаговая отработка программы в автоматическом режиме. Выбрав окно STEP ONCE на экране и нажимая клавишу GO на пульте управления, выполняем программу пошагово (см. рисунок 10).

### Область экрана В.

Левая сторона этой области содержит пошаговое меню из пяти пунктов (рисунок 11).

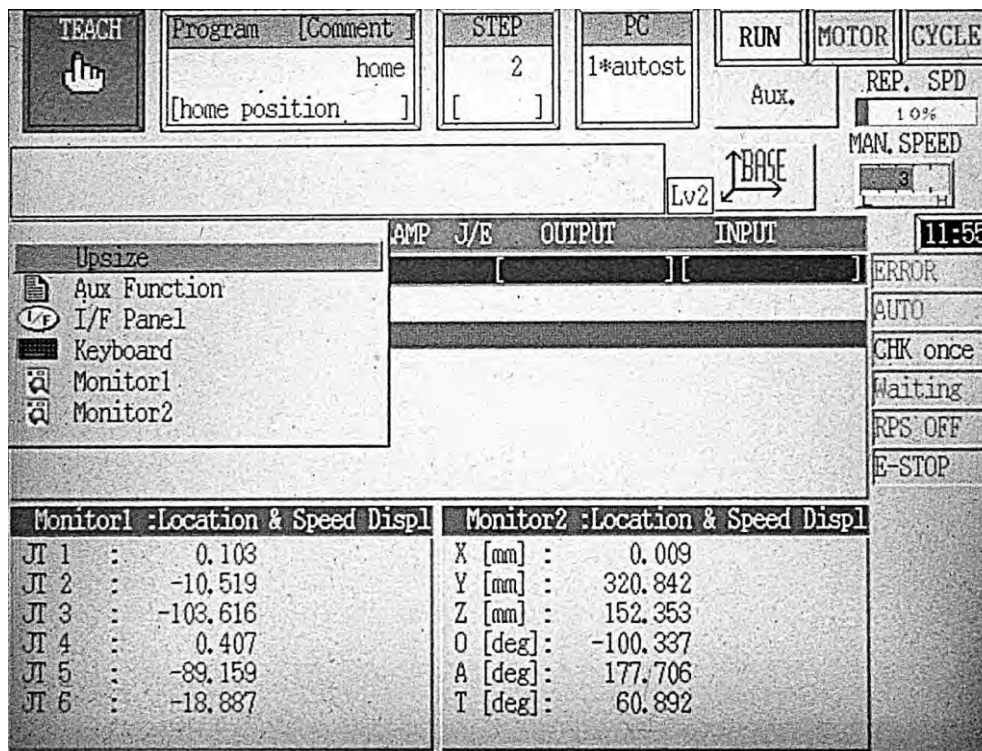


Рисунок 11 – Область экрана В

Чтобы отобразить пошаговое меню, нужно нажать клавиши A + MENU на пульте управления или окно области В непосредственно. Выбор Upsize увеличивает и отображает содержание областей В и С.

**Teach** – экран обучения. Используется для того, чтобы обучать или редактировать программу робота в блочном программировании (на рисунке 11 экран открыт).

**Aux Function** – экран вспомогательных функций. Используется, чтобы установить различные виды данных робота.

**I/F Panel** – экран интерфейсной панели. Этот экран обеспечивает выключатели и функции, необходимые для операций роботов и периферийного оборудования.

**Keyboard** – экран клавиатуры. Используется, чтобы ввести символы имени программы, комментариев и т. д.

**Monitor1** и **Monitor2** – экраны контроля открываются в области С.

### Область экрана С.

Эта область содержит два вида экранов, каждый из которых становится доступным из пошагового меню.



Экран контроля суставов. Этот экран отображает восемь видов информации, включая текущее положение робота, командные значения и т. д.

Экран контроля сигнала. Этот экран отображает предварительно установленные имена входов – выходов и контролирует состояние каждого сигнала.

#### **1.4 Задания к лабораторной работе**

1 Изучить панель контроллера Kawasaki FD70. Изучить функциональное назначение переключателей на панели, внешних входов/выходов.

2 Изучить назначение аппаратных клавиш на пульте ручного управления.

3 Изучить назначение и функционирование областей экрана А, В, С операционного дисплея пульта ручного управления.

4 Изучить методы включения/выключения контроллера, силового питания робота.

После изучения структуры и компонентов учебного роботизированного комплекса студенты под руководством преподавателя должны выполнить следующее:

- включить контроллер и силовое питание;
- на пульте управления выбрать и загрузить одну из программ из списка окна Directory, выбрать выборочно шаги программы;
- создать имя новой программы;
- установить различную скорость перемещения робота в ручном режиме;
- переключиться в каждый из трех режимов движения робота в режиме обучения;
- открыть меню в каждой области экрана А, В, С и изучить содержание функций в пошаговом меню и в открывающихся окнах;
- выключить силовое питание и контроллер.

#### **Контрольные вопросы**

- 1 Что входит в состав учебно-исследовательского комплекса?
- 2 Какие функции выполняет переключатель TEACH/REPEAT?
- 3 Как выполняется включение и выключение силового питания привода робота?
- 4 Какие применяются методы для останова робота?
- 5 Из каких элементов состоит пульт ручного управления?
- 6 Из каких областей состоит операционный дисплей? Их назначение.
- 7 Назовите три режима движения робота в режиме обучения.
- 8 Как выполняется пошаговая отработка программы в режиме обучения?
- 9 Как выполняется пошаговая отработка программы в автоматическом режиме?



## 2 Лабораторная работа № 2. Определение параметров плоских объектов при помощи СТЗ манипулятора Kawasaki FS003N

**Цель работы:** исследование программных методов распознавания и идентификации простых объектов с помощью библиотеки компьютерного зрения Intel® Open Source Computer Vision Library (OpenCV) .

### 2.1 Схема учебно-исследовательского комплекса на базе промышленного робота Kawasaki

В состав комплекса (рисунок 1) входят: манипулятор Kawasaki FS003N; контроллер Kawasaki FD70; пульт оператора; вакуумный схват (установлен на манипулятор); монтажная панель; программа компьютерной симуляции Kawasaki PC ROSET.

Дополнительное оборудование – система технического зрения: видеокамера 2D; библиотека компьютерного зрения с открытым исходным кодом OpenCV; ноутбук.

### 2.2 Основные положения обработки информации в робототехнических системах, полученной с использованием систем технического зрения

Существуют два основных подхода к предварительной обработке информации. Первый подход базируется на методах пространственной области, а второй – на методах частотной области с использованием преобразования Фурье. Вместе эти подходы охватывают большинство из существующих алгоритмов предварительной обработки информации, применяемых в системах технического зрения роботов.

Основными библиотеками, работающими с методами распознавания и идентификации, являются библиотека Integrated Performance Primitives (IPP), библиотека AviCap, библиотека компьютерного зрения с открытым кодом OpenCV.

### 2.3 Библиотека OpenCV

**Операторы для определения изображений при помощи библиотеки OpenCV.**

OpenCV (англ. Open Source Computer Vision Library, библиотека компьютерного зрения с открытым исходным кодом) – библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом. Реализована на C/C++, также разрабатывается для Python, Ruby, Matlab и других языков.

OpenCV предназначена для повышения вычислительной эффективности процедур обработки видеоизображения с особым упором на применение в задачах реального времени.



OpenCV написана на C+, хорошо оптимизирована и может использовать преимущества многоядерных процессоров. Для более полного применения возможностей библиотеки рекомендуется установить Intel Performance Primitives (IPP). Это позволит повысить производительность процедур библиотеки (где взять и как ставить смотреть здесь ).

OpenCV позволяет достаточно быстро и эффективно реализовывать сложные алгоритмы машинного зрения. Библиотека содержит более 500 функций, которые дают возможность реализовывать приложения, работающие во многих областях, в том числе:

- контроля качества выпускаемой продукции;
- обработки изображений в медицине;
- обеспечения безопасности;
- интерфейсе пользователя;
- робототехнике.

OpenCV содержит библиотеку общих функций искусственного интеллекта «Machine Learning Library» (MLL). Она служит, в основном, для распознавания фрагментов изображения и кластеризации.

Данную библиотеку применяют:

- для утверждения общего стандартного интерфейса компьютерного зрения для приложений в этой области;
- для способствования росту числа таких приложений и создания новых моделей использования PC;
- сделать платформы Intel привлекательными для разработчиков таких приложений за счёт дополнительного ускорения OpenCV с помощью Intel® Performance Libraries (сейчас включают IPP (низкоуровневые библиотеки для обработки сигналов, изображений, а также медиакодеки) и MKL (специальная версия LAPACK и FFTPack));

– OpenCV способна автоматически обнаруживать присутствие IPP и MKL и использовать их для ускорения обработки.

Основными являются следующие модули.

Ядро sxcore производит:

- базовые операции над многомерными числовыми массивами;
- матричную алгебру, математические функции, генераторы случайных чисел DFT, DST;
- запись/восстановление структур данных в/из XML/YAML;
- базовые функции 2D-графики;
- поддержку более сложных структур данных: разреженных массивов, динамически растущих последовательностей, графов.

CV – модуль обработки изображений и компьютерного зрения. Его функциями являются:

- базовые операции над изображениями (фильтрация, геометрические преобразования, преобразование цветовых пространств и т. д.);
- анализ изображений (выбор отличительных признаков, морфология, поиск контуров, гистограммы);



- структурный анализ (описание форм, плоские разбиения);
- анализ движения, слежение за объектами;
- обнаружение объектов, в частности лиц;
- калибровка камер, элементы восстановления пространственной структуры.

Highgui – модуль для ввода/вывода изображений и видео, создания пользовательского интерфейса. Необходим для выполнения следующих операций:

- захват видео с камер и из видеофайлов, чтение/запись статических изображений;
- функции для организации простого UI (сейчас все демоприложения используют HighGUI).

Svaux – экспериментальные и устаревшие функции.

Данный модуль выполняет:

- пространственное зрение: стереокалибрация, самокалибрация;
- поиск стереосоответствия, клики в графах;
- нахождение и описание черт лица;
- сравнение форм, построение скелетонов;
- скрытые цепи Маркова;
- описание текстур.

#### ***2.4 Программные методы распознавания и идентификации простых объектов с помощью библиотеки компьютерного зрения OpenCV***

Проект состоит из двух классов: CApp (класс приложения) и CMainWin (главное окно) с соответствующей первой функцией InitInstance(), конструктором и картой сообщений для класса окна [2].

В программе указывается глобальное объявление функции обработки кадра – void mycallback (IplImage \*img).

Указателями на изображение будут являться: IplImage \*image1,\*src2,\*dst,\*dst2,\*dst3,\*dst4,\*gray,\*dst5,\*dst6,\*dst7,\*dst8.

Далее выделяются переменные для сохранения памяти. В данном случае это CvMemStorage \*storage,\*storage2.

Конструктор главного окна программы необходимо описать следующим образом:

```
CMainWin::CMainWin()
{ Create (NULL, "OpenCV");
```

Далее в конструкторе указывается дескриптор главного окна программы:

```
HWND w= this->GetSafeHwnd();
```

Также в конструкторе определяется количество камер с помощью команды

```
int ncams=cvcamGetCamerasCount();
```



Далее идет проверка наличия камер, если их количество ненулевое, выполняется ряд функций, связанных с инициализацией камеры, основных окон программы и свойств изображений:

```
if (ncams) {bCreate=true;
VidFormat vidFmt={ 800,600,20.0};
cvcamSetProperty(0,CVCAM_PROP_ENABLE,CVCAMTRUE);
cvcamSetProperty(0,CVCAM_PROP_CALLBACK,mycallback);
cvcamSetProperty(0,CVCAM_PROP_WINDOW,&w);
cvcamSetProperty(0,CVCAM_PROP_SETFORMAT,&vidFmt);
cvNamedWindow(cvGetWindowName(w),CV_WINDOW_AUTOSIZE);
}
```

На рисунке 12 указано главное окно программы, которое содержит пример исходного изображения, полученного камерой СТЗ робота.

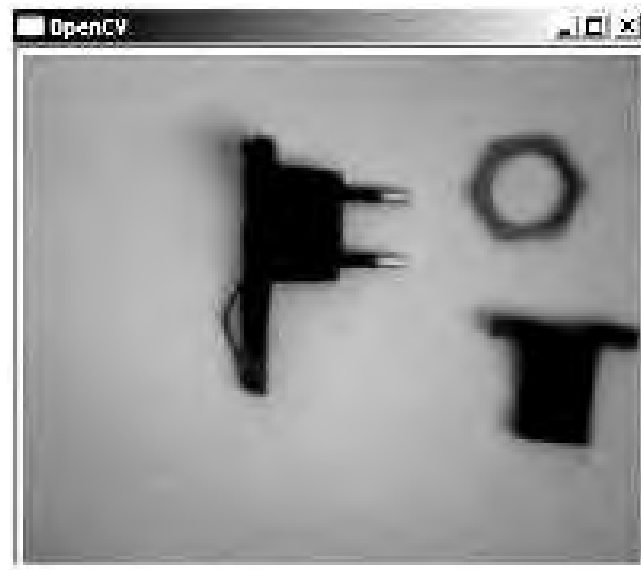


Рисунок 12 – Пример изображения в главном окне программы

Для организации вывода результатов обработки визуальной информации используется функция `cvNamedWindow("Canny", 1)`, где 1 – идентификатор окна. Результат каждого преобразования исходного изображения выводится при помощи команды `cvShowImage()`.

На рисунке 13 приведен пример применения функции `Canny`, обеспечивающей выделение границ изображений.

С помощью функции `cvCreateTrackbar` создается полоса прокрутки `cvCreateTrackbar("CannyTrack", "Canny", &cannyt, 200, NULL)` и задается размер окна для функции `cvResizeWindow("Canny", 320, 200)`.

Аналогичным образом создается окно для вывода функции определения контуров. На рисунке 14 приведено выполнение функции.

Затем производится инициализация камеры либо выводится сообщение об ошибке:

```
if(!cvcamInit()) MessageBox("Error");
```

в противном случае производится запуск камеры:

```
elsecvcamStart();}
```

либо выдается сообщение о том, что она не найдена.

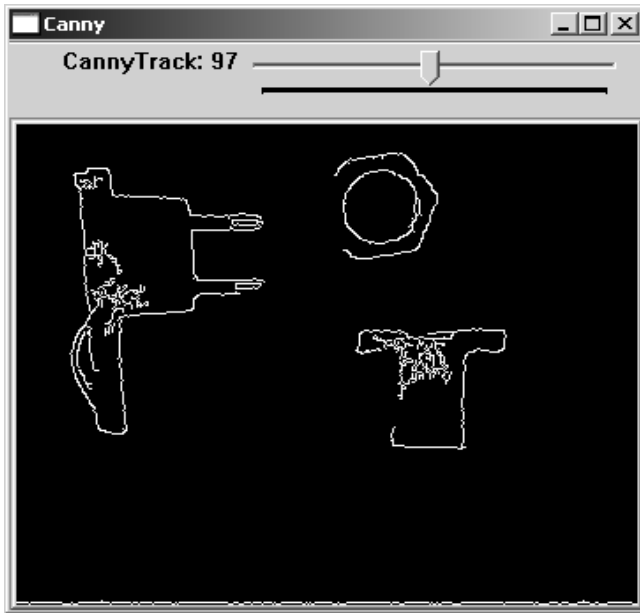


Рисунок 13 – Окно программы с функцией Canny

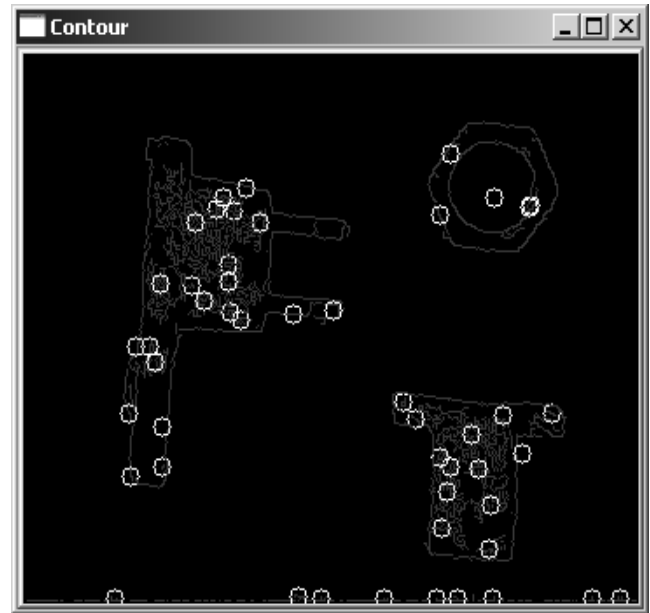


Рисунок 14 – Определение контуров изображений

Закрытие главного окна программы должно обеспечивать закрытие всех программных окон (в противном случае они могут остаться в памяти). Для этого необходимо использовать функцию `void CMainWin::OnClose()`, в которой производится остановка камеры (`cvcamStop()`), закрытие всех окон (`cvDestroyAllWindows()`), выход из режима камеры (`cvcamExit()`) и закрытие главного окна:

```
if(!bCreate) cvReleaseImage(&dst);
DestroyWindow(); }.
```

## 2.5 Реализация функции обработки изображений

Обработка изображений в OpenCV производится кадр за кадром, а функция обратного вызова (`mycallback`) в разработанном программном обеспечении выглядит так: `void mycallback(IplImage *src)`.

Изображение, полученное с камеры в тексте программы, обозначено переменной `src` и выводится в основное окно программы (рисунок 15).

В приложении используются различные типы моделей памяти, соответствующие различным типам изображений, например:

```
src2=cvCreateImage(cvSize(src->width,src->height),IPL_DEPTH_8U,3);
dst2=cvCreateImage(cvSize(src->width,src->height),IPL_DEPTH_32F,3);
```



```
gray=cvCreateImage(cvSize(src->width,src->height),IPL_DEPTH_8U,1);
```

Для поворота изображения использовалась функция `cvFlip(src, src2)`.



Рисунок 15 – Основное окно программы (уменьшенный вид)

Для нормальной работы функции Canny необходимо преобразовать исходное изображение в черно-белое (используется функция `cvCvtColor(src2,gray,CV_RGB2GRAY)`), после чего выполнить само преобразование функцией `cvCanny(gray,dst3,25,100+cannyt,3)`.

Далее производится вывод этого изображения в окно `dst3` при помощи функции `cvShowImage("Canny",dst3)`. На рисунке 16 представлен пример выполнения функции Canny.

Одной из простейших функций обработки изображений является пороговая функция. В библиотеке OpenCV она представлена в следующем виде:

```
cvThreshold( tgray, gray, 100, 255, CV_THRESH_BINARY ).
```

Здесь `tgray` – входной массив (8- или 32-битный); `gray` – выходной массив такого же типа, как и `tgray`; 100 и 255 – минимальное и максимальное значения, используемые пороговой функцией; `CV_THRESH_BINARY` – показывает тип пороговой функции. Пример применения пороговой функции представлен на рисунке 17.

Более улучшенной разновидностью пороговой функции является функция

```
cvAdaptiveThreshold  
(dst3,dst4,5+thresh2,CV_ADAPTIVE_THRESH_MEAN_C,CV_THRESH_BINARY,3,5).
```

Пример использования функции приведен на рисунке 18.





Рисунок 16 – Пример преобразования Canny

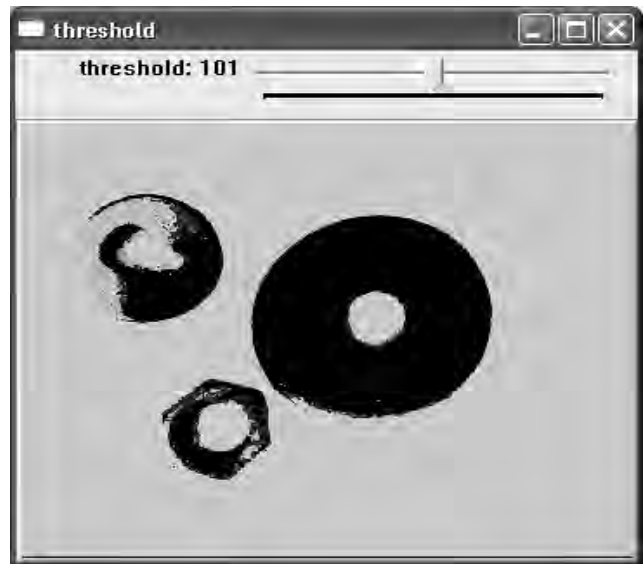


Рисунок 17 – Применение пороговой функции



Рисунок 18 – Применение адаптивной пороговой функции

## 2.6 Реализация функций распознавания и идентификации

Результатом выполнения функции Canny является изображение, содержащее края объектов в исходном кадре и выступающее основой для последующих преобразований, в частности для выделения контуров.

Рассмотрим функцию `cvFindContours`, которая находит контуры объектов, находящихся в поле зрения камеры:

```
cvFindContours(gray, storage, &contours, sizeof(CvContour), CV_RETR_LIST, CV_CHAIN_APPROX_SIMPLE, cvPoint(0,0) ).
```

Здесь `gray` является исходным изображением в 8-битном формате, в области памяти `storage` хранятся полученные контуры; в массиве `contours` содержится информация о первом найденном контуре; `CvContour` показывает размер структуры

данных контура; параметр `CV_RETR_LIST` означает, что все выделенные контуры помещаются в список; параметр `CV_CHAIN_APPROX_SIMPLE` означает, что функция выделяет центры контуров; `cvPoint(0,0)` показывает, что контуры извлекаются из изображений и затем должны быть проанализированы в контексте целого изображения.

Для нахождения нескольких контуров используется следующий цикл:

```
while(contours)
{
    result = cvApproxPoly( contours, sizeof( CvContour), storage,
CV_POLY_APPROX_DP, cvContourPerimeter( contours)*0.02, 0 );
    if( result->total == 4 && fabs( cvContourArea( result, CV_WHOLE_SEQ)) >
1000 && fabs( cvContourArea( result, CV_WHOLE_SEQ)) <( img->height * img-
>width/2 ) && cvCheckContourConvexity( result) )
    {
        s = 0; for( int i = 0; i < 5; i++ )
        {
            if( i >= 2 ) { t = fabs( angle( ( CvPoint*)cvGetSeqElem( result, i ),
(CvPoint*)cvGetSeqElem( result, i-2 ),
( CvPoint*)cvGetSeqElem( result, i-1 ))); s = s > t ? s : t; } }
            if( s < 0.5 ) for( int i = 0; i < 4; i++ ) cvSeqPush( squares, ( CvPoint*)cvGetSeqElem( result, i )); }
        contours = contours->h_next; }.
```

На рисунке 19 представлен результат применения функции выделения контуров.

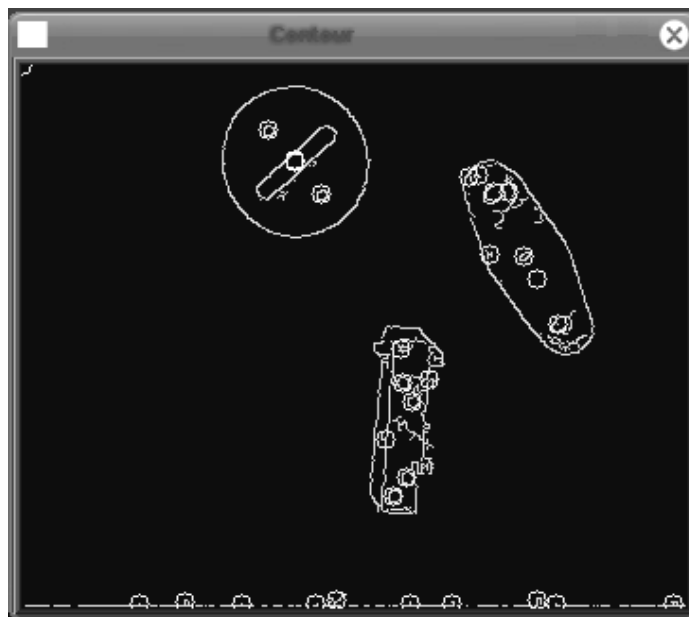


Рисунок 19 – Пример выделения контуров изображений

Функция `cvHoughCircles` находит окружности на сером изображении, используя преобразование Хью:

```
cvHoughCircles( gray, cstorage, CV_HOUGH_GRADIENT, 1, gray-
>height/16, 8, 10, 4, 50 ).
```

В данной функции, применяемой в программном обеспечении, параметр `gray` является 8-битым изображением; `cstorage` – область памяти, в которой хранятся окружности, обнаруженные функцией; `CV_HOUGH_GRADIENT` – метод реализации функции; `1` – стек, который используется для обнаружения центров окружностей с тем же разрешением, что и исходное изображение; `gray->height/16` – вычисляется минимальное расстояние между центрами обнаруженных окружностей; следующие 2 параметра отвечают за накопление порогов в обнаруженных окружностях; последние 2 параметра отвечают за минимальный и максимальный радиусы находимых окружностей.

Для отображения окружностей используется цикл, который выглядит так:

```
for( int i = 0; i < circles-> total; i++ )
{ float* p = ( float*)cvGetSeqElem( circles, i );
  cvCircle( out, cvPoint( cvRound( p[0]), cvRound( p[1])),
    2, CV_RGB( 200, 0, 0), -1, 8, 0 );
  cvCircle( out, cvPoint( cvRound( p[0]), cvRound( p[1])), cvRound( p[2]),
    CV_RGB( 200, 0, 0), 1, 8, 0 ); }
```

Пример выполнения функции представлен на рисунке 20.

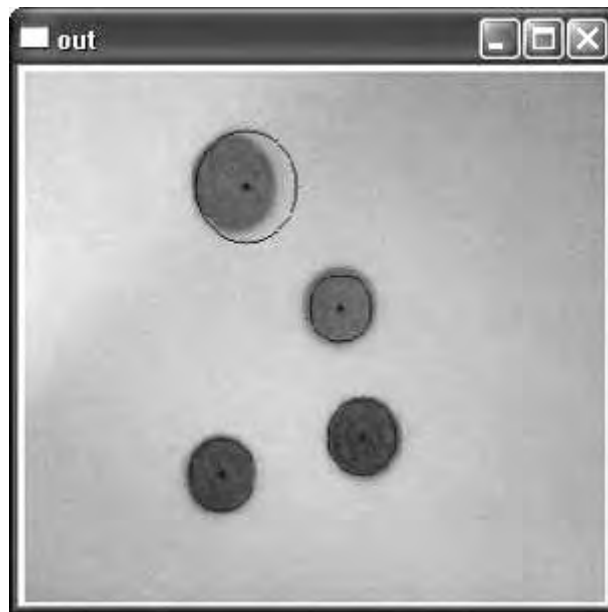


Рисунок 20 – Применение функции выделения кругов

Отображение прямоугольных изображений осуществляется командой  
`void drawSquares(IplImage *img, CvSeq* squares);`

Прямоугольные объекты определяются при помощи функции полилиний  
`cvPolyLine( img, &rect, &count, 1, 1, CV_RGB(200,0,0), 1, CV_AA, 0 );`

Далее читается последовательность 4 линий

```
CV_READ_SEQ_ELEM( pt[0], reader );
CV_READ_SEQ_ELEM( pt[1], reader );
```

```
CV_READ_SEQ_ELEM( pt[2], reader );
CV_READ_SEQ_ELEM( pt[3], reader ).
```

Угол поворота линий полилинии задан при помощи функции

```
double angle = abs(pt[1].y-pt[2].y)/sqrt((pt[1].x-pt[2].x)*
(pt[1].x-pt[2].x)+(pt[1].y-pt[2].y)*(pt[1].y-pt[2].y)+0.00001);
```

Сами линии полилинии прорисовываются с помощью следующих функций:

```
cvLine( img, cvPoint(0,img->height/2), cvPoint(img->width,img->height/2),
CV_RGB(200,200,200),1, 8, 0 );
cvLine( img, cvPoint(img->width/3,0), cvPoint(img->width/3,img->height),
CV_RGB(200,200,200),1, 8, 0 );
cvLine( img, cvPoint(img->width/3*2,0), cvPoint(img->width/3*2,img-
>height), CV_RGB(200,200,200),1, 8, 0 );cvPutText( img, st, pt[1], &font,
CV_RGB(200,0,0).
```

Вывод прямоугольных объектов производится функцией `drawSquares(out,findSquares4(gray, mainStorage))`, где `out` – изображение, содержащее прямоугольные фигуры.

Пример выполнения выделения прямоугольных объектов показан на рисунке 21.

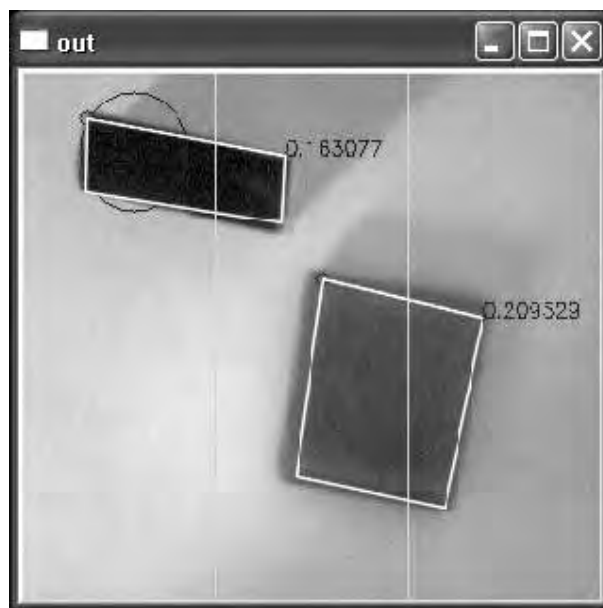


Рисунок 21 – Выполнение выделения прямоугольных объектов

Таким образом, разработанное программное обеспечение осуществляет съем и обработку информации, поступающей с камеры робота (СТЗ робота), обеспечивает выполнение границ, контуров, идентификацию простых объектов.

## Задания для выполнения лабораторной работы

Разработать программное обеспечение с использованием библиотеки OpenCV для распознавания плоских объектов в соответствии с заданием, выданным преподавателем.

### *Контрольные вопросы*

- 1 Методы идентификации объектов в роботизированных системах.
- 2 Метод сравнения с эталоном.
- 3 Методы теории графов и распознавание.
- 4 Корреляционный метод.
- 5 Распознавание через связь шаблонов.

## 3 Лабораторная работа № 3. Исследование индуктивных датчиков перемещений подвижных органов робота

**Цель работы:** изучение принципа действия, конструкции и экспериментальное определение характеристик индуктивных датчиков перемещений.

### *3.1 Лабораторное оборудование*

Нереверсивный индуктивный датчик перемещения ДПА-Ф60-40У-2110-Н (ЗАО «Сенсор»); измеритель-регулятор ТРМ-1 ОВЕН микропроцессорный одноканальный; микрометр.

### *3.2 Теоретические сведения об индуктивных датчиках перемещений*

Индуктивные датчики предназначены для преобразования механического перемещения в электрическую величину в виде напряжения или тока. По принципу действия они являются устройствами, в которых изменяется индуктивное сопротивление при изменении положения подвижной части датчика. Индуктивные датчики делятся на две группы: датчики с переменным коэффициентом самоиндукции и трансформаторные датчики, в которых изменение положения подвижной системы вызывает изменение коэффициента взаимной индукции.

*Датчики с переменной индуктивностью.*

Простейшим датчиком с переменной индуктивностью является дроссель, у которого может изменяться воздушный зазор между сердечником 1 и подвижным якорем 2 (рисунок 22).

При отсутствии насыщения магнитопровода индуктивность дросселя  $L$ , Гн, определяется формулой



$$L = \frac{\omega^2}{R_m + 2 \frac{\delta_{\text{возд}}}{S_{\text{возд}} \cdot 4 \cdot \pi \cdot 10^{-9}}} \approx \frac{\omega^2 \cdot S_{\text{возд}} \cdot 4 \cdot \pi \cdot 10^{-9}}{2 \cdot \delta_{\text{возд}}}, \quad (1)$$

где  $\omega$  – число витков обмотки;

$S_{\text{возд}}$  – площадь воздушного зазора, см<sup>2</sup>;

$R_m$  – магнитное сопротивление сердечника и якоря;

$\delta_{\text{возд}}$  – величина воздушного зазора, см.

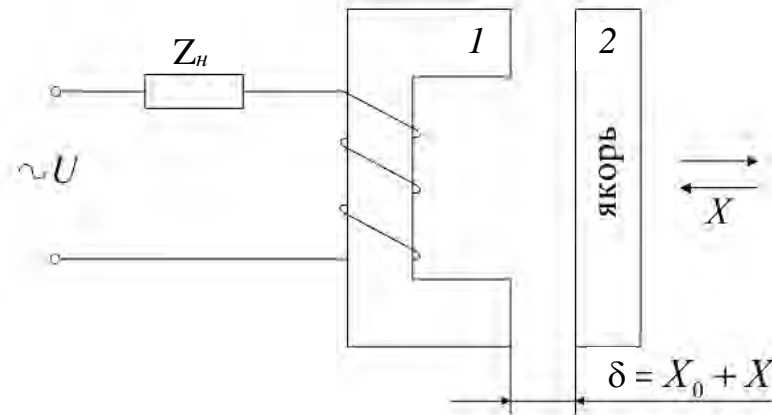


Рисунок 22 – Датчик с переменной индуктивностью

Для ненасыщенного магнитопровода магнитное сопротивление железа  $R_m$  мало по сравнению с сопротивлением воздушного зазора и им можно пренебречь. Так как воздушный зазор  $\delta_{\text{возд}} = X_0 + X$  состоит из начального зазора  $X_0$  и переменной величины  $X$ , определяемой перемещением якоря, то формула (1) преобразуется к виду

$$L = \frac{L_0}{1 + \sigma}, \quad (2)$$

где  $L_0$  – начальная индуктивность,

$$L_0 = \frac{1}{2X_0} \cdot \omega^2 \cdot S_{\text{возд}} \cdot 4 \cdot \pi \cdot 10^{-9}; \quad (3)$$

$\sigma$  – относительное перемещение якоря,

$$\sigma = \frac{X}{X_0}. \quad (4)$$

Выражение (2) показывает, что при изменении положения якоря изменяются индуктивность, индуктивное сопротивление и ток в цепи нагрузки. Ин-

дуктивный датчик конструкции, показанный на рисунке 22, прост, но имеет некоторые недостатки. При перемещении якоря нужно развивать большое усилие, необходимое для преодоления не только трения, но и силы притяжения якоря сердечником. Кроме того, датчик является одноконтурным, и вследствие этого изменение направления перемещения якоря не вызывает изменения знака выходной величины датчика.

### *Трансформаторные датчики.*

Примером трансформаторного датчика, в котором при изменении положения якоря изменяется коэффициент взаимной индукции, может служить следующий трансформатор (рисунок 23).

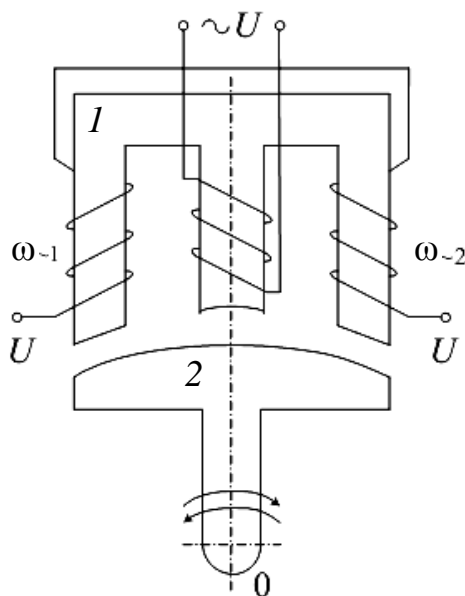


Рисунок 23 – Трансформаторный датчик

На среднем стержне Ш-образного сердечника имеется первичная обмотка, к которой подводится напряжение питания переменного тока. На двух крайних стержнях располагаются две одинаковые вторичные обмотки  $\omega_{-1}$  и  $\omega_{-2}$ , включаемые в схему последовательно и встречно. Подвижной частью датчика является якорь 2, поворачивающийся вокруг точки 0.

Если якорь 2 занимает нейтральное симметричное положение относительно крайних стержней, то магнитное сопротивление обеих выходных обмоток одинаково и, следовательно, одинаковы коэффициенты взаимной индукции между каждой из выходных обмоток и первичной обмоткой трансформатора. В этом случае в обмотках  $\omega_{-1}$ ,  $\omega_{-2}$  находятся равные по амплитуде и противоположные по фазе ЭДС, что соответствует выходному напряжению  $U = 0$ .

Поворот якоря 2 в ту или иную сторону вызывает увеличение или уменьшение магнитного сопротивления для одного крайнего стержня и противоположное изменение магнитного сопротивления для другого крайнего стержня сердечника. Это приводит к соответствующим изменениям коэффициентов взаимной индукции и к появлению результирующей ЭДС, амплитуда которой будет зависеть от



величины угла поворота якоря 2, а фаза – от направления поворота.

При использовании трансформаторного датчика в следящих системах предусматривается возможность соединения Ш-образного сердечника с выходной осью системы, а якоря – с входным устройством.

Индуктивные преобразователи используются для преобразования перемещения и других неэлектрических величин, которые могут быть преобразованы в перемещение (усилие, давление, момент и др.).

По сравнению с другими преобразователями перемещения индуктивные преобразователи отличаются значительными по мощности выходными сигналами, простотой и надежностью в работе.

Недостатком их является наличие обратного воздействия преобразователя на измеряемый объект (воздействие электромагнита на якорь) и влияние инерции якоря на частотную характеристику прибора.

### 3.2 Описание лабораторной установки

В качестве объекта испытания используется нереверсивный индуктивный датчик перемещения ДПА-Ф60-40У-2110-Н (ЗАО «Сенсор») (рисунок 24). Технические характеристики – в таблице 3.

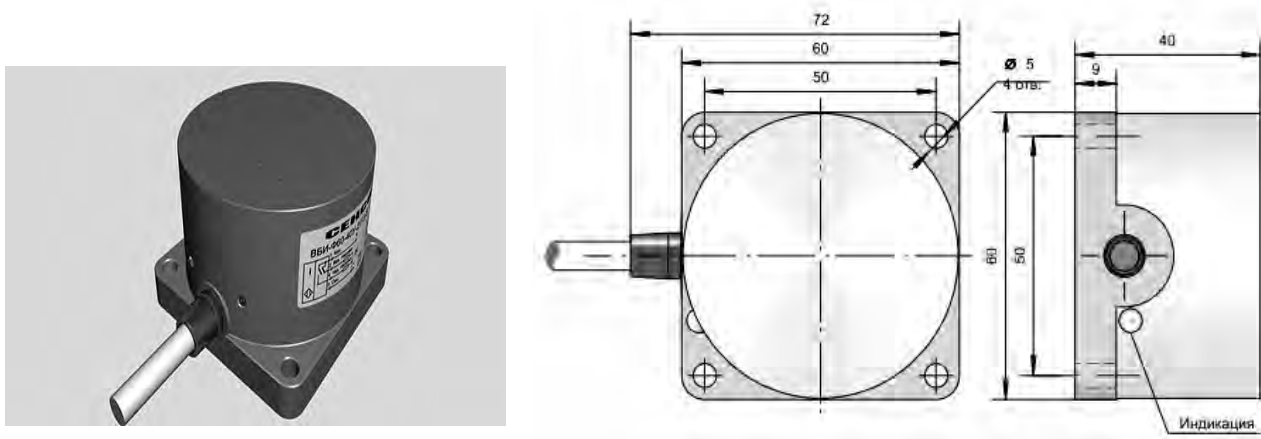


Рисунок 24 – Датчик перемещения ДПА-Ф60-40У-2110-Н

Таблица 3 – Технические характеристики индуктивного датчика

Наименование	ДПА-Ф60-40У-2110-Н
Диапазон рабочих напряжений	10...30 В DC
Рабочая зона чувствительного элемента	5...35 мм
Линейная зона чувствительного элемента	7...30 мм
Нелинейность в линейной зоне	5 %
Установка в металл	Не встраиваемый
Схема подключения	PNP общий (-)
Выход датчика	Токовый выход 1,25...20 мА
Сопротивление нагрузки токового выхода	До 500 Ом
Максимальная скорость изменения сигнала выхода	3 мА/мс

Схема лабораторной установки показана на рисунке 25.

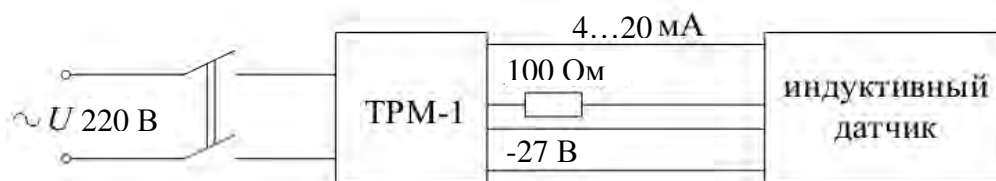


Рисунок 25 – Схема лабораторной установки

Расположение приборов на стенде представлено на рисунке 26.

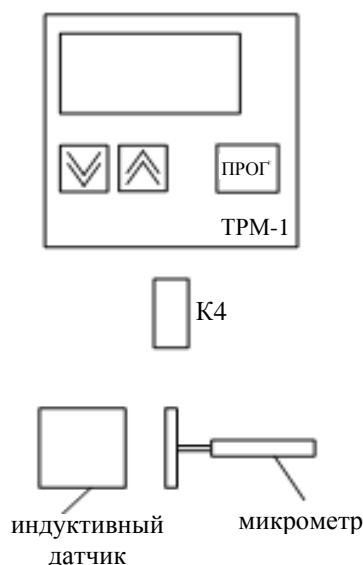


Рисунок 26 – Схема расположения приборов на стенде установки

С помощью микрометра осуществляется перемещение якоря индуктивного датчика. Питание на схему испытания подается через блок питания от сети переменного тока 220 В, 50 Гц выключателем К4. Подача напряжения сигнализируется световой сигнализацией. Отсчет величин перемещения якоря осуществляется по шкале микрометра. Выходной сигнал индуктивного датчика в виде унифицированного токового сигнала (4...20 мА) преобразуется измерителем-регулятором ТРМ-1 в текущее цифровое значение измеряемой величины (перемещение).

### 3.3 Описание измерителя-регулятора ТРМ-1

Измеритель-регулятор ТРМ-1 фирмы «ОВЕН» (рисунок 27) совместно с входным датчиком предназначен для контроля и управления различными технологическими производственными процессами.

При работе с датчиками, формирующими на выходе унифицированный сигнал тока или напряжения, предусматривается произвольное масштабирование шкалы измерения. Для этого в соответствующих функциональных пара-

метрах устанавливаются нижняя и верхняя границы диапазона отображения, а также положение десятичной точки.



Рисунок 27 – Измеритель-регулятор ТРМ-1 фирмы «ОВЕН»

Нижняя граница (параметр b1-5) определяет, какое значение будет выводиться на индикатор при минимальном уровне сигнала с датчика (при 4 мА для датчика с выходным сигналом 4...20 мА).

Верхняя граница (параметр b1-6) определяет, какое значение будет выводиться на индикатор при максимальном уровне сигнала с датчика (при 20 мА для датчика с выходным сигналом 4...20 мА).

Параметр «положение десятичной точки» b1-7 определяет количество знаков после запятой, которое после масштабирования будет выводиться на индикатор полученный результат.

Вычисленные прибором значения могут быть откорректированы пользователем с целью устранения начальной погрешности преобразования входных датчиков. Эти погрешности выявляются после проведения метрологических испытаний и устраняются путем ввода корректирующих значений.

В приборе заложены два параметра, позволяющие осуществить сдвиг и изменение наклона измерительной характеристики прибора на заданную величину (рисунок 28).

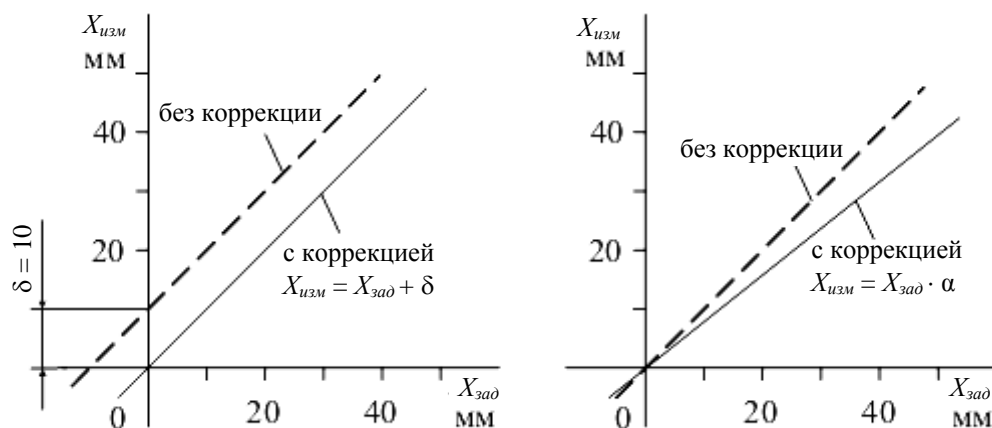


Рисунок 28 – Редактирование характеристики сдвигом и наклоном [5]

При сдвиге к вычисленному значению измеренной величины прибавляется значение, заданное параметром b1-1. Изменение наклона характеристики, скорректированной «сдвигом», осуществляется умножением значения на поправочный коэффициент, задаваемый параметром b1-2. Этот коэффициент близок к единице и находится в пределах 0,900...1,100.

### 3.4 Программирование измерителя-регулятора ТРМ-1

Программирование предназначено для задания и записи в энергонезависимую память прибора требуемых при эксплуатации рабочих параметров измерения. При входе в режим ПРОГРАММИРОВАНИЕ, если в течение 20 с не производится операций с кнопками, прибор автоматически возвращается в режим РАБОТА.

Функциональные параметры прибора разложены на группы *A* и *b*. В группе *A* находятся параметры, определяющие логику работы прибора. В группе *b* – параметры, отвечающие за настройку измерительной части прибора.

Вход в режим ПРОГРАММИРОВАНИЕ осуществляется нажатием и удерживанием кнопки «ПРОГ» более 6 с. Последовательность процедуры программирования прибора для группы *b* приведена на рисунке 29.

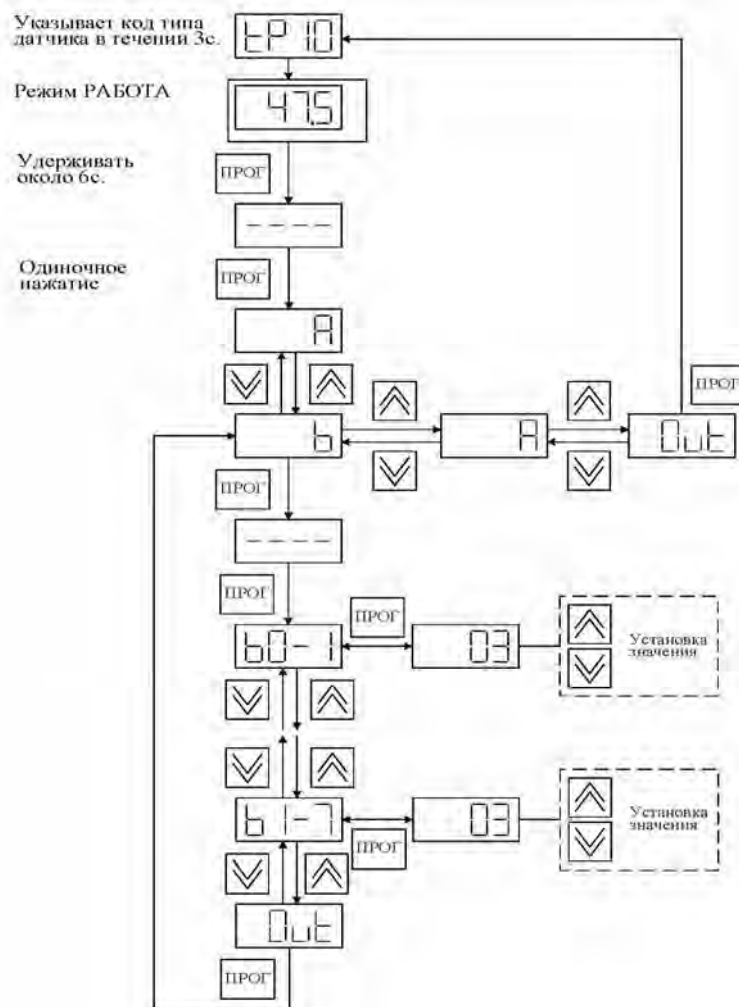


Рисунок 29 – Последовательность процедуры программирования прибора для группы *b*

## Порядок выполнения работы

- 1 Ознакомиться с данными методическими рекомендациями и лабораторным стендом.
- 2 Подготовить протокол для записи показателей опытов.
- 3 Установить автомат К0 в положение «ВКЛ». Подать питание на схемы выключателя К4.
- 4 Запрограммировать параметры b1-5 (0) и b1-6 (30) (см. рисунок 29).
- 5 Снять характеристику  $x_{вых} = f(x_{вх})$ .
- 6 Определить погрешности.
- 7 Используя сдвиг и изменение наклона характеристики, при помощи параметров b1-1 и b1-2 снизить погрешность к минимуму.

### Контрольные вопросы

- 1 Чем вызвано изменение индуктивности индуктивных преобразователей?
- 2 От каких параметров индуктивного датчика зависит индуктивность индуктивного преобразователя?
- 3 Объясните принцип действия трансформаторного индуктивного датчика.

## 4 Лабораторная работа № 4. Изучение и исследование датчиков скорости тахогенераторов

**Цель работы:** ознакомление с принципами действия и устройством тахогенератора постоянного тока ТПП-5.

### 4.1 Лабораторное оборудование

Коллекторный тахогенератор постоянного тока с зубцовым ротором ТПП-5; электрический двигатель постоянного тока ДПМ-25-Н1; регулируемый источник питания; осциллограф; мультиметр.

### 4.2 Теоретические сведения о тахогенераторах

В мехатронных и робототехнических системах в большинстве случаев приходится определять скорости вращающихся деталей или узлов, поэтому под тахометрическими датчиками обычно понимают датчики угловой скорости. Они служат для измерения и стабилизации скорости привода в заданных пределах. Контроль скоростных показателей существенно повышает плавность хода и точностные характеристики приводов, является необходимым условием при построении систем управления позиционно-контурного типа.

Принцип действия большинства промышленных датчиков скорости основан на законе Фарадея (5), в соответствии с которым ЭДС индукции прямо про-



порциональна скорости изменения магнитного потока  $\Phi$ .

$$\varepsilon = -\frac{d\Phi}{dt}. \quad (5)$$

Наиболее известным типом углового датчика скорости является тахогенератор (ТГ). Среди основных задач, решаемых с помощью ТГ, необходимо выделить измерение угловой скорости вала, осуществление обратной связи по скорости, а также электромеханическое преобразование (интегрирование и дифференцирование).

На рисунке 30 изображен датчик угловой скорости постоянного тока, представляющий собой маломощную электрическую машину с независимым возбуждением или с постоянными магнитами (рисунок 31, *a*). Конструктивно он состоит из статора, выполненного в виде ферромагнитного каркаса с  $2p$  полюсами, ротора – в виде многослойного цилиндра и щеточно-коллекторного узла.



Рисунок 30 – Тахогенератор постоянного тока ТПП-5

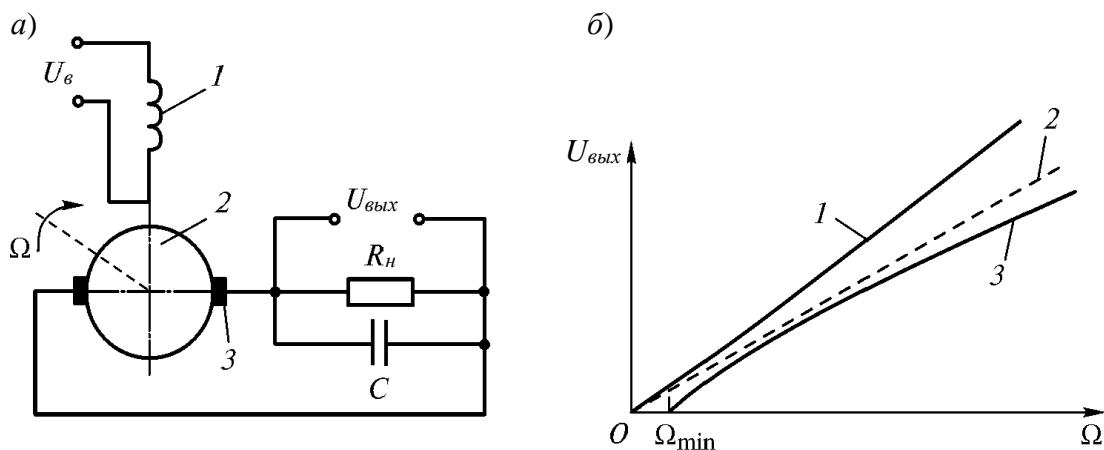


Рисунок 31 – Тахогенератор постоянного тока

Функция преобразования ТГ постоянного тока (как и других индукционных машин) зависит от конструктивных особенностей и величины нагрузки. Данная функция преобразования является линейной, она справедлива при до-

пушении, что магнитный поток возбуждения  $\Phi_{\text{в}}$ , а также сопротивления якорной (роторной) обмотки  $R_{\text{я}}$  и нагрузки  $R_{\text{н}}$  постоянны. Реальный режим работы ТГ постоянного тока значительно отличается от идеального. Следствием этого является зависимость  $K_T = f(\Omega)$  (кривая 3 на рисунке 31, б).

### Порядок выполнения работы

1 Собрать установку согласно схеме на рисунке 32.

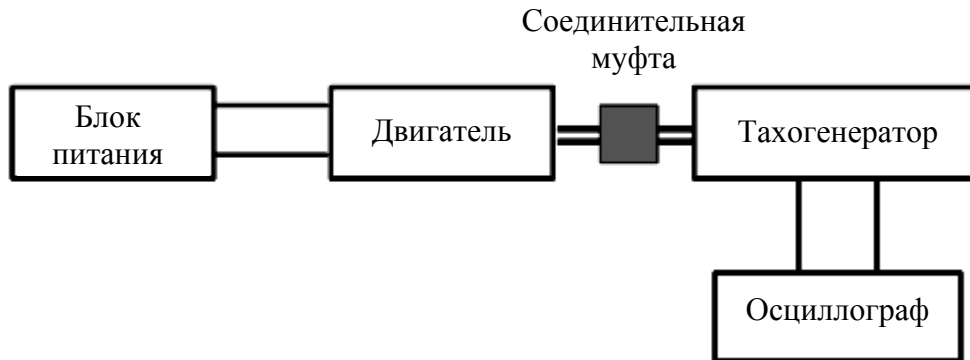


Рисунок 32 – Схема экспериментальной установки

2 С помощью мультиметра определить сопротивление роторной обмотки  $R_{\text{я}}$ .

3 Включить источник питания и осциллограф в сеть с напряжением 220 В 50 Гц.

4 Ознакомиться с характеристиками двигателя и тахогенератора по таблице 4.

Таблица 4 – Основные характеристики двигателя и тахогенератора

Тип	$U_{\text{ном}}$ , В	$M_{\text{ном}}$ , мН·м	$n_{\text{ном}}$ , об/мин	$I_{\text{ном}}$ , А	$I_{\text{хх}}$ , А	$M_{\text{н}}$ , мН·м	$I_{\text{н}}$ , А	Ресурс, ч
ДПМ-25-Н1								
ТГП-3								

5 Вращая регулировочное колесико, изменять напряжение питания двигателя в диапазоне 1...20 В с шагом 1 В.

6 Вычислить значения угловых скоростей двигателя при заданных напряжениях питания, учитывая, что значение  $n_{\text{ном}}$  соответствует значению  $U_{\text{ном}}$ .

7 Снять значения  $U_{\text{вых}}$  и занести результаты измерений в первые три столбца таблицы 5.

8 Построить график функции преобразования датчика и определить значение крутизны характеристики  $K_T$ .

9 Определить максимальное изменение крутизны  $K_T$  при изменении угловой скорости датчика с помощью формулы

$$\Delta K_{T \max} = \Delta U_{\text{вых max}} / \Delta U_{\text{ин}}. \quad (6)$$



Таблица 5 – Таблица записи результатов эксперимента

Номер эксперимента	$U_{ин}, В$	$U_{вых}, В$	$n$ или $\Omega$	$\Delta K_{T \max}$	$R_{я}, Ом$	$\Delta a, В$	$\epsilon_{нл}, \%$
1							
2							
3							
...							
20							

10 Определить абсолютную погрешность  $\Delta a$ , соответствующую зоне нечувствительности при малых скоростях ротора (рисунок 31, б).

11 Вычислить относительную погрешность нелинейности тахогенератора  $\epsilon_{нл}$ .

12 Заполнить остальные столбцы таблицы 5.

13 Сделать выводы.

### **Контрольные вопросы**

1 Принцип действия и область применения ТГ постоянного тока.

2 Влияние угловой скорости ротора на функцию преобразования датчика.

3 Причина появления случайных и систематических погрешностей ТГ постоянного тока.

4 Как рассчитываются аддитивная и мультипликативная погрешности датчика?

## **5 Лабораторная работа № 5. Изучение и исследование фотоэлектрических датчиков положения**

**Цель работы:** ознакомление с принципами действия и устройством кодового оптического датчика положения и перемещения типа ППК-15.

### **5.1 Лабораторное оборудование**

Фотоэлектрический датчик (энкодер) ППК-15 с блоком индикации; источники питания напряжением +5 В и –15 В.

### **5.2 Теоретическая часть**

В большинстве случаев датчики положения и перемещения являются элементами цифровых систем управления, что требует преобразования выходного сигнала в цифровую форму. Именно такой сигнал формируется в кодовом оптическом датчике положения и перемещения (кодовом ОДП, или энкодере). Кодированная шкала кодового ОДП представляет собой стеклянное основание с кодовой маской, выполненной в виде нескольких (обычно до 20) дорожек с прозрачными и непрозрачными сегментами. Число дорожек, как правило, опре-



деляет разрядность выходного двоичного кода. Осветитель (инфракрасные светодиоды) создает лучистый поток, падающий на шкалу. В момент съема информации луч, проходя через прозрачные сегменты кодовых дорожек шкалы и ограничивающую защитную диафрагму, освещает фотоприемники (фотодиодные матрицы), усиленные сигналы с которых принимаются за двоичные единицы.

Отсутствие сигнала с фотоприемника соответствует двоичному нулю. В результате каждому перемещению соответствует определенная комбинация двоичных единиц и нулей, являющаяся его цифровым кодом.

Для углового энкодера разрешение шкалы составляет  $2\pi/2^n$ . Следовательно, разрешающая способность датчика определяется числом разрядов его кодовой шкалы. Зависимость разрешающей способности от разрядности кодовой шкалы датчика приведена в таблице 6.

Таблица 6 – Таблица определения разрешающей способности кодового ОДП

Число разрядов шкалы	1	2	4	8	10	16
Разрешающая способность, °	180°	90°	22°30'	1°24'	21'5'	20'

Вид конструктивной схемы кодового ОДП определяется главным образом числом разрядов шкалы, а также способом кодирования и считывания. Наиболее часто в энкодерах используют два типа шкал: с прямым двоичным кодом и с кодом Грея (рисунок 33).

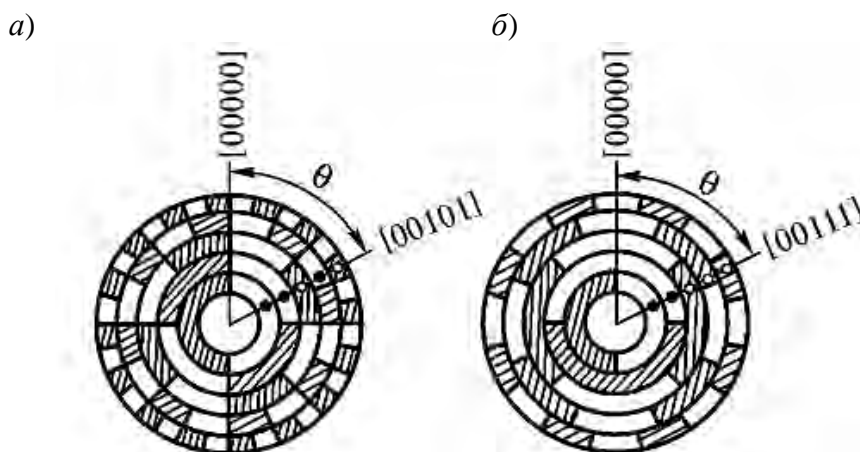


Рисунок 33 – Кодированный диск для двоичного кода (а) и для кода Грея (б)

Шкалы с прямым двоичным кодом представляют собой оптические рейки или диски, разделенные на равновеликие площадки – полосы для реек и сектора для дисков, на которых записаны бинарные слова, соответствующие прямому двоичному коду. Светлый элемент шкалы, пропускающий свет, переводит фотозаэлементы в состояние «1», темный – в состояние «0». Число площадок определяет разрешающую способность кодового ОДП.

Несмотря на простоту кодирования и считывания, шкалам с обычным двоичным кодом присущ крупный недостаток, связанный с появлением ложных кодов. Обусловлено это невозможностью изготовления идеальных шкал и про-

является во время движения шкалы (в момент изменения состояния «1» на «0» или «0» на «1») одновременно в нескольких разрядах.

Так, при изменении кода 15 на 16, т. е. 01111 на 10000, происходит замена значений сразу в пяти разрядах (рисунок 33, а). Если же случайно (из-за погрешности шкалы), например во втором разряде, состояние «1» не меняется на «0», тогда вместо значения 16 будет считано 18 (10010).

Существенно, что ошибка измерения превышает цену деления шкалы, равную одному младшему значащему разряду.

Вероятность возникновения неоднозначности считывания информации в кодовом ОДП особенно велика при высоких скоростях движения шкалы.

Для устранения неоднозначности считывания применяют специальные методы считывания и специальные коды. Так, при использовании циклического кода Грея ошибка считывания не превышает младшего значащего разряда независимо от того, в каком из разрядов она произошла (таблица 7).

Таблица 7 – Таблица перевода десятичных чисел из двоичного кода в код Грея

Десятичное число	Двоичный код	Код Грея	Десятичное число	Двоичный код	Код Грея
0	00000	00000	9	01001	01101
1	00001	00001	10	01010	01111
2	00010	00011	11	01011	01110
3	00011	00010	12	01100	01010
4	00100	00110	13	01101	01011
5	00101	00111	14	01110	01001
6	00110	00101	15	01111	01000
7	00111	00100	16	10000	11000
8	01000	01100			

Функциональная схема датчика приведена на рисунке 34.

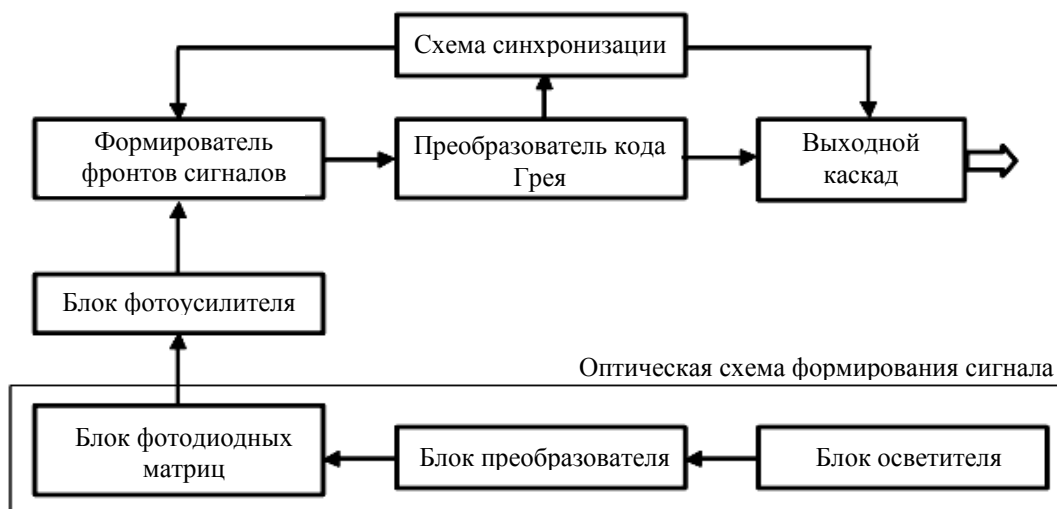


Рисунок 34 – Функциональная схема оптического энкодера ППК-15 [6]

Оптическая схема формирования сигнала состоит из блока осветителя (светодиодов), блока преобразователя, включающего оптические диски и диафрагмы, и блока фотодиодных матриц. Первая из них имеет 11 отдельных светочувствительных элементов, расположенных на одном кристалле кремния, а вторая – 5 элементов. Фотодиодные матрицы воспринимают комбинацию световых сигналов, соответствующую в данном положении вала коду на обоих дисках, и преобразуют ее в электрический сигнал. При этом с кодового диска первой ступени (точного отсчета) считывается 11 младших разрядов, а с диска второй ступени (грубого отсчета) – 4 разряда и сигнал согласования отсчетов.

Сигналы с фотодиодных матриц поступают в электронный блок, где усиливаются и с помощью триггеров Шмита формируются в прямоугольный сигнал. Далее код Грея преобразуется в натуральный двоичный код и формируется сигнал – «направление».

### Порядок выполнения работы

- 1 Включить источники питания в сеть с напряжением 220 В 50 Гц.
- 2 Тумблер блока индикации кодового датчика перевести в положение «ON».
- 3 Вращая ротор датчика, установить его в нулевое положение (при этом все светодиоды блока индикации должны быть погашены).
- 4 Поворачивая ротор, определить направление, в котором происходит увеличение цифрового кода (по или против часовой стрелки).
- 5 Установив ротор датчика в нулевое положение, определить количество оборотов ротора, соответствующее полному диапазону измерений.
- 6 Определить зависимость показаний блока индикации от угла поворота ротора датчика во всем диапазоне через каждые 30°.
- 7 Заполнить таблицу 8.

Таблица 8 – Таблица записи результатов эксперимента

$\theta, ^\circ$	Код
0	
30	
60	
90	
120	
150	
180	
210	
240	
270	
300	
330	
360	



8 Построить график и определить линейность характеристики.

9 Установить упор в одном из резьбовых отверстий в торцевой части корпуса датчика.

10 Установить ротор датчика на упор.

11 Перемещая ротор по и против часовой стрелки, снять показания с блока индикации через каждые  $10^\circ$ .

12 Результаты измерений занести в таблицу 9.

Таблица 9 – Таблица записи результатов эксперимента

$\theta, ^\circ$	Код
0	
10	
20	
30	
40	
50	

13 Построить график зависимости двоичного кода от угла поворота ротора датчика. Определить гистерезис и повторяемость показаний датчика, а также аддитивную составляющую погрешности.

14 Сделать выводы.

### ***Контрольные вопросы***

1 Принцип действия и кинематическая схема датчика.

2 Тип цифрового кода.

3 Алгоритм определения направления вращения.

4 Расчет метрологических параметров датчика.

## **Список литературы**

1 Официальный сайт Kawasaki robotics [Электронный ресурс]. – Режим доступа: <http://www.khi.co.jp/english/robot/index.html/>. – Дата доступа: 28.10.2018.

2 **Бронников, А. И.** Методы идентификации объектов в роботизированных системах / А. И. Бронников. – Харьков, 2010. – 137 с.

4 **Келим, Ю. М.** Электромеханические и магнитные элементы систем автоматизации / Ю. М. Келим. – Москва : Высшая школа, 2004. – 352 с.

5 **Аристов, Е. В.** Исследование индуктивных датчиков / Е. В. Аристов, Р. А. Хузин. – Пермь : Пермский гос. техн. ун-т, 2007. – 12 с.

6 Роботы и робототехника [Электронный ресурс]. – Режим доступа: <http://www.prorobot.ru/>. – Дата доступа: 28.10.2018.