Министерство науки и образования РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Уральский государственный педагогический университет»
Институт математики информатики и информационных технологий
Кафедра информатики, информационных технологий
и методики обучения информатики

# МЕТОДИКА ОБУЧЕНИЯ УЧАЩИХСЯ РЕШЕНИЮ ЗАДАЧ ПО РОБОТОТЕХНИКЕ С ИСПОЛЬЗОВАНИЕМ РЕГУЛЯТОРОВ

*Выпуская квалификационная работа по направлению «44.03.01 – Педагогическое образование», профиль «Информатика»*

Исполнитель: студент группы БИ-41
Никулина Е.Ю.
Руководитель: старший преподаватель
кафедры ИИТиМОИ
Шимов И.В.

Работа допущена к защите

«___»_____2017 г.Р

Руководитель_____

Екатеринбург – 2017

# РЕФЕРАТ

**Никулина Е.Ю.** МЕТОДИКА ОБУЧЕНИЯ УЧАЩИХСЯ РЕШЕНИЮ ЗАДАЧ ПО РОБОТОТЕХНИКЕ С ИСПОЛЬЗОВАНИЕМ РЕГУЛЯТОРОВ, выпускная квалификационная работа: 71 стр., рис.38, табл.1, библ.18 назв.

**Ключевые слова:** регуляторы, робототехнические устройства, LEGO MINDSTORMS Education EV3, среда программирования, решение задач, разработка методических рекомендаций, авторские рабочие программы.

**Объект исследования**: процесс обучения робототехнике в средней общеобразовательной школе.

**Цель работы**: разработка методических рекомендаций по решению задач с использованием регуляторов в курсе робототехники.

В работе рассматриваются различные авторские программы по курсу робототехники, приведена сравнительная таблица данных программ. Проведен обзор робототехнических устройств и сред программирования, в следствии чего выявлен наиболее популярный конструктор. Подробно разобраны типы регуляторов, которые применяются в робототехнике.

В ходе работы были разработаны методические рекомендации по организации процесса обучения и методические рекомендации по решению задач с использованием регуляторов в курсе робототехники.

Разработанная методика прошла апробацию в Уральском государственном педагогическом университете, после некоторой адаптации может быть использована преподавателями по курсу робототехнике в школах и учреждениях дополнительного образования.

**Оглавление**

# ВВЕДЕНИЕ

**Актуальность исследования**. В настоящее время роботизированные устройства повсеместно используются на производстве, транспорте и в быту в виде технических приборов с программным управлением, роботов-манипуляторов, умных домов и автомобилей, шагающих и бытовых машин.

В связи с развитием вычислительной техники, появлением дешевых и мощных микропроцессоров и сетей программно-управляемые устройства получили широкое распространение, что обуславливает необходимость их изучения в общеобразовательных учреждениях, внедрения робототехники как в основной учебный процесс, так и во внеурочные занятия. Не вызывает сомнений и необходимость методической поддержки педагогов, преподавателей центров технического творчества и специалистов, ведущих практическую деятельность по реализации образовательных программ в области образовательной робототехники в условиях новых Федеральных государственных образовательных стандартов (ФГОС).

Образовательная робототехника – цикл мероприятий в общеобразовательной школе или учреждениях дополнительного образования, в котором программирование и конструирование, объединяясь, позволяют формировать навыки технического творчества, мотивируют учащихся на изучение точных наук и обеспечивают их раннюю профессиональную ориентацию, способствуют развитию у учащихся моторики, усидчивости и трудолюбия, а также тяги к исследовательской и проектной деятельности.

В настоящее время существуют различные учебные программы по курсу робототехники. В этой работе будут рассмотрены некоторые авторские программы, а также выделены общие темы, изучаемые при обучении в курсе робототехники. Акцент ставится на тему использования регуляторов в робототехнике и решении задач с их использованием.

**Объект исследования –** процесс обучения робототехнике учащихся средней общеобразовательной школы.

**Предмет исследования –** применение регуляторов в решении задач по робототехнике.

**Цель исследования –** разработать методические рекомендации по решению задач с использованием регуляторов в курсе робототехники.

**Задачи исследования:**

1. Провести обзор различных авторских учебных программ.
2. Выделить типы регуляторов, используемых при решении задач по робототехнике.
3. Провести обзор конструкторов и сред программирования, применяемых в курсе робототехники.
4. Разработать методические рекомендации по организации процесса обучения по робототехнике.
5. Разработать методические рекомендации по решению задач с использованием регуляторов в курсе робототехники.
6. Провести апробацию разработанных материалов.

# ГЛАВА 1.ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРЕПОДАВАНИЯ КУРСА РОБОТОТЕХНИКИ В ШКОЛЕ

## *1.1. Обзор тем, изучаемых в курсе робототехники*

На сегодняшний день в школах и учреждениях дополнительного образования, набирает популярность курс робототехники. Рассматриваемая дисциплина является достаточно молодой по сравнению с другими предметами, преподаваемыми в школе. По основным, уже устоявшимся дисциплинам в школе, существуют примерные рабочие программы, рекомендованные к реализации учебного процесса. По курсу робототехники таких программ не существует, поэтому можно обратиться к опыту педагогов, которые внедрили робототехнику в учебный процесс. Для этого проведем обзор некоторых авторских рабочих программ по курсу робототехники с целью выявления общих тем.

***Рабочая дополнительная общеобразовательная программа «Робототехника»***

1. Автор: Яковлев Н.М., учитель информатики высшей квалификационной категории.
2. Аудитория: 13 - 17 лет (обучающиеся 8-11 классов).
3. Цель: развитие интереса к естественнонаучным дисциплинам, научно-техническому творчеству в области робототехники на основе приобретения профильных знаний, умений и навыков.
4. Задачи:
    - освоить конструирование роботоустройств на базе микропроцессора EV3;
    - освоить среду программирования Lego Mindstorms Education EV3;
    - получить навык программирования посредством управления роботом в зависимости от поставленных условий;
    - развивать творческие способности и логическое мышление обучающихся;

- развивать умение выстраивать гипотезу и сопоставлять с полученным результатом;
- развивать образное, техническое мышление и умение выразить свой замысел;
- развивать умение применять знания из различных областей знаний;
- развивать умения излагать мысли в четкой логической последовательности, отстаивать свою точку зрения, анализировать ситуацию и самостоятельно находить ответы на вопросы путем логических рассуждений;
- получить навыки проведения физического эксперимента.

5. Темы:
   1) Введение.
   2) Принципы робототехники Lego Mindstorms.
   3) Программирование в среде Lego Mindstorms Education EV3.
   4) Игры роботов.
   5) Решение инженерных задач.
   6) Творческие проекты. [1.а)16].

***Рабочая программа курса «Образовательная робототехника и 3D моделирование»***

1. Автор: Николаева Наталья Евгеньевна, учитель информатики.
2. Аудитория: 12 - 17 лет (обучающиеся 6 - 11 классов).
3. Цель: образование детей и молодежи в сфере инновационных технологий и содействие развитию технического творчества, развитие инновационной деятельности в образовательных учреждениях.
4. Задачи:
   - вовлечение детей и молодежи в научно-техническое творчество, ранняя профориентация;

- расширение политехнического кругозора, закрепление в практической деятельности знаний, полученных при изучении основ наук;
- развитие навыков проектной и конструкторской деятельности в сочетании с готовностью к исполнительской деятельности;
- формирование умений самостоятельной индивидуальной и согласованной коллективной работы, развитие навыков делового общения;
- подготовка специалистов дополнительного образования по вопросам образовательной робототехники и 3D моделирования.

5. Темы:
    1) Подключение, интерфейс и основное меню Scratchduino и LEGO Mindstorms.
    2) Создание программ с помощью меню Scratchduino и LEGO Mindstorms.
    3) Датчики Scratchduino и LEGO Mindstorms.
    4) Шагающий робот.
    5) Команды действия. Команды ожидания. Управляющие структуры.
    6) Модификаторы. Контейнеры.
    7) Подпрограммы и П-регуляторы.
    8) Простейшие алгоритмы "Квадрат". Простейшие алгоритмы "Бесконечность".
    9) Простейшие алгоритмы "Движение по комнате".
    10) Движение по линии с одним датчиком.
    11) Движение вдоль стенки.
    12) Движение по линии с двумя датчиками
    13) Удаленное управление. Использование П-регулятора при удаленном управлении моделью [1.а)17].

***Повышение уровня компетенций школьников в области соревновательной робототехники***

1. Автор: Подгорный А.Н., Агафонов М.А.
2. Аудитория: 12 – 18 лет.
3. Цель: повышение результативности выступлений учащихся на соревнованиях по робототехнике.
4. Задачи:
    - способствовать овладению учащимися навыков построения эффективных мобильных платформ;
    - обучить базовым элементам теории автоматического управления;
    - развить навыки командной работы;
    - сформировать навыки ведения проекта с точки зрения управления временем;
    - развить навыки программирования систем управления.
5. Темы:
    1) Вводное занятие (в том числе техника безопасности).
    2) Разработка конструкции робота для подготовки к соревнованиям.
    3) Чувствительные элементы измерительной системы.
    4) Написание программы робота.
    5) Итоговое занятие.

*Таблица 1. Сравнение рабочих программ по робототехнике*

| | Яковлев Н.М. | Николаева Н.Е | Подгорный А.Н., Агафонов М.А. |
|---|---|---|---|
| Возраст учащихся | 13-17 | 12-17 | 12-18 |
| Уровень знаний по предмету | Начальный | Начальный | Базовый |
| Кол-во часов | 140 | 70 | 90 |
| Материально – техническое обеспечение | Lego Mindstorms EV3 | Scratchduino и LEGO Mindstorms | Lego Mindstorms NXT, EV3 |
| Форма реализации | кружок | кружок | кружок |
| Срок реализации | 1 год | 1 год | 1 год |
| Направленность дополнительной образовательной программы | Познавательная деятельность, конструирование и моделирование | Развитие технического творчества и инновационной деятельности в ОУ | Научно-техническая |

Таким образом на основании рассмотренных в данном параграфе рабочих программ по курсу робототехники, мы пришли к выводу, что не существует единой рабочей программы, но при этом большая часть практикующих педагогов, используют боле менее схожее содержание для своих курсов. Далее будут рассмотрены типы регуляторов.

### *1.2. Типы регуляторов в курсе робототехники*

Под регулятором будем понимать устройство, которое с помощью чувствительного элемента (датчика) измеряет регулируемую величину и в соответствии с законом регулирования вырабатывает воздействие на регулирующий орган объекта. Система, состоящая из объекта и регулятора, называется системой управления [1.а)22].

Одной из главных задач теории автоматического управления является управление с помощью обратной связи. В таких задачах можно выделить четыре основных компонента:

- управляемую систему (или как говорят специалисты, объект управления) – о, чем мы хотим управлять;
- цель управления – то, чего мы хотим достичь при помощи управления, т.е. желаемое поведение объекта управления;
- список измеряемых переменных (или выходов) – то, что мы можем измерять;
- список управляющих переменных (или входов) – то, что мы можем менять для того, чтобы воздействовать на объект управления.

Еще один важный компонент – регулятор – устройство, вырабатывающее входные величины, необходимые для достижения заданной цели. Этот пятый элемент обычно появляется после того, как теоретическое решение задачи найдено. Под решением проблемы управления будем понимать нахождение закона управления (алгоритма управления), обеспечивающего достижение цели. Как только искомый закон найден, он может быть использован для вычисления управляющих входов по

измеренным значениям выходов объекта управления. Полученные значения входов в виде некоторых сигналов подаются на исполнительные устройства. В формировании этих сигналов может принимать участие микропроцессор, производящий достаточно сложные вычисления в соответствии с заданным алгоритмом [1.а)22].

**Типы регуляторов**

С развитием технологий в инженерных науках широко используется множество различных регуляторов в построении автоматизированных устройств. В робототехнике применяются следующие регуляторы:

- пропорциональный регулятор (П-регулятор);
- интегральный регулятор (И-регулятор);
- пропорционально-интегральный регулятор (ПИ-регулятор);
- дифференциальный регулятор (Д-регулятор);
- пропорционально-дифференциальный регулятор (ПД-регулятор);
- пропорционально-интегрально-дифференциальный регулятор (ПИД-регулятор);
- релейный регулятор.

Рассмотрим регуляторы, применяемые в робототехнике, более подробно.

***Пропорциональный регулятор***

Под пропорциональным регулятором понимается реализация пропорционального управления, когда величина коррекции заданного порогового значения пропорциональна величине ошибки. В результате небольшая ошибка будет приводить к небольшой корректировке, а большая ошибка приведёт к большому корректирующему действию. То есть выходной сигнал регулятора up(t) будет пропорционален ошибке слежения e(t):

$u_p(t) = k_p e(t)$, $k_p > 0$, где $k_p$ – усиление регулятора, положительная константа.

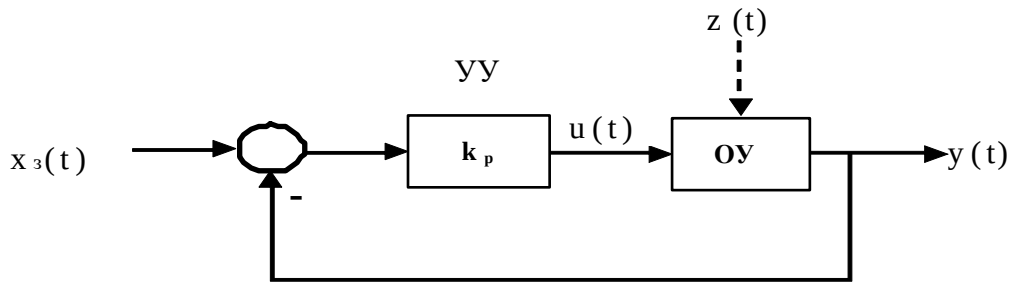


**Рис 1. Схема работы пропорционального регулятора**

Фактически пропорциональный регулятор является усилителем. При использовании пропорционального регулятора (П-регулятора) выход системы будет всегда меньше заданного порогового значения. Причиной является то, что П-регулятор имеет на выходе ненулевое значение только в случае ненулевого входа. Если ошибка слежения отсутствует, то пропорциональный регулятор не создаёт выходного сигнала. Практически на входе мы всегда будем иметь ненулевое значение в стационарном состоянии. Следствием этого будет сохранение некоторой статической ошибки. Эта ошибка может быть уменьшена за счёт увеличения усиления регулятора $k_p$, но при слишком больших значениях это может привести к автоколебаниям и потере устойчивости системы. Значение регулируемой величины в П-регуляторе никогда не стабилизируется на заданном значении, поэтому для устранения статической ошибки нужен другой способ. Конечно, можно изначально задать большее пороговое значение, но существует вид регулятора, который позволяет устранить статические ошибки автоматически. Это приводит нас к интегральному управлению.

Интегральное управление – это стратегия, основанная на общей накопленной ошибке. П-регулятор основывается только на отслеживании мгновенной ошибки. Если ошибка слежения мала, то пропорциональный регулятор теряет свою эффективность (так как результирующие корректировки также будут малы). Одним из способов «усиления» таких маленьких статических ошибок является использование накопленной ошибки с течением времени, что обеспечивает значительный уровень управляющего сигнала. С другой стороны, если ошибка слежения равна нулю, то

накопленное значение также будет нулевым. В этом заключена идея интегрального управления.

***Интегральный регулятор***

Выход интегрального регулятора пропорционален интегралу ошибки слежения за промежуток времени:

$u_i = k_i \int_0^t e(\tau) d\tau$ где $k_i > 0$ – константа.

В случае дискретного времени в цифровых системах интеграл заменяется суммой ошибок, и интегральный регулятор реализуется в виде сумматора значений ошибок. Этот подход легко реализуется в виде схемы рекурсивного обновления:

$E_t = \delta t \, e_t + E_{t-1}$,

$u_{i,t} = k_i \, E_t$ ,

где $E_t$ – накопленная ошибка на временном шаге t, $k_i$ – интегральное усиление и $u_{i,t}$ – выход интегрального регулятора в момент времени t.

Эта дискретная схема обновления предполагает, что управляющие действия производятся периодически. Коэффициент δt представляет собой промежуток времени между последовательными управляющими воздействиями, выраженный в единицах измерения времени (если мы измеряем время в секундах и производим 100 управляющих действий в секунду, то δt = 0,01, если же мы измеряем время в днях и производим одно обновление в день, тогда δt = 1). Конечно, δt могло бы быть включено в коэффициент усиления регулятора $k_i$ , но это означало бы, что при изменении частоты обновления усиление регулятора тоже бы изменилось. Поэтому лучше их разделять: δt включает временной интервал между последовательными обновлениями, а $k_i$ независимо управляет вкладом интегрального члена на выход регулятора.

Выход интегрального регулятора зависит не только от мгновенного значения ошибки, но и от интеграла (или суммы) наблюдаемых ошибок слежения с начального момента времени. Эта зависимость от предыдущих значений приводит к тому, что интегральный регулятор обладает

нетривиальной динамикой, которая может изменить качественное поведение всей замкнутой системы. В частности, интегральный регулятор может приводить к колебаниям, даже если управляемая система им не подвержена. При сохранении положительной ошибки слежения интегральная составляющая в регуляторе начнёт расти и результатом будет положительное значение входа системы, которое сохраняется даже после того, как ошибка слежения будет устранена. Вследствие чего выходной сигнал системы управления будет «промахиваться», и ошибка слежения станет отрицательной. В свою очередь ошибка слежения уменьшает значение интегрального компонента. В зависимости от выбранных значений для коэффициентов усиления регулятора $k_p$ и $k_i$, эти колебания могут затухать более или менее быстро. Настройка регулятора представляет собой поиск таких значений параметров, которые бы приводили к приемлемому динамическому поведению системы с обратной связью.

Однако интеграция ошибок – это процесс опасный. Ошибки имеются всегда, и просто накопление их приводит к снижению стабильности системы или вообще делает систему нестабильной.

Чистое И-регулирование приведет к тому, что колебания системы будут становиться все больше и больше, пока система не пойдет вразнос. Именно поэтому интегратор используется вместе с пропорциональным звеном. В этом случае мы получаем пропорционально-интегральный закон управления (ПИ-регулятор).

***ПИ-регулятор***

Пропорционально-интегральным регулятором называется реализация параллельно включенных в схему пропорционального и интегрального регуляторов.

$$u_{PID}(t) = u_p(t) + u_i(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau$$

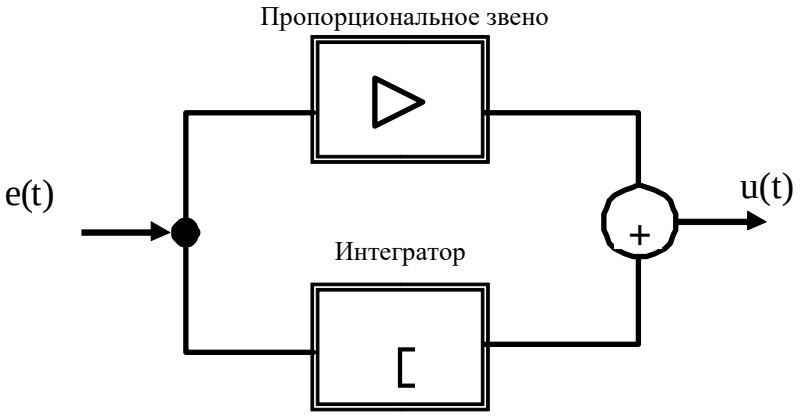


**Рис 2. Схема работы ПИ-регулятора**

Итак, интегральное управление используется, чтобы добавить "долгосрочной точности" управлению и практически всегда используется совместно с пропорциональным управлением. Параллельное соединение двух звеньев позволяет использовать достоинства П и И регуляторов.

ПИ-регуляторы имеют наибольшее распространение в промышленной автоматике.

Преимущества: лучшая по сравнению с П-регулятором точность в установившемся режиме.

Недостатки: худшие свойства в переходных режимах (меньшее быстродействие и большая колебательность).

Для того чтобы система не шла вразнос из-за постоянного накопления ошибок, работу интегратора обычно ограничивают, т.е. определяют минимальное и максимальное накопленного сигнала. Это обычно позволяет предотвратить "вылет системы" - неограниченный рост управляющего воздействия.

***Дифференциальный регулятор***

Дифференциатор – это достаточно простой в реализации элемент. Его задача – умножить разность между текущим значением выходного сигнала и значением выхода в предыдущий момент времени на какой-то постоянный коэффициент.

Подобно интегральному регулятору, дифференциальный регулятор зависит от предыдущих значений и, следовательно, привносит свою

нетривиальную динамику в систему. Интегральные регуляторы часто используются вместе с пропорциональными регуляторами, что нельзя сказать о дифференциальном управлении. Проблемой является потенциальное присутствие высокочастотного шума на входе регулятора. Шумовая составляющая будет осциллировать возле нулевой отметки и, таким образом, в интегральном регуляторе она аннулируется. Однако если взять производную зашумлённого сигнала, то это только усилит влияние шума. Поэтому часто бывает необходимо сглаживание сигнала. Это добавляет сложности и нетривиальной динамики в регулятор. Также появляется риск «потери цели» при дифференциальном управлении. Если сгладить сигнал слишком сильно, то будут теряться изменения в сигнале, которые должен использовать дифференциальный регулятор. Ещё одной проблемой, связанной с дифференциальным управлением, является влияние резкого изменения заданного порогового значения, что приводит к моментальному скачку значения на выходе дифференциального регулятора. В то время как пропорциональное управление занимает центральное место в системах с обратной связью, а интегральное управление требуется для устранения статической ошибки.

Дифференциальный регулятор может быть реализован в виде:

$$u_d(t) = k_d * \big(y(t) - y(t-1)\big)$$

Здесь $u_d(t)$ – значение управляющего сигнала, $k_d$ – постоянный коэффициент, y(t) – текущее значение выходного сигнала (в момент времени t), y(t-1) – предыдущее значение сигнала (в момент времени t-1).

Работает эта компонента (Д-компонента) достаточно очевидно. Если на выходе у нас постоянный сигнал (y(t)=y(t-1)), то значение $u_d(t)$ равно нулю и никакого изменения не происходит. Если же что-то начинает меняться (y(t)≠y(t-1)), то соответственно начинает изменяться величина дифференциальной компоненты. Причем, чем больше разнятся между собой значения выходного сигнала, тем больше будет вклад этой компоненты. Отсюда, следует вывод о том, что использование одной лишь Д- компоненты

в регуляторе совершенно неприемлемо. Если ошибка постоянна (всегда y(t)=y(t-1)), то эта компонента никак не прореагирует на это. Поэтому Д-компонента работает в паре с пропорциональной компонентой. И тогда мы получаем следующую схему регулятора, который называется пропорционально-дифференциальным.

***ПД-регулятор***

В пропорционально-дифференциальном регуляторе сигнал ошибки поступает как на пропорциональное звено (усилитель), так и на дифференцирующее звено (дифференциатор). Далее выходные сигналы этих компонент складываются, формируя управляющее воздействие u(t). Таким образом, работа ПД-регулятора описывается следующим соотношением:

$$u_{pd}(t) = u_p(t) + u_d(t) = k_p * e(t) + k_d * (y(t) - y(t-1))$$

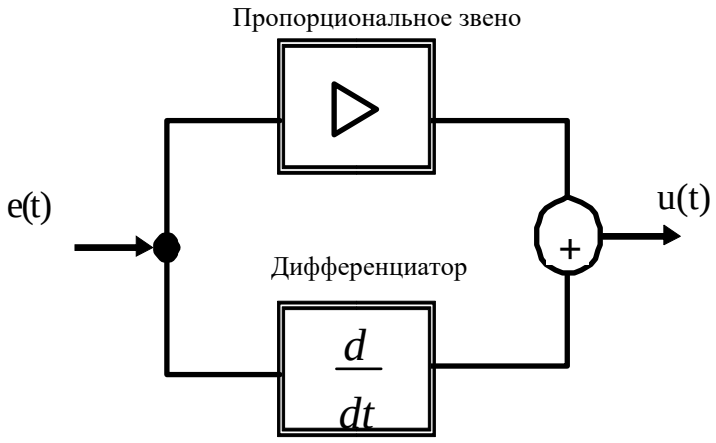


**Рис 3. Схема работы ПД-регулятора**

*Достоинства*. Этот закон управления имеет наивысшее быстродействие. ПД-регулятор реагирует не только на величину отклонения e(t), но, что наиболее важно, на скорость ее изменения.

Недостатками ПД-регулятора являются малая точность и чувствительность к шумам. Дифференциальная компонента управления является самой проблемной и капризной из всех типов управления. И связано это как раз с тем, что Д-компонента очень чувствительна к скорости изменения ошибки, ведь, как ни странно, не на каждую ошибку надо реагировать.

1. Проблема шумов. Дело в том, что все шумит. Шумят датчики, выдавая не всегда то, что необходимо. Шумит трасса (линия нарисована не всегда четко, имеются всякие неоднородности), шумят исполнительные схемы и механизмы. Д-компонента реагирует на все эти шумы, заставляя систему совершать ненужные действия, реагировать на то, на что реагировать вовсе не обязательно. Особенно остро эта проблема касается высокочастотных шумов. И Д-регулятор упорно будет на них реагировать. В то же время изгибы трассы относятся к низкочастотным помехам (по сравнению со скоростью процессов в системе управления), и регулятору надо бы реагировать именно на них. Выходом является установка низкочастотного фильтра на входе Д-компоненты (этот фильтр будет отсекать высокочастотные помехи).

2. Чувствительность к частоте сбора информации. Поскольку в ПД-законе фигурирует сигнал "в предыдущий момент времени", то очень важно, чтобы интервал времени сбора информации выдерживался как можно более строго. Если время "будет плавать", то ни о какой адекватности реакции системы речи быть не может. Считается, что временные интервалы должны выдерживаться с точностью не менее 1%. Это само по себе не так просто. При программной реализации выходом может является использование прерываний. Именно в обработчике прерываний должна содержаться короткая (и быстрая) процедура сбора информации от датчиков. Таким образом, главная проблема ПД-регулятора заключается в том, что он усиливает шумы. И если от сильных шумов в системе не удается избавиться, то Д-компоненту, пожалуй, лучше вовсе не использовать.

***ПИД-регулятор***

Регулятор, включающий все три компонента (пропорциональный, интегральный и дифференциальный), называют ПИД-регулятором.

Варьирование его параметров позволяет реализовывать все остальные законы. Он объединяет, все достоинства и недостатки законов, его составляющих.

Каждый из элементов регулятора (пропорциональное, интегральное и дифференциальное звенья) выполняет свою задачу и оказывает свое специфическое воздействие на функционирование системы: пропорциональный закон отвечает за настоящее (реагирует на текущую ошибку), дифференциальный – за будущее (реагирует на тенденцию изменения ошибки), а интегральный – за прошлое (накапливая предыдущие ошибки и сглаживая высокочастотные шумы) [1.а)5].

Выход этого регулятора представляет собой комбинацию трёх его составляющих:

$$u_{PID}(t) = u_p(t) + u_i(t) + u_d(t)$$
$$= k_p e(t) + k_i \int_0^t e(\tau)d\tau + k_d * \big(y(t) - y(t-1)\big)$$

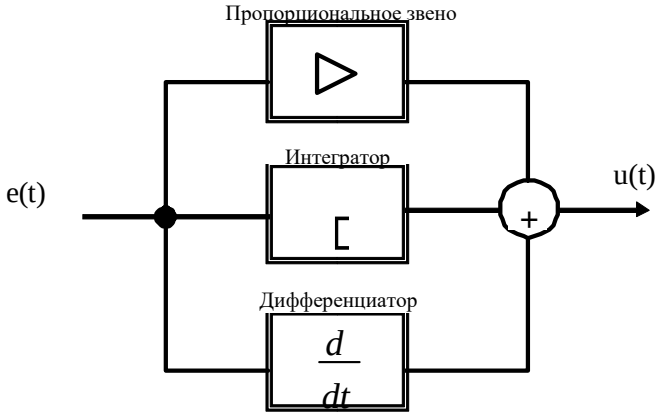


**Рис 4. Схема работы ПИД-регулятора**

ПИД-регулятор – это старое изобретение. Он был создан еще в 1910 году, однако значительно позже, лишь в 1942 г. была разработана методика его настройки (Зиглер и Никольс). Несмотря на свою распространенность, долгое время ПИД-регулятор был достаточно сложным и дорогим устройством. Но после появления микропроцессоров в 80 – х гг. развитие и распространение (где надо и где не надо) ПИД-регуляторов стало происходить нарастающими темпами. Сейчас ПИД-регулятор относится к наиболее распространенному типу регуляторов. Считается, что почти 90% регуляторов, находящихся в настоящее время в эксплуатации, используют ПИД алгоритм. Причиной столь высокой популярности является прежде

всего их низкая стоимость. Правда, в последние годы наблюдается тенденция вытеснения "классических" ПИД-регуляторов их аналогами – регуляторами, использующих т.н. "нечеткую логику". Нечеткие регуляторы, не уступая в быстродействии, все-таки проще в настройке и более универсальны.

***Релейный регулятор***

Одной из главных задач теории автоматического управления является управление с помощью обратной связи.

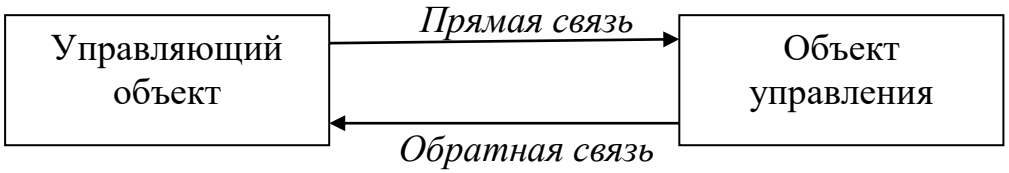


**Рис 5. Схема работы релейного регулятора**

Важнейшим компонентом этой системы является **регулятор.** Релейным двухпозиционным регулятором называется регулятор, у которого регулирующий орган под действием сигнала от датчика может принимать одно из двух крайних положений: "открыт" - "закрыт".

В данном параграфе работы требовалось изучить различные типы регуляторов, которые применяются в курсе робототехники. В процессе изучения были сделаны выводы. Регуляторы занимают важное место в изучении робототехники. Их применение упрощает решение той или иной задачи, главное нужно правильно подобрать регулятор, который будет уместно использовать в задаче [1.а)15].

## *1.3. Обзор конструкторов и сред программирования для курса робототехники*

В век инновационных технологий, становится все больше роботизированных устройств, на заводах, в освоении космоса, здравоохранении, общественной безопасности, развлекательных целях, обороне, медицине и в других отраслях человеческой деятельности. Поэтому не удивительно, что на сегодняшний день в школах и учреждениях дополнительного образования, набирает популярность робототехника. Уже сейчас существует более 23 наборов для самостоятельной сборки и программирования роботов. Такие конструкторы как:

- LEGO Education WeDo
- LEGO Education WeDo 2.0
- LEGO Mindstorms Education EV3
- TETRIX
- MATRIX
- Robotis OLLO
- Robotis Bioloid
- Hovis Lite
- VEX EDR
- VEX IQ
- VEX PRO
- Технолаб
- Arduino
- #Структор
- Multiplo
- Makeblock
- HUNA-MRT
- RoboRobo
- fischertechnik
- Engino Robotics Platform
- ТРИК
- MOSS
- Robo Wunderkind

Рассмотрим конструкторы, которые чаще используются в школах и учреждениях дополнительного образования.

***LEGO Mindstorms Education EV3***


**Рис 6. Конструктор LEGO Mindstorms Education EV3**

Робототехнический конструктор EV3 является третьей версией образовательной серии LEGO Mindstorms Education, выпускаемой с 2013 года. Конструктор LEGO Mindstorms EV3 предназначен для детей старше 10 лет. Он создавался для внедрения в учебный процесс средних школ, при этом домашнее конструирование рассматривалось как дополнение к школьным занятиям [1.а)4].

В базовый набор входит контроллер EV3 Intelligent, три сервомотора и два датчика касания, датчик цвета, гироскоп, ультразвуковой датчик, аккумулятор, кабели и 500 пластиковых деталей.

Собранные модели роботов обладают такими возможностями:

- распознавать 7 цветов и реагировать на изменение освещённости;
- улавливать ультразвуковые волны и обнаруживать предметы на расстоянии до 2,5 м преодолевать препятствия и двигаться по лабиринту;
- управляться мобильными устройствами с ОС Android или iOS.

Специально для LEGO компания National Instruments разработала графическое ПО, не требующее кодировки. Блочное программирование роботов производится с помощью готовых алгоритмов в виде иконок на основе программной среды MyBlocks.

Конструктор Mindstorms EV3 обладает широкими возможностями для творчества благодаря совместимости с другими конструкторами LEGO и представляет собой комплексное решение для обучения подростков основам робототехники. Он позволяет собирать высоко детализированные уникальные модели роботов самого разного назначения. К недостаткам Mindstorms EV3 можно отнести небольшую мощность сервомоторов, достаточно хрупкие детали и соединения [1.a)1].

***HUNA-MRT***


**Рис 7. Конструктор HUNA-MRT**

Линейка конструкторов HUNA-MRT достаточно широкая: это и простейшие наборы с минимумом электроники, и продвинутые наборы с контроллерами, датчиками и исполнительными устройствами. Конструкторы ориентированы на детей от 5-6 лет и до студентов.

Выпускаются как пластиковые, так и металлические наборы. Причем конструкторы разных ступеней совместимы между собой и можно собирать металлопластиковые конструкции. Детали, сенсоры, моторы всех серий унифицированы.

Оригинальными являются сами детали – они допускают соединение с 6 сторон и дают широкие возможности 3D моделирования объектов по своему замыслу.

То, что конструкторы начального уровня не требуют программирования, обеспечивает их доступность и для детей, и для начинающих педагогов – что не маловажно с учетом дефицита кадров в области образовательной робототехники младшего возраста [1.а)14].

Все конструкторы линейки имеют методическое сопровождение для детского сада, школы и кружков.

Оборудование HUNA-MRT соответствует ФГОС и может использоваться в дошкольных образовательных учреждениях и школах.

***ТРИК***

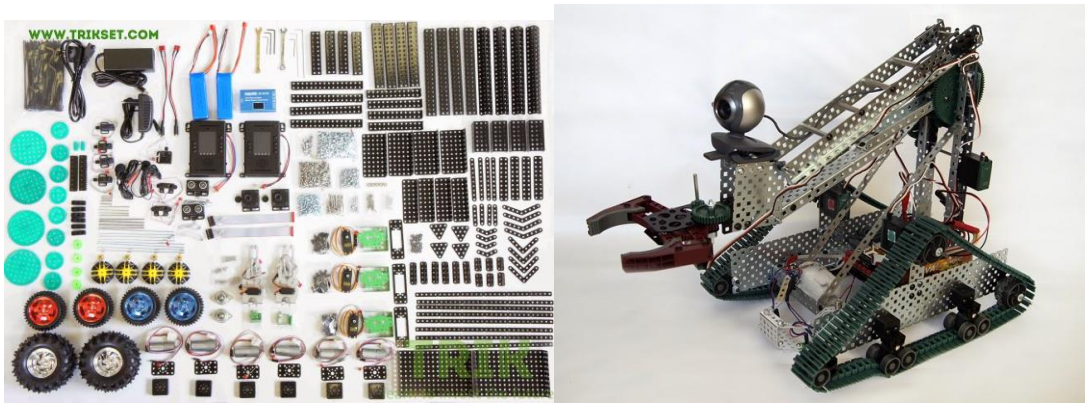


**Рис 8. Конструктор ТРИК**

ТРИК – конструктор, позволяющий без помощи профессиональных инженеров и программистов собирать роботов: от простых радиоуправляемых моделей до сложных кибернетических систем. Законченная платформа для занятий персональной робототехникой, интересная в том числе профессионалам.

Принципиальное отличие от обычных конструкторов в следующем:

- крепкий металл. Детали изготавливаются толщиной 0,75 мм, 0,8 мм, 1,0 мм;
- жесткий профиль. Балки П-образного профиля;
- удобно собирать. Предусмотрены переходники (адаптеры) для крепления типовых моторов, сервомоторов и т.д.

Контроллер ТРИК позволяет легко создавать современных роботов, способных даже «видеть» и «слышать». На его основе сделан конструктор

ТРИК для использования в школах и вузах. Среда визуального программирования TRIKStudio позволяет составлять программы роботов из картинок с готовыми алгоритмами. Простота и удобство делают конструктор интересной игрушкой для широкого потребителя, желающего создать собственного робота или радиоуправляемую модель [1.а)11].

***Robotis Bioloid***


**Рис 9. Конструктор Robotis Bioloid**

BIOLOID – это популярная серия программируемых робототехнических конструкторов компании Robotis. Наборы для робототехники BIOLOID включают в себя микроконтроллеры, датчики, а также уникальные сервомоторы Dynamixel, используя которые можно создавать своими руками различные продвинутые модели автономных или управляемых роботов, например, робот андроид (человекоподобный робот или гуманоид), паук, динозавр, трансформер и другие модели.

Серия представлена разнообразными универсальными наборами, которые подойдут как начинающим робототехникам, так и специалистам, работающим над решением актуальных робототехнических задач.

Существуют 4 модификации этого робота: комплект для начинающих (Beginner Kit), комплексный комплект (Comprehensive Kit), премиум комплект (Bioloid Premium Kit) и эксперт (Expert).

Комплект для начинающих (Beginner Kit), позволяет создать до 14 различных модификаций роботов. Все остальные комплекты позволяют построить до 26 модификаций. Используя этот комплект, вы можете построить роботов с 4 степенями свободы. Он рекомендуется для юных и начинающих робототехников.

Комплексный комплект (Comprehensive Kit), позволяет строить простые механизмы с 1 степенью свободы, а также пауков или гуманоидов с 18 степенями свободы.

Премиум комплект (Premium Kit) состоит из более продвинутых и уже металлизированных деталей, а также различных датчиков: дальномер, двухосный гироскоп, датчик прикосновения и некоторые другие, менее важные компоненты.

Комплект Bioloid эксперт (Expert Kit), предназначен для образовательных и исследовательских целей. В отличие от модели Bioloid Comprehensive Kit, эксперт имеет два дополнительных сервопривода, один дополнительный контроллер, модуль беспроводного соединения, беспроводную камеру, два дополнительных датчика и алюминиевый корпус.

Из данного комплекта можно собрать 26 различных роботов, каждый из них может быть запрограммирован на выполнение отдельных действий. Это своего рода комплект роботов или робот трансформер.

Комплект идет с набором сервомоторов, из которого можно с легкостью создать любого робота. Сервомоторы объединяются друг с другом через интерфейс RS-485 и передают данные к главному блоку управления. Благодаря уникальному идентификационному номеру у каждого сервомотора, любая возможная путаница исключается.

Всё что нужно для того, чтоб построить робота, это комплект элементов и отвёртка. Роботы используют наиболее распространённую и недорогую модель сервопривода Dynamixel AX -12. Он обеспечивает вращающий момент высокой производительности и высокую точность

позиционирования. Несколько сервоприводов объединяются в сеть и имеют множество параметров конфигурации.

Мозг робота управляется микроконтроллером CM - 5, основанным на ATmega128. CM - 5 программируется с помощью программного обеспечение RoboPlus. Все ПО поставляется в комплекте с роботом. CM - 5 оснащён флэш-памятью 128 КБ, 2 портами UART и шестью кнопками управления. Светодиодные индикаторы обеспечивают визуальную обратную связь о текущем состоянии робота.

Dynamixel AX-S1 используется в качестве корпуса для датчиков. Этот сенсорный модуль "все в одном" может определить звуковой источник и измерить расстояние. Он включает в себя: IrDA приёмник, 3 инфракрасных датчика и микрофон [**Ошибка! Источник ссылки не найден.**].

***VEX IQ***


**Рис 10. Конструктор VEX IQ**

Металлические наборы VEX занимают особое место среди образовательных робототехнических конструкторов. Они состоят из перфорированных металлических деталей – профиля и пластин, пластиковых элементов передач – зубчатые колеса, шкивы и колеса. Наборы VEX укомплектованы современными микроконтроллерами Cortex, сервомоторами и разнообразными датчиками. Отдельно стоит отметить, что среди комплектующих VEX есть элементы пневматики и линейные передачи, различные колеса и гусеничные траки. Благодаря вышеперечисленным

качествам металлические наборы VEX обладают уникальными функциональными возможностями.

VEX IQ – это уникальная линейка конструкторов, которая сочетает в себе разнообразие металлических конструкторов VEX и простоту использования пластиковых конструкторов. В комплекты VEX IQ входит большое количество пластиковых деталей, сенсоров, контроллеров. VEX IQ очень просты в использовании, структурные элементы соединяются и разъединяются без специальных инструментов. Огромное количество шестеренок, колес и других соединительных механизмов позволяет конструировать разнообразных мобильных роботов. Robot Brain – высокотехнологичный и мощный контроллер, специально разработанный для использования в учебных целях. Контроллер обеспечивает возможность подключения произвольной комбинации из 12 датчиков, которыми можно управлять с помощью встроенных программ или запрограммировать их самостоятельно, подключив через компьютер и совместимое программное обеспечение.

В дополнение к автономному режиму работы по предварительно запрограммированным командам, VEX IQ роботами можно управляться дистанционно с помощью удобных контроллеров. VEX IQ укомплектован необходимыми датчиками, например, датчиком цвета, гироскопом, потенциометром, ИК - датчиками и многими другими. Наличие разнообразных датчиков открывает обширные возможности для разработки уникальных конструкций роботов.

Серия VEX IQ состоит из трех основных наборов (Starter Kit with Controller, Starter Kit with Sensors, Super Kit). Самым популярным из них, является Super Kit [1.а)3].

***Fischertechnik***


**Рис 11. Конструктор Fischertechnik**

Fischertechnik Robotics, как и LEGO Mindstorm, направлен на то, чтобы ребёнок был привязан к робототехнике конкретной компании и не пользовался конструкторами других разработчиков не менее интересных продуктов. Оба эти производителя создают искусственные препятствия для того, чтобы другие электронные модули, в том числе используемые во взрослой робототехнике, невозможно было бы подключить к моделям из-за несоответствия стандарта разъёмов.

В техническом отношении Fischertechnik Robotics – это «продвинутая» игрушка, цель которой обучить ребёнка механике, электротехнике, химии и физике и совсем немного – программированию. В составе наборов конструктора есть множество элементов, позволяющих собирать совершенно невероятные вещи из области схемотроники: пневматические приводы, электрохимические суперконденсаторы. Из наборов специальных серий, например, «Экологическая энергетика», можно собрать целое производство, например, электромобиль с заправочной станцией, настоящий водородный топливный элемент или электростанцию на солнечных батареях. Развлекательный и образовательный проект Fischertechnik Robotics охватывает широкую аудиторию пользователей – от школьников младших классов до студентов и является самым популярным роботизированным конструктором в Европе. «Материальная» часть конструктора базируется на оригинальной детали Артура Фишера (выдающийся изобретатель и

основатель компании), обеспечивающей соединение элементов по типу «ласточкин хвост». Таким образом, детали могут крепиться друг к другу по всем 6 граням.

Основой роботизированных моделей является программируемый контроллер ROBOTICS TXT, состоящий из главного 2-х ядерного процессора ARM Cortex A8, периферийного процессора Cortex M3, встроенного динамика и 2,4'' цветного сенсорного дисплея. Контроллер оснащён оперативной памятью 128 Мб, модулями WiFi и Bluetooth, имеет 8 универсальных и 4 счётных входа для подключения периферийных устройств, 4 выхода для подключения моторов - сервоприводов. Программирование может осуществляться беспроводным способом или с помощью USB кабеля через мобильные устройства и персональный компьютер.

Преимуществом робототехнических конструкторов «Фишертехник» перед аналогичными игрушками «LEGO» является использование такого же языка программирования, каким пользуются программисты: C-Compiler, PC Library, MS-RDS, в то время как LEGO Mindstorm имеет свой собственный закрытый язык, применяемый только в «LEGO». Программное обеспечение на компакт – диске входит в состав всех наборов Fischertechnik Robotics вместе книжкой, в которой описаны исторические факты, технические решения и другая полезная для ребёнка информация, связанная с тематикой конкретного набора. В настоящее время выпускается 6 наборов конструктора Fischertechnik Robotics [1.a)13].

## ОБЗОР СРЕД ПРОГРАММИРОВАНИЯ

С каждым годом все больше становится программируемых конструкторов, благодаря этому, увеличивается и количество сред программирования. Уже сегодня можно перечислить не менее 10 сред:

- NXT-G
- TrikStudio
- Microsoft Robotics Developer Studio (MRDS)

- RobotC
- BricxCC
- Arduino
- Robolab
- RoboPlus
- LEGO Mindstorms Education EV3
- Urbi
- ROS

Рассмотрим наиболее часто используемые среды на сегодняшний день, которыми являются: RobotC, Robolab, EV3, RoboPlus и TrikStudio.

***Robolab***

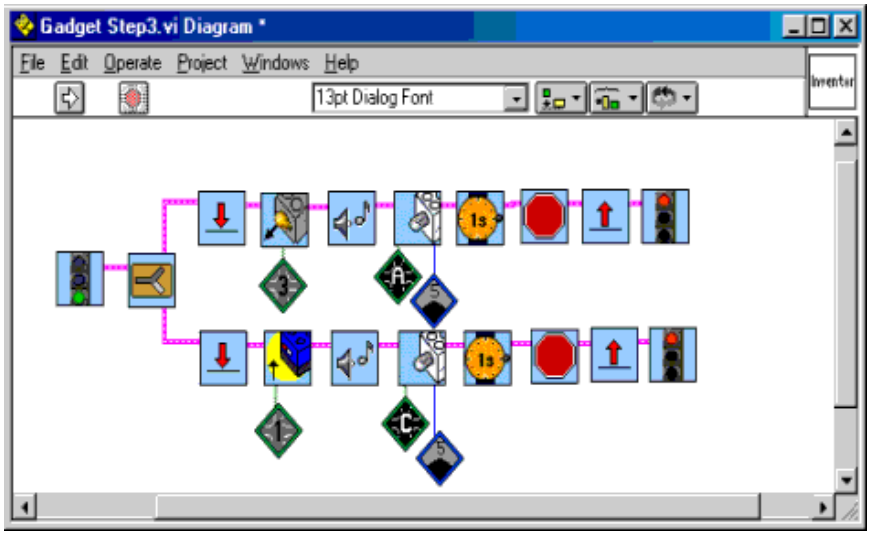


**Рис 12. Robolab**

*Robolab* – это многофункциональная графическая среда программирования, которая создана на основе LabView и ориентирована на самые разные возрасты – от дошкольников до студентов. Robolab позволяет программировать несколько типов микроконтроллеров – Control Lab, RCX, NXT, также проводить независимые расчеты на компьютере. При запуске, Robolab предлагает три уровня работы: Администратор, Программист и Исследователь.

Режим Администратора позволяет настраивать контроллер на работу со средой. С помощью режима Программиста можно создавать программы и

загружать их в микроконтроллер. Режим Исследователя осуществляет запись данных, поступающих с датчиков микроконтроллера, с их последующим анализом.

Режим «Администратор». Первое, что следует сделать администратору – выполнить «Проверку связи с RCX» (подразумевается NXT). Есть три варианта исхода: – успешное завершение со звуковым сигналом на микроконтроллере; – сообщение о необходимости сменить операционную систему (ОС NXT или, иначе говоря, Firmware); – сообщение об ошибке связи. Если связь компьютера и NXT не появилась сразу, нужно воспользоваться меню «Select COM Port» для выбора USB-порта. Лучше выбрать автоопределение, выключить NXT, убедиться, что он хорошо соединен, и снова включить Для поддержки новых возможностей NXT дополнительные установки добавлены в таблицу «Установка RCX/NXT». В ней можно выбрать имя файла загружаемой в NXT программы. По умолчанию стоит имя rbl. Разрешается использовать до шести символов или можно задать его из программы с использованием расширенного NXT – светофора (блок NXT begin). Поскольку Robolab поддерживает несколько устройств (RCX, NXT, Control Lab) диалоговое окно «Choose Hardware» добавлено в установки автоопределения. Оно может появляться в разных ситуациях при потере связи с NXT.

Режим «Программист». Раздел программиста делится на два: Pilot и Inventor, что не совсем точно переведено как Управление и Конструирование. Разделы условно можно обозначить как Новичок и Изобретатель. В них можно не углубляться и пройти дальше, начав сразу с последнего уровня Inventor 4, в котором представлены все основные возможности программирования среды Robolab.

Режим «Исследователь». В режиме исследователь программа позволяет выбрать датчик, с которого предполагается снимать показания. Далее устанавливается интервал, в течение которого показания будут регистрироваться. Помимо табличного вида, данные могут выводиться в виде

диаграммы. Этот режим весьма специфический и больше подходит для математических расчетов на основе произвольных данных и т.п.

Программа похожа на блок-схему, положенную на левый бок. Она читается слева направо, хотя блоки можно располагать, как угодно. Блоки команд находятся в окне Functions Palette (Палитра команд). Они связываются между собой проводами, а также управляются инструментами, находящимися в меню Tools Palette (Палитра инструментов).

Все команды можно разделить на два типа: Жди и Делай. Команды типа «Делай» посылают управляющий сигнал на одно из устройств управления микроконтроллера. Например, «включить моторы», «остановить моторы», «издать звуковой сигнал» и т.п. Это действие, как правило, выполняется практически мгновенно (за исключением звуковых сигналов), после чего программа переходит к следующему блоку. Включенный мотор продолжает работать до тех пор, пока не выполнится команда выключения или программа не закончится.

Команды типа «Жди» не выполняют никакого ощутимого действия, хотя и активно взаимодействуют с оборудованием NXT. Эти команды останавливают ход выполнения программы (точнее, текущей задачи) в ожидании некоторого события. Как только событие происходит, управление переходит к следующей команде. Примеры таких команд: «жди громкого звука», «жди яркого света», «жди заданное время» и т.д. Во время выполнения команды «Жди» все запущенные ранее процессы (включенные моторы и др.) продолжают работать [1.а)12].

***RobotC***

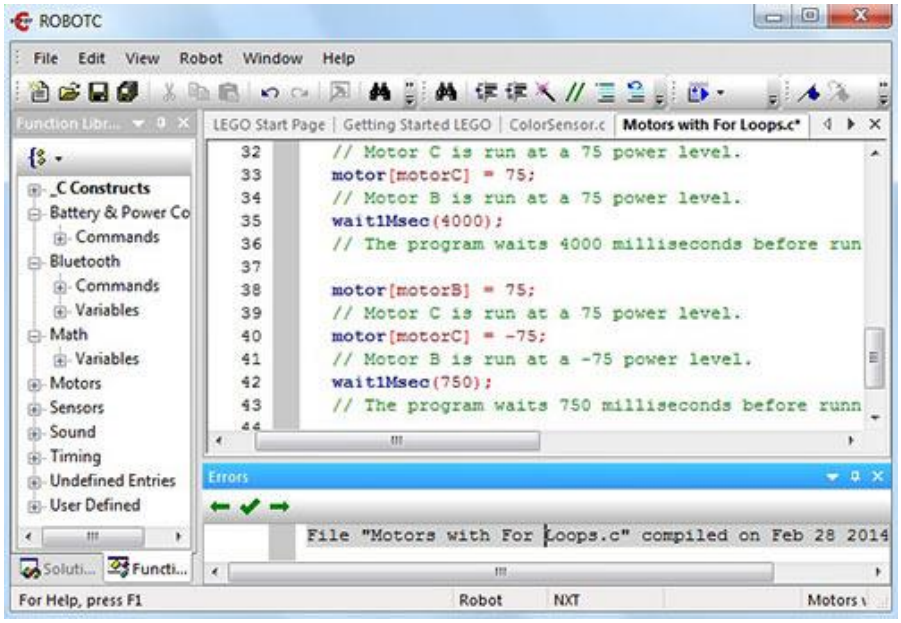


**Рис 13. RobotC**

*RobotC* – это мощный язык программирования, основанный на языке C (СИ) и имеющий среду для написания и отладки программ. В настоящий момент – это единственный язык программирования для роботов, который предоставляет развитый режим отладки во время выполнения программ. RobotC является кроссплатформенным решением, которое позволит студентам и ученикам изучить C-подобный язык, используемый в большинстве образовательных и профессиональных приложений [1.a)2]. RobotC предназначен как для на новичков, так и для подготовленных программистов и имеет два режима работы – базовый и расширенный.

Программное обеспечение имеет схожую с Visual Studio среду и включает в себя мощный интерактивный отладчик, способный функционировать в режиме реального времени, тем самым существенно сокращая время отладки кода. Данная среда обладает развитыми возможностями для работы с математическими выражениями, с помощью которых можно составлять весьма эффективные и сложные программы. В RobotC существует опция предоставления данных с датчиков в «сыром» виде в формате RAW. Среда может поддерживать связь с устройствами посредством инфракрасного канала или Wi-Fi [1.a)2].

### *LEGO MINDSTORMS Education EV3*

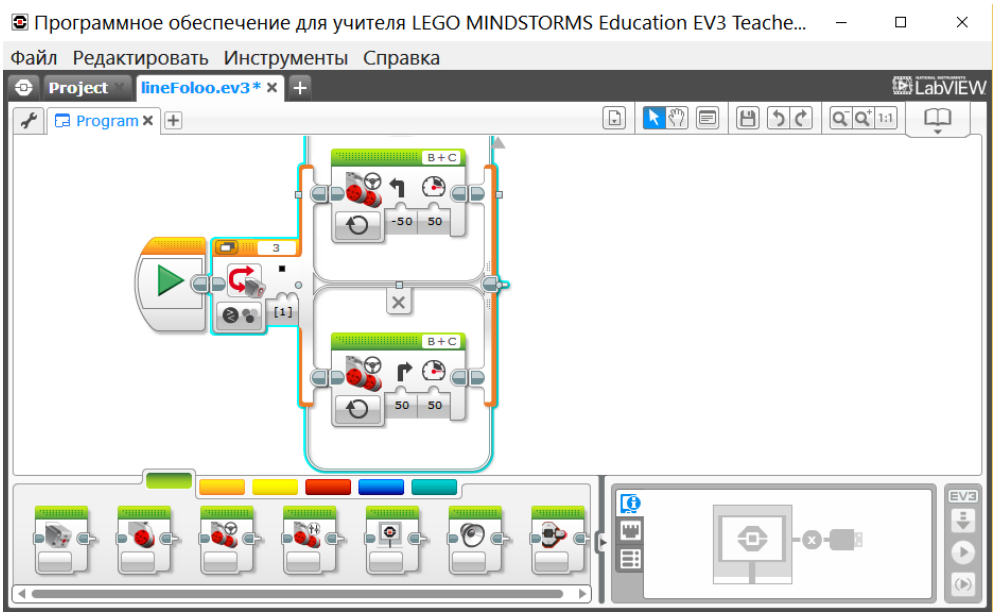


**Рис 14. EV3**

С EV3 в комплекте поставляется новая среда разработки на базе LabView, похожая на NXT-G. Работать она будет, как и NXT-G, на ОС Windows и Mac. Но есть и различия. Появилось такое понятие как проект, который содержит программу для робота, документацию и результаты экспериментов. В проект можно добавлять новый и уже существующие программы. Также был добавлен инструмент zoom, который позволяет масштабировать программу, чтобы, например, увидеть всю программу целиком. Можно программировать NXT блок с помощью новой среды EV3, однако он поддерживает не все особенности нового языка программирования.

С точки зрения программирования, вкратце, можно отметить следующие новшества:

- тесная интеграция между P-блоком (новое название, вместо NXT блока) и средой программирования;
- специальная страница с подключенным оборудованием позволяет отслеживать его статус и получать значения на датчиках в реальном времени;

- оборудование автоматически распознается при подключении, благодаря функции auto-id (автоматическое определение оборудования). То есть. не нужно указывать, к какому порту какой датчик или мотор подключен.

Новый режим отладки:

- подсветка места исполнения позволит определить в каком конкретном месте алгоритма исполняется сейчас программа;
- специальный символ будет отображаться на соответствующем программном блоке, если с заданным портом используется не тот датчик или мотор. Это так же достигнуто с помощью auto-id функциональности;
- есть возможность просматривать значения, передаваемые через каналы данных (data wires);

Новые возможности программных блоков:

- сцепление блоков друг с другом позволило отказаться от "балки исполнения", на которой располагались блоки в среде NXT-G;
- у блоков нет такого понятия, как панель настройки, – поведение теперь настраивается непосредственно на блоке, что привело к увеличению их размера. Удобство заключается в том, что программу теперь становиться легче читать, то есть видно сразу на что реагирует датчик или как ведет себя мотор;
- появились блоки "ждать изменения", которые позволяют реагировать просто на изменение, а не на изменение до определенного значения (обычные блоки Ожидания/Wait в NXT-G);
- улучшения в передачи данных от блока к блоку позволяют упростить преобразование типов;
- есть возможность работать с массивами;
- стал возможен досрочный выход из цикла.

Кроме нового языка программирования появились программы под Android и iPhone\iPad для управления роботом. Также на базе программы Autodesk Invertor Publisher создана программа для создания и просмотра пошаговых 3D инструкций. В этой программе можно масштабировать и вращать модель на каждом этапе сборки, что позволяет строить более сложных роботов по инструкциям [1.a)11].

***RoboPlus***

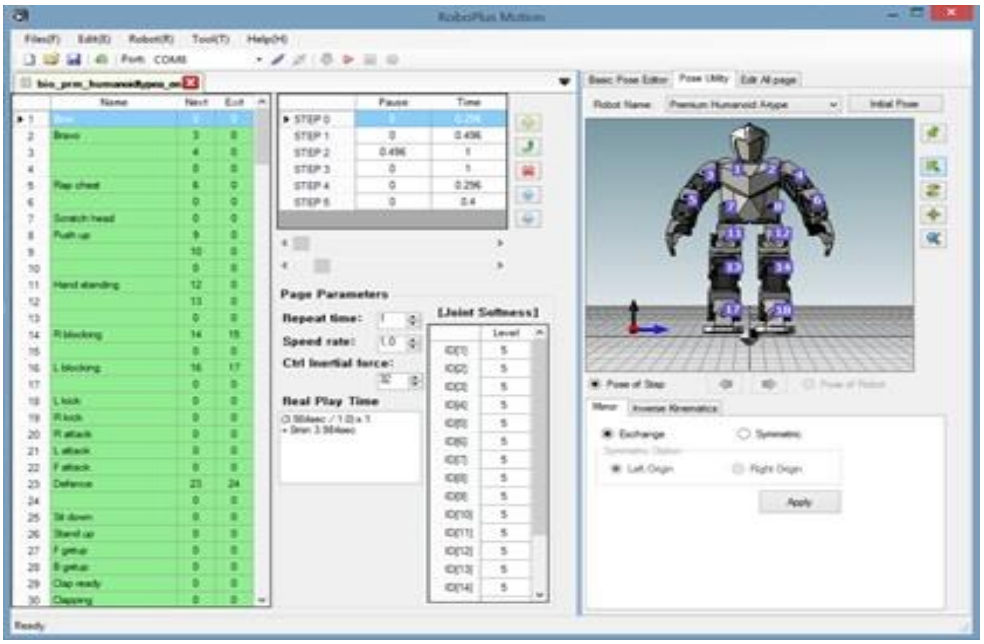


**Рис 15. RoboPlus**

Среда разработки RoboPlus содержит в себе все необходимые инструменты для программирования робототехнических наборов на базе конструкторов фирмы ROBOTIS. С ее помощью можно программировать модели на базе наборов серий OLLO и Bioloid, а также отдельные устройства, например, сервопривода Dynamixel и модули беспроводной связи.

В состав среды разработки RoboPlus входят специальные программы, предназначенные для настройки различных устройств, входящих в состав робота, а также программирования и управления роботами.

- RoboPlus Task
- RoboPlus Manager
- RoboPlus Motion

- RoboPlus Terminal
- Dynamixel Wizard

RoboPlus Task – программная среда для написания и редактирования управляющих программ. Данная программа является основным инструментом для разработки программ для образовательных робототехнических модулей [1.a)20].

Программирование в RoboPlus Task осуществляется с помощью специализированного языка, подобного языку программирования С. Для удобства пользователя в RoboPlus Task в виде графических блоков реализованы базовые возможности набора, такие как таймеры, блоки обработки данных с датчиков и блоки передачи данных между устройствами.

В языке среды RoboPlus все служебные слова, программные блоки, команды и комментарии располагаются в строках. Для того чтобы активировать строку, по ней нужно нажать дважды. После нажатия появится диалоговое окно, позволяющее выбрать различные команды языка.

После выбора определенного набора команд необходимо задать необходимые параметры или условия выполнения команд. В зависимости от типа программируемого контроллера и числа подключенных внешних устройств в диалоговом окне предлагаются различные наборы команд. Перечень команд определяется автоматически и обновляется при подключении новых устройств.

После написания кода программы в RoboPlus Task, ее необходимо загрузить в программируемый контроллер. Для этого контроллер нужно подключить к порту компьютера и воспользоваться функцией автоматического поиска, чтобы компьютер определил тип программируемого контроллера и порт, к которому он подключен.

После того как программа определит тип программируемого контроллера, можно произвести компиляцию программы, тем самым проверить ее на наличие ошибок и подготовить к загрузке в программируемый контроллер.

Полученную программу можно загрузить в контроллер робота, и она запустится сразу же после подачи на него питания.

RobotPlus Manager – программа для настройки оборудования, входящего в состав робототехнических конструкторов ROBOTIS. С помощью данной программы RoboPlus обновляет собственные файлы и производит тестирование оборудования, подключенного к компьютеру в данный момент при помощи контроллера или специализированных переходников. Благодаря использованию RoboPlus Manager возможно изменять параметры контроллера, сервоприводов и производить настройку коммуникационных устройств.

RoboPlus Motion – среда программирования сложных движений робота. Благодаря RoboPlus Motion можно запрограммировать различные действия робота, а после использовать их в основной программе. Зачастую в процессе движения робота участвует множество различных приводов, и задать их скорости вращения и углы поворотов вслепую крайне затруднительно.

RoboPlus Motion позволяет промоделировать движение в процессе написания управляющей программы. В специальном окне отображаются все привода, подключенные в данный момент к роботу. Пользователь в режиме реального времени может задать для каждого из приводов скорость и угол поворота, а после запустить программу и увидеть результат ее работы на реальном роботе. Таким образом, можно разработать программу, реализующую сложное движение робота.

RoboPlus Terminal – программа, предназначенная для получения и отправки данных посредством терминала операционной системы компьютера. Применяется для отладки управляющих алгоритмов, например – для вывода на экран показаний датчиков, то есть для отображения той информации, к которой пользователь обычно не имеет доступа в процессе выполнения программы.

Dynamixel Wizard – программа, предназначенная для настройки и калибровки сервоприводов Dynamixel. С помощью данной программы для

каждого из приводов можно задать ограничения скоростей вращения и углов поворота, а также получить код ошибки, препятствующей работе устройства [1.a)20].

***TRIK Studio***

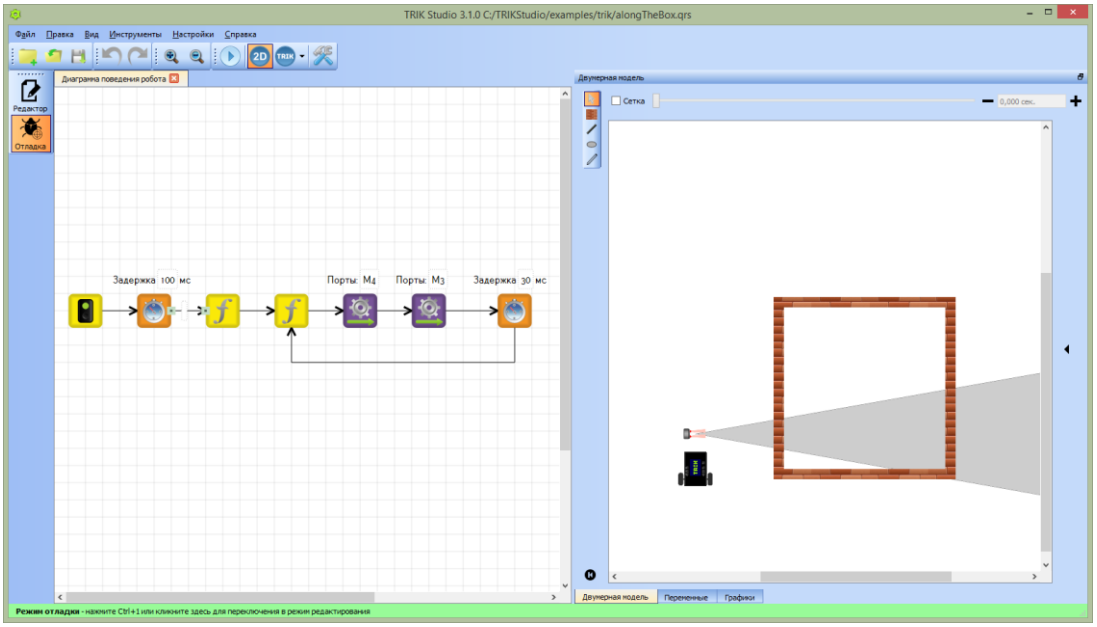


**Рис 16. TRIK Studio**

Среда TRIK Studio позволяет визуально программировать различные робототехнические платформы. Программа в TRIK Studio составляется из блоков и стрелок. Исполнение начинается со специального начального блока и далее передается по стрелкам. Условие рисуется как развилка (две стрелки, отходящие от блока), бесконечный цикл – как связь назад, арифметический цикл – как набор блоков со связью назад и выходной стрелкой. Таким образом, поток управления программы наглядно визуализируется создаваемой диаграммой. Часть диаграммы может быть вынесена в подпрограмму для ее последующего переиспользования. Математические выражения, условия на развилках, значения свойств описываются на встроенном текстовом языке – Lua [1.a)21].

Для облегчения рисования диаграммы среда может распознавать жесты мышью. Например, если пользователь правой кнопкой мыши в произвольном месте окна редактора нарисует стрелку вперед, появится блок включения моторов, если нарисует пиктограмму часов – появится блок ожидания, если

провести линию от одного блока к другому, появится стрелка, их соединяющая. Жесты распознаются довольно сложным алгоритмом, поэтому могут восприниматься системой даже если не будут в точности соответствовать идеальным.

На данный момент среда поддерживает программирование конструкторов Lego Mindstorms NXT, Lego Mindstorms EV3 и конструктора ТРИК. Для каждого конструктора среда предоставляет три режима работы с ним: режим интерпретации, режим автономного исполнения и режим отладки на симуляторе.

В режиме интерпретации программа исполняется на компьютере с отправкой команд роботу по какому-либо низкоуровневому протоколу (USB и Bluetooth для NXT и EV3, Wi-Fi для ТРИК). Значения всех переменных во время интерпретации могут быть просмотрены в соответствующем окне, а также можно отслеживать графики показаний датчиков, строящиеся в реальном времени.

В режиме автономного исполнения среда генерирует код, компилирует его, если целевой язык не скриптовый, загружает по низкоуровневому протоколу на робота и запускает его на исполнение, показывает его во встроенном текстовом редакторе. Код генерируется в читаемом виде, он может быть открыт и отредактирован во встроенном текстовом редакторе с подсветкой синтаксиса и автоматическим дополнением. Для одного конструктора TRIK Studio может поддерживать больше одного текстового языка. Например, в режиме ТРИК возможна генерация в JavaScript, F# и Pascal ABC.NET, в режиме NXT программа может быть сгенерирована в NXT OSEK C или русскоязычном школьном алгоритмическом языке (ШАЯ).

В третьем режиме, доступном для каждого из поддерживаемых конструкторов, режиме симуляции, программа будет выполнена на двумерной модели робота, открываемой внутри окна среды. Двумерный симулятор позволяет пользователю нарисовать произвольную модель мира, состоящую из стенок, регионов и цветных элементов, нарисованных на полу.

Например, могут быть нарисованы все стандартные поля и полосы препятствий, используемые в спортивной робототехнике. Далее указывается, какие датчики подключены к роботу, их пространственное положение и ориентация. Программа затем может быть исполнена на нарисованной модели мира, при этом, так же, как и в режиме интерпретации на реальном устройстве, можно отслеживать значения переменных и графики значений сенсоров. Для удобства отладки скорость течения времени в модельном мире может быть уменьшена или увеличена.

Наличие режима симуляции полезно не только для отладки. Возможность программирования виртуального робота может быть полезна образовательным учреждениям и индивидуальным пользователям, у которых по тем или иным причинам отсутствует реальный робот. К примеру, детям, у которых дома нет роботов, преподаватели могут выдавать домашнее задание, которое нужно решить для виртуального робота. Двумерный симулятор робота может рассматриваться как исполнитель. В частности, робот может рисовать на полу след траектории его перемещения (аналогично исполнителю «Чертежник») [**Ошибка! Источник ссылки не найден.**].

В среде имеется возможность автоматической проверки заданий. Задание описывается на внутреннем языке ограничений и может быть сохранено особым образом для последующего его распространения между учениками.

Рассмотрев конструкторы и проведя некоторые исследования, мы можем сделать вывод, что наиболее распространяемым конструктором является *LEGO Mindstorms Education EV3* со своей средой программирования.

Таким образом, после изучения теоретических основ преподавания курса робототехники, мы можем сделать вывод, что не существует единой рабочей программы для данного курса, но есть большое количество примерных авторских программ, которые имеют схожие темы, но

различаются количеством отведенных часов на курс, а соответственно и полнотой содержания.

Рассмотрев типы регуляторов, применяемых в робототехнике, можно сделать вывод, что регуляторы являются неотъемлемой частью в изучении робототехники, и каждая рабочая программа должна содержать в себе данную тему, так как с их помощью можно более правильно и точно решать задачи.

Также в ходе изучения робототехнических устройств, мы выделили более востребованный и популярный конструктор LEGO Mindstorms Education EV3, так как он более прост в изучении и доступный в цене. Что касается сред программирования, то самой часто используемой, также является LEGO Mindstorms Education EV3, так как данное программное обеспечение включается в состав набора конструктора, а также является доступным для понимания и изучения программирования робототехнических устройств.

# ГЛАВА 2.МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО РЕШЕНИЮ ЗАДАЧ С ИСПОЛЬЗОВАНИЕМ РЕГУЛЯТОРОВ В КУРСЕ РОБОТОТЕХНИКИ

## *2.1. Методические рекомендации по организации процесса обучения робототехнике*

Целью внедрения робототехники в образовательный процесс является создание условий для мотивации, подготовки и профессиональной ориентации школьников для возможного продолжения учебы в СУЗах, ВУЗах и последующей работы на предприятиях по специальностям, связанным с робототехникой.

Для развития данного направления обучения сегодня могут быть предложены различные методические и программные комплексы в зависимости от уровней начального или более продвинутого обучения.

Новые ФГОС ориентированы на результаты образовательной деятельности, в основу получения которых положен системно-деятельностный подход. Данная стратегия обучения может быть реализована с помощью образовательной среды LEGO, содержащей LEGO-конструкторы различных модификаций, используемых в образовательной робототехнике. По сути, они являются основным оборудованием, используемым при обучении робототехнике, как в общеобразовательных, так и в учреждениях системы дополнительного образования. Каждый LEGO-конструктор разработан компанией LEGO с учетом возрастных особенностей и потребностей обучаемого [1.а)6].

В рамках школьного урока и дополнительного образования робототехнические комплексы LEGO могут применяться по следующим направлениям:

- демонстрация;
- фронтальные лабораторные работы и опыты;
- исследовательская проектная деятельность.

Разрабатываемые методические рекомендации будут основываться на конструкторе LEGO Mindstorms EV3.

Основные задачи:

- развитие у школьников инженерного мышления, навыков конструирования, программирования и эффективного использования кибернетических систем;
- развитие мелкой моторики, внимательности, аккуратности и изобретательности;
- развитие креативного мышления и пространственного воображения учащихся;
- формирование умения анализировать, рассуждать и ставить эксперименты;
- формирование умения работать в команде.

Для внедрения в образовательный процесс курса робототехники, необходим кабинет с программным обеспечением и оборудованием, необходимыми для работы с конструкторами.

1. Компьютер
    - Windows Vista или более поздние версии Windows;
    - разрешение экрана 1024x768
    - процессор – 2 ГГц;
    - оперативная память – 2 Гб;
    - свободное место на жестком диске – 2Гб;
    - 1 доступный USB-порт.

2. Программное обеспечение LEGO MINDSTORMS Education EV3.Бесплатно загрузить программное обеспечение на русском языке можно с сайта LEGO.com/mindstorms.

3. Проектор

Подсоединяемый к компьютеру проектор значительно повышает уровень наглядности в работе учителя, предоставляет возможность учащимся демонстрировать результаты своей работы всему классу.

4. Базовый набор LEGO MINDSTORMS Education EV3.

Методика LEGO Education основа на парной работе учащихся, поэтому комплектацию конструкторами рекомендуется осуществлять из расчета один конструктор на двух учащихся. Это позволяет ученикам приобретать навыки сотрудничества и одновременно справляться с индивидуальными заданиями.

Работа также возможна и при наличии 5-7 комплектов, так как на одно и то же устройство в течение урока могут присылать программы различные группы обучающихся. Вместе с этим основной принцип обучения LEGO-технологиям – «шаг за шагом» – обеспечивает учащимся возможность работать в собственном темпе.

Помимо образовательных наборов программируемых устройств важно обеспечить кабинет информатики траекториями для движения исполнителя. Как правило, траектория движения представляет собой черную линию на белом фоне. Чем больше траекторий имеется в образовательном учреждении, тем разнообразнее круг задач, решаемых в рамках дисциплины.

Для безопасной эксплуатации робототехнических устройств рекомендуется работать на траекториях, оснащенных бортиками. Это снизит риск падения и поломки собранных устройств, что, в свою очередь, поможет избежать непредусмотренных трат времени на конструирование нового исполнителя.

Рассмотрим особенности проведения уроков по робототехнике.

Вся мебель в кабинете должна быть расставлена свободно, чтобы учащиеся могли передвигаться, не беспокоясь о том, что могут создать помехи друг другу и, тем самым, избежать травмоопасных ситуаций. Необходимо установить столы и регулируемые по высоте стулья для индивидуальной работы учащихся. В центре кабинета, рекомендуется оборудовать площадку для тренировочных и контрольных заездов.

Для хранения программируемых комплектов конструкторов лучше использовать встроенные шкафы. Если кабинет оборудован не стационарными компьютерами, а ноутбуками, то необходимо предусмотреть также стеллажи для их хранения и коробы для хранения зарядных устройств.

Рассмотрим, на какие этапы можно разделить урок, с использованием регуляторов при решении задач по робототехнике.

*1 этап. Деление учащихся на рабочие мини-группы.*

В связи с ограниченным временем урока, целесообразно использовать приемы, которые позволят быстро произвести деление класса на группы. Например, деление по 2 человека с сидящим справа соседом или же по индивидуальному желанию учащихся.

*2 этап. Постановка задачи.*

Задачу нужно преподнести учащимся так, чтобы замотивировать их к рабочему процессу. Можно устроить соревнования в конце урока и выявить победителя.

*3 этап. Обсуждение способов решения задачи.*

На данном этапе учитель вместе с учениками обсуждает возможные варианты решения задачи. На этом же этапе возможна постановка индивидуальных задач.

*4 этап. Конструирование робота с необходимыми блоками, моторами и сенсорами.*

В зависимости от темы урока, учащиеся могут использовать уже готовые устройства, дополняя их датчиками, необходимыми для решения поставленной задачи или же полностью самостоятельно собрать индивидуального робота для более корректного решения задачи.

*5 этап. Программирование.*

Составление программы на компьютере, расчет коэффициентов для использования регуляторов.

*6 этап. Отработка на полигоне.*

Учащиеся экспериментируют на полигоне своих роботов, тем самым обращая внимание на недочеты, которые допустили в программе при решении поставленной задачи, либо в неточной конструкции робота.

*7 этап. Подведение итогов.*

Итоговый контроль знаний и умений может быть реализован посредством мини-соревнований, где каждая группа учащихся продемонстрирует результаты решения поставленной задачи. Здесь может учитываться время и скорость прохождения устройством дистанции; точность выполняемых действий; точность калибровки датчиков и многое другое.

Эффективность обучения основам робототехники зависит и от организации занятий, проводимых с применением следующих методов:

- объяснительно-иллюстративный – предъявление информации различными способами (объяснение, рассказ, беседа, инструктаж, демонстрация, работа с технологическими картами);
- эвристический – метод творческой деятельности (создание творческих моделей и т.д.);
- проблемный – постановка проблемы и самостоятельный поиск её решения обучающимися;
- программированный – набор операций, которые необходимо выполнить в ходе выполнения практических работ (форма: компьютерный практикум, проектная деятельность);
- репродуктивный – воспроизводство знаний и способов деятельности (форма: собирание моделей и конструкций по образцу, беседа, упражнения по аналогу);
- частично - поисковый – решение проблемных задач с помощью педагога;
- поисковый – самостоятельное решение проблем;
- метод проблемного изложения – постановка проблемы педагогом, решение ее самим педагогом, соучастие обучающихся при решении.

Наиболее часто используемыми методами в курсе робототехники являются проблемный метод обучения и метод проблемного изложения.

Метод проблемного изложения отлично подходит для изучения новой темы, когда преподаватель самостоятельно, с пояснениями решает задачу, но не без участия учеников, которые следят за эти процессом и пытаются воспроизводить те же действия, что и учитель. Благодаря этому методу, учащиеся лучше усваивают новый материал, который в дальнейшем самостоятельно смогут воспроизвести [1.а)7].

В данном параграфе были рассмотрены методические рекомендации по организации урока, где выяснилось, какие задачи содержит внедрение курса робототехники в школах, какое оборудование должна иметь школа для хорошего уровня подготовки школьников по курсу робототехники, а также были рассмотрены эффективные методы обучения.

### *2.2. Решение задач с использованием регуляторов*

На сегодняшний день существует множество задач, при решении которых уместно применение регуляторов. Главной целью регулятора является вырабатывать входные величины, необходимые для достижения заданной цели. Регулятор – это элемент, который обычно появляется после того, как теоретическое решение задачи найдено.

Рассмотрим несколько тем и задач с применением регуляторов.

#### *Тема 1. «Пропорциональный регулятор»*

*Задача*. Робот должен двигаться вдоль стены на расстоянии 15 см, повторяя все ее изгибы и повороты. Реализовать движение робота с левой стороны от стены с помощью П-регулятора и использования датчика ультразвука.

*Решение задачи:*

Для решения задачи соберите базовую конструкцию робота и установите с правой стороны робота датчик ультразвука.

В том случае, когда роботу необходимо следовать вдоль стены с внешними поворотами, датчик ультразвука может быть установлен перпендикулярно движению робота. В этом случае движения будут плавными, с меньшими колебаниями.

Однако если предстоит преодолевать внутренние углы, то датчик ультразвука должен быть установлен под углом, так как при перпендикулярном расположении датчик не увидит поворот, робот застрянет в углу, а моторы продолжат вращение. Движение вдоль прямой будет происходить с большими колебаниями из-за того, что даже при небольшом отклонении от нужного курса диагональ увеличивается больше по сравнению с перпендикуляром. Следовательно, пропорциональное управление будет задавать большие управляющие воздействия на моторы, что вызывает резкие движения.

Если при движении встречаются внешние и внутренние углы, например, при проходе лабиринта, датчик ультразвука следует устанавливать под углом.

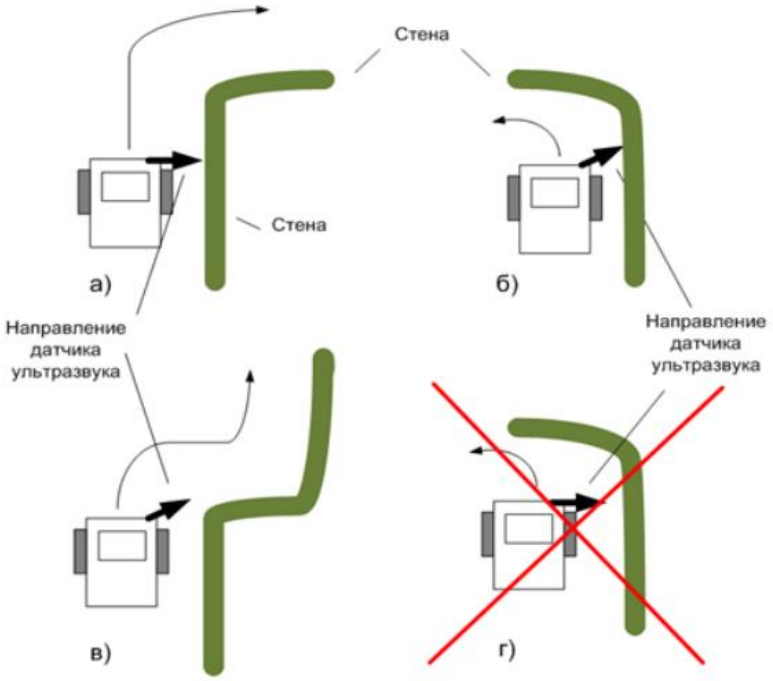


**Рис 17. Схема расположения робота и датчика ультразвука в зависимости от типа поворота**

При любом расположении датчик ультразвука должен устанавливаться впереди робота на некотором расстоянии относительно колес. Это вызвано тем, что робот при управлении инерционен и любое воздействие на моторы отрабатывает с некоторым запозданием.

**Рис 18. Варианты расположения датчика ультразвука на роботе**

Перейдем к алгоритму решения задачи движения робота с левой стороны от стены на основе датчика ультразвука, установленного перпендикулярно движению.

1. Создаем цикл 01. Условие выход из цикла – неограниченно.
2. Первый программный блок – «Ультразвуковой датчик» в режиме «Измерение» – «Расстояние в сантиметрах».
3. Результат измерений заводим в блок «Математика» в режиме «Дополнения (ADV)».
4. Задаем формулу управления: (a-b)*c.
5. Результат, вычисленный в блоке «Математика», подаем на вход «Рулевое управление» программного блока «Рулевое управление».
6. Задаем параметр «Мощность» блока «Рулевое управление» равным 30.

Пусть расстояние до стены становится больше 20 см. На блок рулевого управления подается положительное значение, мотор «B» начинает вращаться быстрее, пропорционально величине ошибки рассогласования. Это значит, что чем дальше робот отъехал от стены, тем с большей скоростью он будет к ней подъезжать. Мотор «C» вращается с заданной скоростью 30 единиц.

В результате робот поворачивается направо в сторону стены, продолжая при этом движение вперед.

Если расстояние до стены становится меньше 20 см, то на блок рулевого управления подается отрицательное значение, мотор «B» продолжает вращаться с заданной скоростью 30 единиц, мотор «C» вращается быстрее пропорционально величине ошибки рассогласования. Чем ближе робот подъехал к стене, тем с большей скоростью он будет он нее отъезжать.

В результате робот поворачивает налево от стены, продолжая при этом движение вперед.

Программа движения робота на заданном расстоянии от стены с использованием алгоритма пропорционального управления на основе одного датчика ультразвука представлена на рисунке 19.

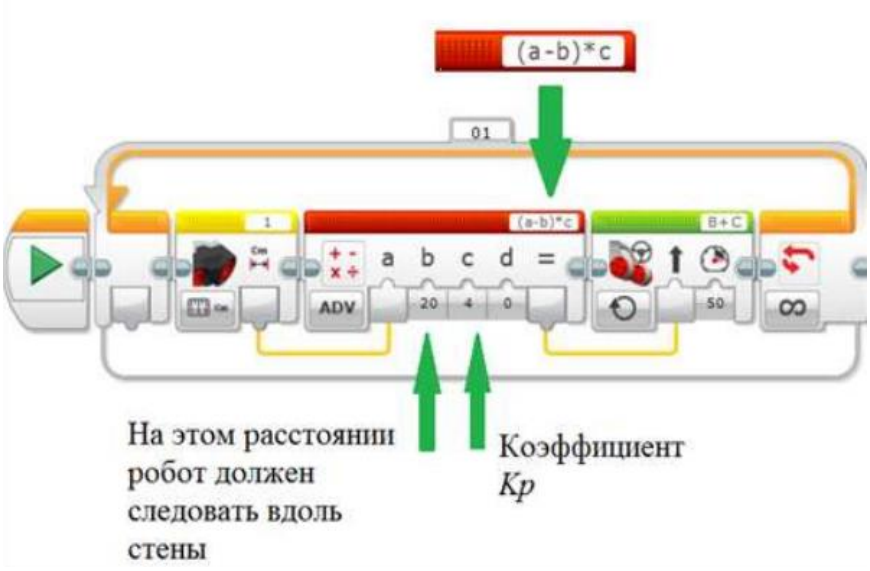


**Рис 19. Программа движения робота на заданном расстоянии слева от стены на основе одного датчика ультразвука**

Коэффициент $k_p$ необходимо подбирать с учетом конструкции робота, покрытия поля и отражающей способности стены. Если стена будет покрыта пористым или мягким материалом, то ультразвук может не увидеть стену или показать значения, значительно превышающие истинные.

Чем меньше коэффициент, том более плавным будут движения, но повороты робот будет проходить под большими углами.

Если необходимо следовать вдоль стены на определенном расстоянии от нее и преодолевать внутренние углы с расположенным под углом

датчиком ультразвука, то необходимо опытным путем определить длину диагонали, измеряемую датчиком ультразвука. Из-за разброса пучка лучей ультразвукового датчика при необходимости движения на расстоянии 20 см от стены параметр «b» программного блока «Математика» нужно задавать вдвое больше.

Из-за этой особенности пропорциональный коэффициент невозможно рассчитать математически, а только опытным путем.

### *Тема 2. «Пропорционально дифференциальный регулятор»*

*Задача*. Организовать плавное движение робота справой стороны от черной линии по трасе представленной ниже, с использованием ПД-регулятора.

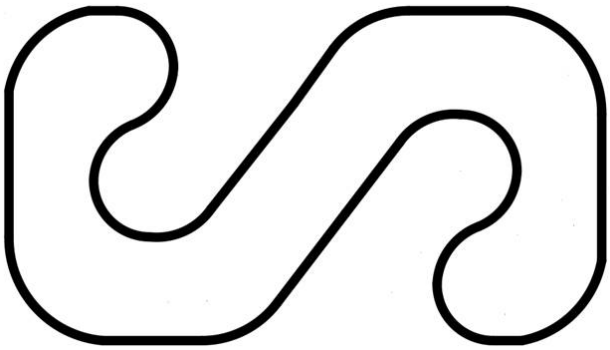

**Рис 20. Трасса 1**

*Решение задачи:*

И-за особенностей трассы и ее крутых поворотов, возникает проблема использования пропорционального регулятора, так как при резких поворотах управляющего воздействия не хватит, и робот потеряет линию. Если мы зададим большой пропорциональный коэффициент, который проходит круче повороты, то на прямом участке робот будет ехать медленно, совершая постоянные колебания. Поэтому необходимо подавать дополнительное воздействие на управление. Для этого к требуемой скорости П-регулятора мы добавляем производную (дифференциал) ошибки по времени (или скорость изменения ошибки), умноженную на положительный коэффициент.

Когда робот едет по прямому участку, скорость изменения ошибки мала, поскольку значение датчика меняется медленно. Как только линия

искривляется, показания датчика быстро изменяются, скорость изменения ошибки сильно увеличивается.

Уравнение ПД-регулятора имеет вид:

***Управление*** $= k_p \times$ ***e*** $+ k_d \times$ ***Differ***, где

$k_p$ – пропорциональный коэффициент;

$k_d$ – дифференциальный коэффициент;

***e*** – ошибка, равная разнице между средним значением серого и текущим показанием датчика цвета;

***Differ*** – дифференциальная составляющая управления или скорость изменения ошибки.

Дифференциальную составляющую можно вычислить как:

***Differ*** = **ошибка (на текущем шаге) – ошибка (на предыдущем шаге)**.

Из приведенного уравнения следует, что

- если значение ***Differ*** положительно и велико, то есть ошибка быстро увеличивается, и робот отклоняется от линии, то дифференциальная составляющая управления будет действовать складываться с пропорциональной составляющей и быстрее возвращать робота на линию;
- если значение ***Differ*** отрицательно и велико, то есть ошибка быстро уменьшается, и робот с большой скоростью приближается к линии, то дифференциальная составляющая управления будет вычитаться от пропорциональной составляющей и притормаживать робота до тех пор, пока скорость изменения ошибки не замедлится, тем самым предотвращая проскакивание робота линии по инерции;
- если значение ***Differ*** мало, то есть робот едет по прямой или плавно поворачивает, четко следуя изгибам линии, то дифференциальная составляющая управления практически не оказывает влияния на управление.

Алгоритм реализации ПД-регулятора

1. Инициируем переменную ***LastError***, в которой будет хранится значение ошибки на предыдущем шаге, и присваиваем ей начальное значение 0.
2. Создаем непрерывный цикл 01.
3. Вычисляем текущее значение ошибки. Из среднего значения, равного 40, вычитаем текущее значение датчика цвета, подключенного к порту 2.
4. Вычисляем величину изменения ошибки: из предыдущего значения переменной ***LastError*** вычитаем значение текущей ошибки. Перезаписываем в переменную ***LastError*** значение текущей ошибки. Таким образом, мы подготовились к следующему шагу цикла, на котором текущая ошибка станет ошибкой.
5. В математическом блоке суммируем:
    - пропорциональное управление, равное значению текущей ошибки, умноженной на пропорциональный коэффициент (в данном случае $k_p$=3);
    - дифференциальное управление, равное изменению ошибки, умноженному на дифференциальный коэффициент (в данном случае $k_d$=15).
6. Результат математического блока подаем на выход блока рулевого управления. параметр «Мощность» в данном случае установлен на 50 единиц.

Самый большой вклад в управление вносит пропорциональная составляющая, поэтому сначала отработайте пропорциональное управление. Это можно сделать, умножив «с» в последнем блоке математики на 0, то есть задав нулевое дифференциальное управление.

Найдите наилучшее значение пропорционального коэффициента для вашего робота, поля, освещения и трассы. А затем немного (на 0,1 – 0,3 единицы уменьшите его.

Далее установите в последнем блоке математики дифференциальный коэффициент (коэффициент перед «с»), равный 1, и постепенно увеличивайте его, добиваясь наилучших показателей движения. В наших условиях он оказался равен 15.

Теперь формула управления (последний блок математики) имеет вид:

***3a+25c.***

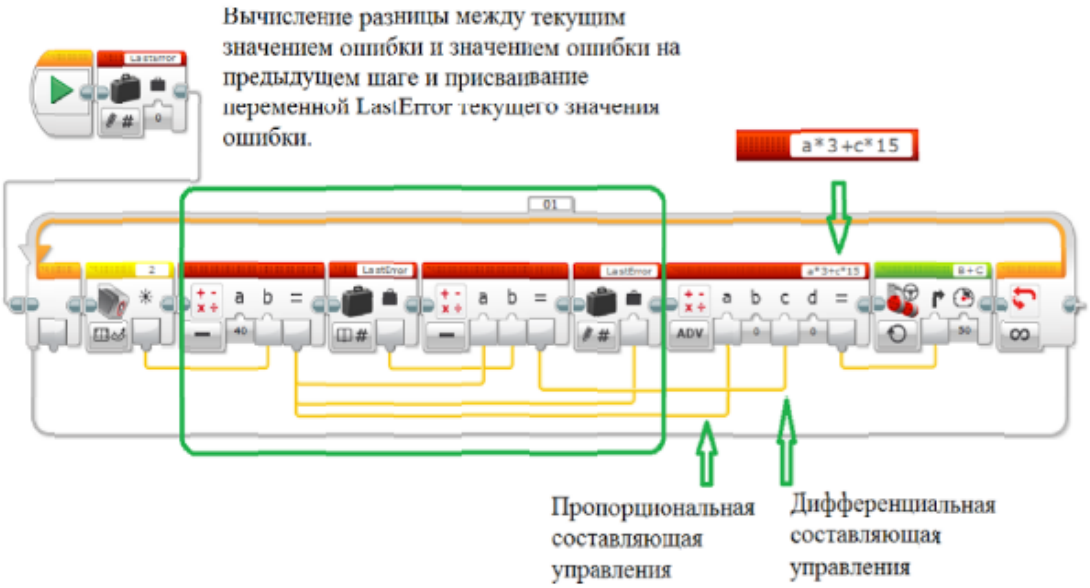


**Рис 21. Реализация ПД-регулятора на основе одного датчика цвета (движение справа от черной линии)**

### *Тема 3 «Пропорционально – интегральный регулятор»*

*Задача*. Робот должен проехать с правой стороны от черной линии по трассе, представленной ниже, используя ПИ-регулятор.

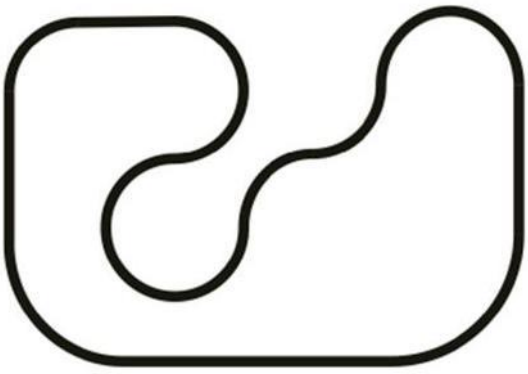

**Рис 22. Трасса 2**

*Решение задачи:*

Из-за конструктивных особенностей моделей, разницы в поверхности шин, разницы во вращении моторов или небольшом наклоне поля робота может всегда слегка «тянуть» в одну сторону. Пропорциональная составляющая будет стараться вернуть робота на линию, но из-за несбалансированности робота эффект появиться только тогда, когда робот немного отъедет от линии.

Пропорциональное управляющее воздействие вернет робота обратно на линию, после чего его снова будет тянуть в сторону. Таким образом, возникают ненужные колебательные движения.

Выход из описанной ситуации состоит во введении интегральной составляющей управления, которая, суммирует все предыдущие ошибки, умножает их на коэффициент и добавляет к пропорциональному управлению. Если робота постоянно тянет в одну сторону, то ошибка будет накапливаться, и управляющее воздействие станет увеличиваться, быстрее возвращая робота на линию и препятствуя возникновению колебаний.

Если робот едет прямо, совершая равномерные колебательные движения относительно линии, то ошибка постоянно меняет знак и при суммировании превращается в 0, не оказывая дополнительного воздействия на моторы.

Уравнение ПИ-регулятора имеет вид:

***Управление*** $=k_p \times e + k_i \times$ ***Integral,*** где

$k_p$ – пропорциональный коэффициент (коэффициент П-регулятора)

$k_i$ – интегральный коэффициент

***e*** – ошибка, равная разнице между средним значением серого и текущим показанием датчика цвета в случае движения по одному датчику.

***Integral*** – интегральная составляющая ошибок или сумма ошибок

Сумму ошибок можно вычислить как:

***Integral = 0,5 × Integral (на предыдущем шаге)+ ошибка (на текущем шаге).***

Алгоритм реализации ПИ-регулятора

1. Инициируем переменную ***Integral***, в которой будет храниться сумма ошибок, и присваиваем ей начальное значение 0. Для этого в программу вставляем блок «Переменная» в режиме «Записать – Числовое значение» и указываем ее значение (0).
2. Создаем непрерывный цикл 01.
3. Вычисляем текущее значение ошибки. В случае использования одного датчика цвета из среднего значения серого, равного 40, вычитаем текущее среднее значение датчика цвета, подключенного к порту 2.
4. Вычисляем сумму ошибок: предыдущее значение переменной ***Integral*** умножаем на 0,5, добавляем текущее значение ошибки и перезаписываем в переменную ***Integral.***
5. В математическом блоке суммируем:
   - пропорциональное управление, равное текущей ошибки, умноженной на пропорциональный коэффициент (в данном случае $k_p$=2);
   - интегральное управление, равное сумме ошибок, умноженной на интегральный коэффициент (в данном случае $k_i$=0,2).
6. Результат математического блока подаем на вход блока рулевого управления. Параметр «Мощность» в данном случае установлен на 50 единиц.

По аналогии с предыдущей задачей, сначала отрабатываем пропорциональное управление. Это можно сделать, умножив «b» в последнем блоке математики на 0, то есть задав нулевое интегральное управление.

Найдите наилучшее значение пропорционального коэффициента для вашего робота, поля, освещения и трассы. А затем немного (на 0,1 – 0,3 единицы уменьшите его.

Далее установите в последнем блоке математики интегральный коэффициент (коэффициент перед «b»), равным 0,05, и постепенно увеличивайте его, добиваясь наилучших показателей движения. В данном случае равен 0,2.

Теперь формула управления (последний блок математики) имеет вид: ***2a+0,2b***.

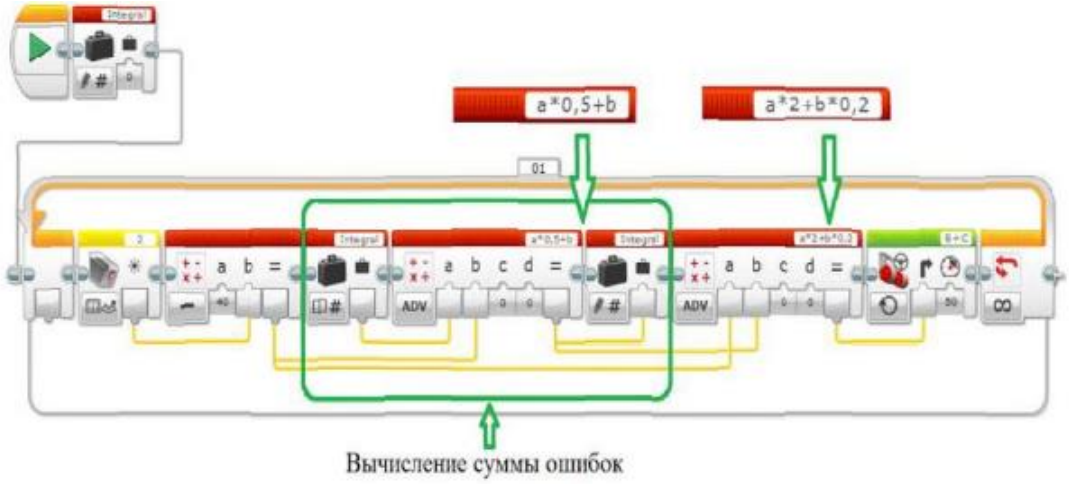


**Рис .23 Реализация ПИ-регулятора на основе одного датчика цвета (движение справа от черной линии)**

### *Тема 4 «Пропорционально-интегрально-дифференциальный регулятор»*

*Задача*. Организовать движение робота по черной линии по трассе, представленной ниже, используя ПИД-регулятор.

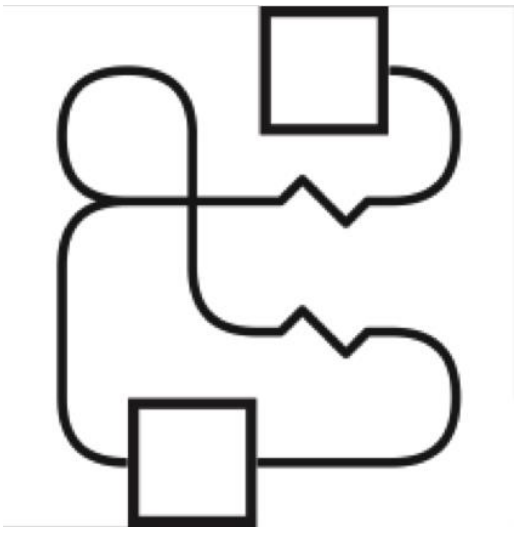

**Рис .24 Трасса 3**

*Решение задачи:*

Можно сказать, что алгоритм ПИД-регулятора позволяет следовать текущему (пропорциональная составляющая), учитывать прошлое (интегральная составляющая) и прогнозировать будущее (дифференциальная составляющая) состояние робота.

Самая серьезная задача при реализации ПИД-регулятора – это подбор коэффициентов. При правильном выборе значений для конкретной поверхности поля, типа трассы, освещения, качества шин, размера колес и расположения датчиков относительно моторов движение робота получается плавным и быстрым как на прямых участках, так и на крутых поворотах.

При выборе коэффициентов следуйте советам, указанным в предыдущих задачах. Прежде всего установите интегральный и дифференциальный коэффициенты в последнем математическом блоке равными 0 и найдите оптимальное значение пропорционального коэффициента, при котором преодолеваются все участки трассы от прямых линий до крутых поворотов.

Постепенно, на 0,01 единицу, увеличивайте интегральный коэффициент до получения наилучших характеристик движения. Затем аналогичную процедуру проделайте с дифференциальным коэффициентом.

В наших условиях (поле, освещение) наилучшие коэффициенты оказались равны $k_p$=2,5, $k_i$=0,15, $k_d$=15.

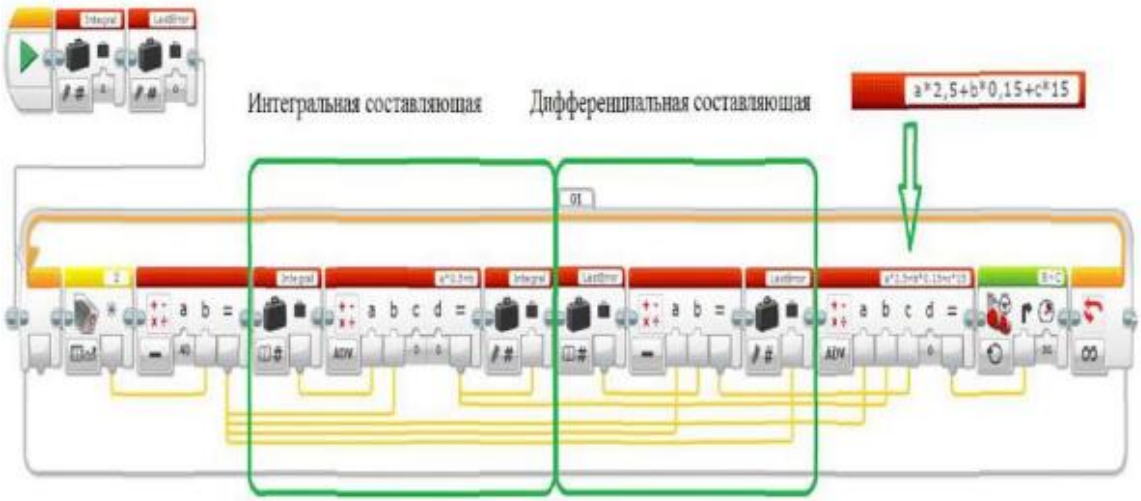


**Рис .25 Реализация ПИД-регулятора на основе одного датчика цвета (движение справа от черной линии)**

### *Тема 5. «Релейный регулятор»*

*Задание.* Построить трехколесную тележку с одним датчиком освещенности. Написать программу движения робота по черной линии, с использованием релейного регулятора, проехать по трассе, представленной ниже.

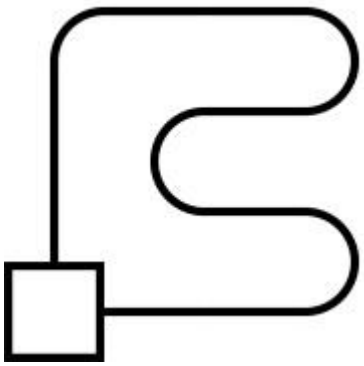

**Рис .26 Трасса 4**

*Решение задачи:*

Для реализации этой задачи нам понадобится датчик освещенности, который включен в комплект базового набора EV3. Для того чтобы понять, как его использовать рассмотрим принцип его работы.

В основе данного датчика находятся два основных элемента: это фоторезистор или фототранзистор (полупроводниковые элементы способные преобразовывать световую энергию в электрический сигнал) и светодиод

(создающий световой поток). Свет, выпущенный этим элементом, отражается от поверхности и попадает обратно в светочувствительный элемент. Из курса физики можно вспомнить, что светлая поверхность (белая) хорошо отражает свет, а темная (черные) хорошо поглощает свет. Уровень сигнала отраженного света от белой поверхности будет больше, чем от черной поверхности (линии), и это свойство мы будем использовать для достижения нашей цели.

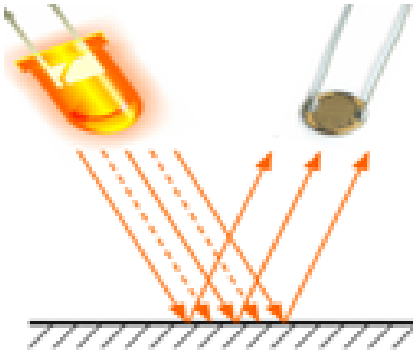

**Рис .27 Схема работы датчика освещенности**

Перед началом программирования, необходимо определить значение черного и значение белого, для этого необходимо поднести робота сначала к белой области, затем запомнив значение белого, поднести робота к черной области, так же запомнив значение черного. Значения отражённого света можно определить следующими способами:

Способ №1

1. Когда робот включен, нажать кнопку №3 три раза выбрав третью вкладку.
2. Нажимаем кнопку №1, выбрав Port View.
3. Выбираем порт, в который включен датчик света, нажимая кнопки 2 и 3.
4. Смотрим значения, запоминаем или записываем их.

Способ №2

1. Подключаем включенного робота к компьютеру через USB кабель.
2. Запускаем программу LEGO MINDSTORMS Education EV3.
3. Заходим во кладку, находящуюся в нижнем правом углу.
4. Смотрим значения датчика.

Следующий этап является составление алгоритма, он достаточно прост.

1. Робот поворачивает налево, если он находится на белом фоне (высокий уровень сигнала)
2. Робот поворачивает направо, если он находится на черном фоне (низкий уровень сигнала)

В результате должна получится следующая траектория движения:

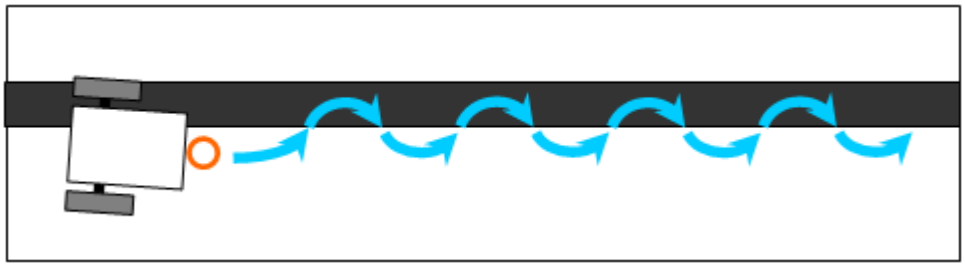

**Рис .28 Траектория движения робота**

Реализуем наш алгоритм в среде LEGO MINDSTORMS Education EV3.

1. Так как робот должен выполнять заданные действия много раз, то нам понадобится цикл, для этого заходим в оранжевую в кладку и переносим иконку «цикл» в начало программы.

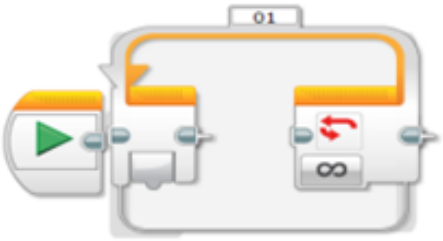


**Рис .29**

2. Затем нам нужно задать условие в зависимости, от которого робот должен поворачиваться то налево, то направо. Для этого переносим иконку переключатель из оранжевой вкладки внутрь.

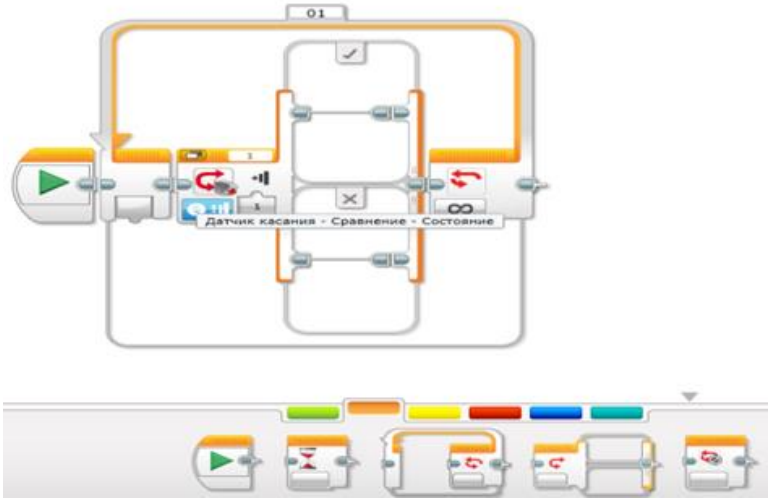


**Рис .30**

3. Заходим в свойства переключателя, и выбираем сравнение яркости отражённого света.

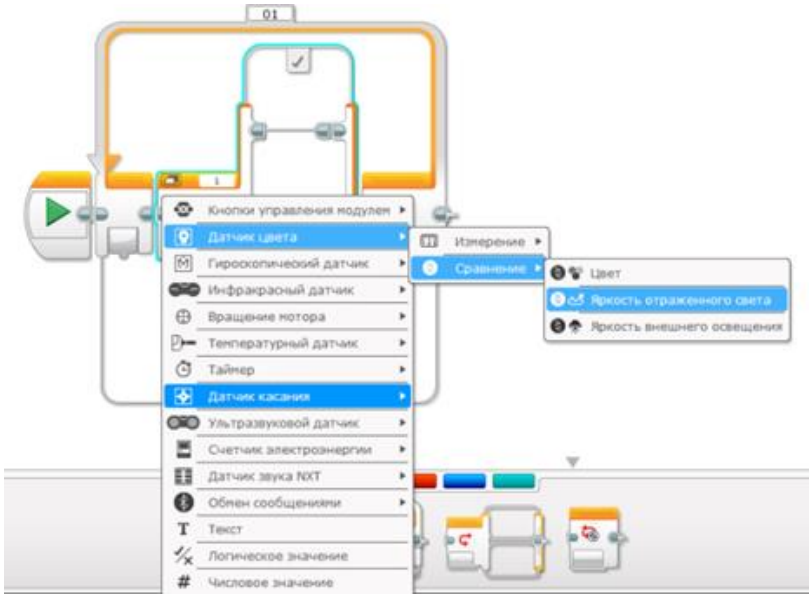


**Рис .31**

4. Задаем условие, если меньше или равно значения черного цвета. Предварительно необходимо проверить какой порт (к которому подключен датчик света) у вас стоит в окошке, если порт будет указан неправильно, программа работать не будет.

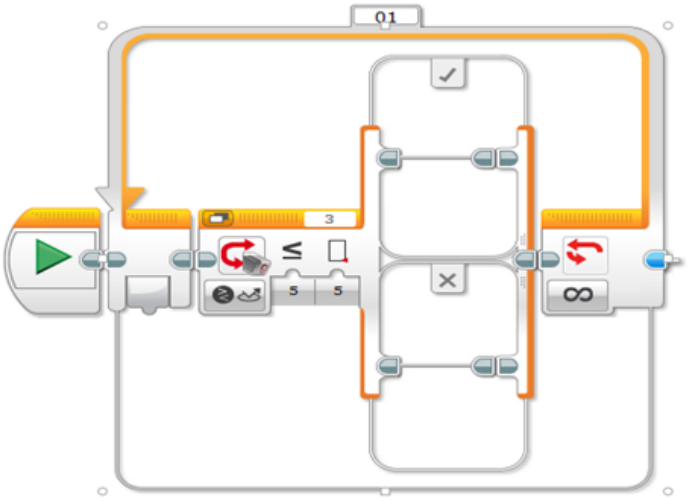

**Рис .32**

5. Переносим иконку «независимое управление моторами» из зеленой вкладки «действие» в истину и в лож. Так же проверив, какие порты (куда подключены моторы) указаны в окошках.

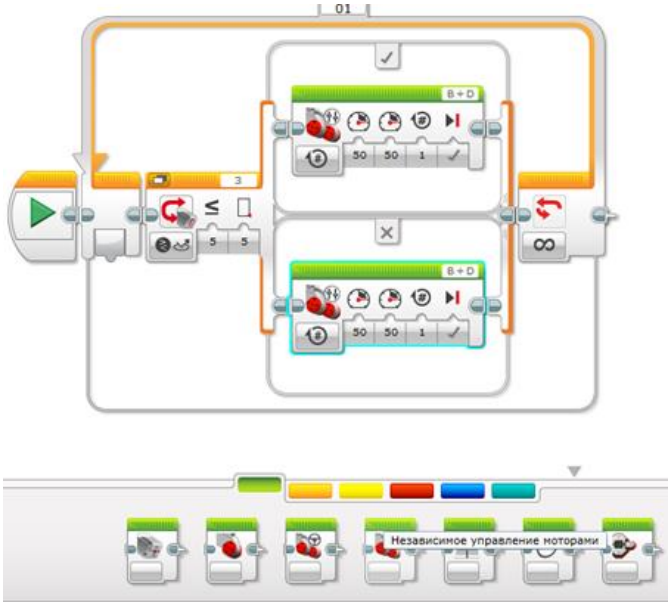


**Рис .33 Реализация алгоритма движения по черной линии с применением релейного регулятора**

### *2.3. Апробация*

Апробация разработанных материалов проводилась методом экспертных оценок. Экспертами выступали студенты Уральского государственного педагогического университета Института математики, информатики и информационных технологий, в количестве 10 человек.

Экспертам были представлены разработанные учебные задания и методические рекомендации. Целью анкет было оценить разработанные учебные задания.

**Анкета 1. Экспертная оценка разработанных методических материалов**

1. Соответствуют ли разработанные материалы выбранному разделу «Методические рекомендации по решению задач с использованием регуляторов в курсе робототехники»?
    a) полностью соответствуют;
    b) частично соответствуют;
    c) не соответствуют.
2. Насколько интересны разработанные материалы?
    a) очень интересны;
    b) интересны;
    c) не интересны.
3. Насколько эффективны разработанные материалы для внедрения в образовательный процесс?
    a) эффективны;
    b) частично эффективны;
    c) не эффективны.
4. Соответствуют ли разработанные материалы требованиям ФГОС к формированию УУД?
    a) постностью соответствуют;
    b) частично соответствуют;
    c) не соответствуют.
5. Стали бы вы использовать разработанные задания в своей педагогической деятельности?
    a) использовал(а) бы полностью;
    b) использовал(а) бы частично;
    c) не использовал(а) бы.

## Результаты апробации

Результаты апробации представлены в виде диаграмм, сформированным по результатам обработки оценок экспертов.

Диаграмма 1

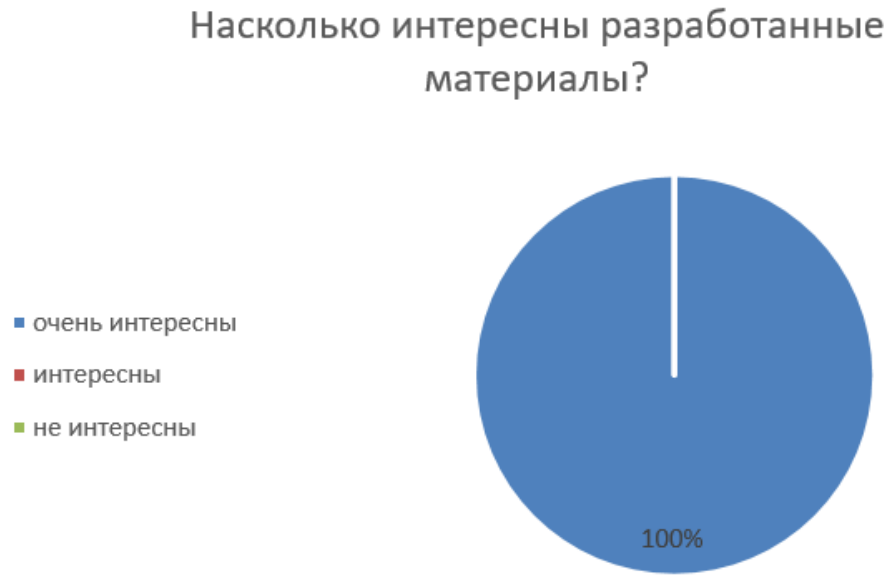


Диаграмма 2


Диаграмма 3

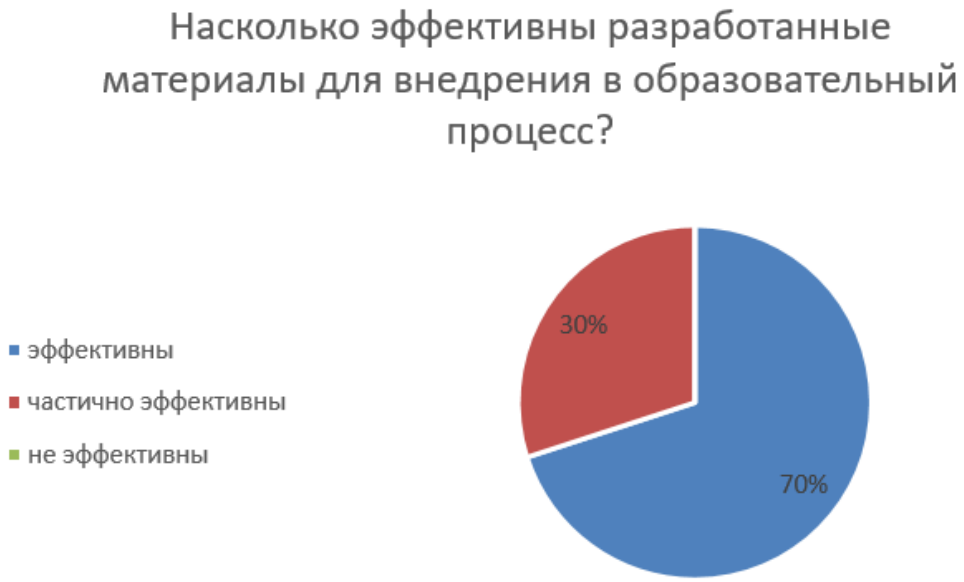


Диаграмма 4

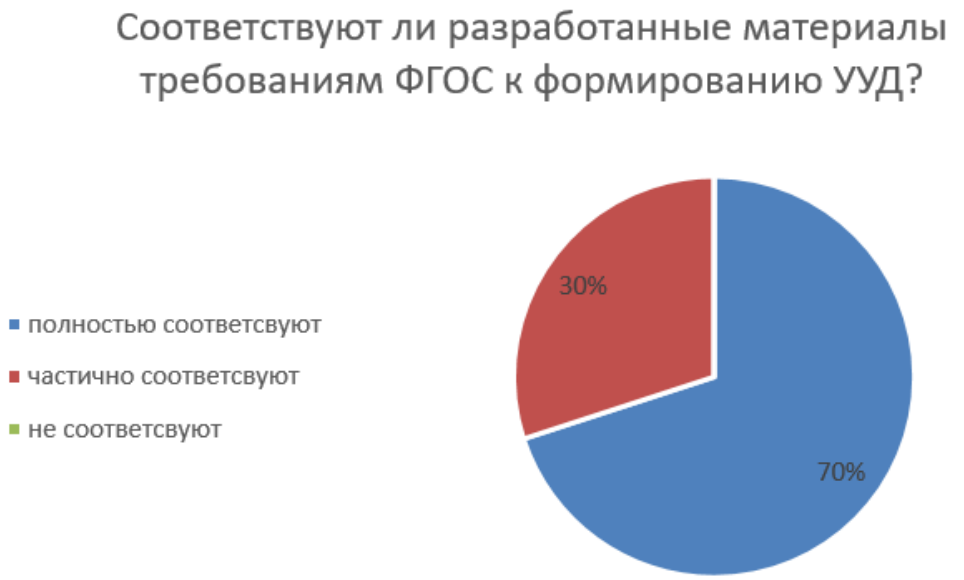


Диаграмма 5


В данной главе были рассмотрены методические рекомендации по организации урока робототехники, а также задачи с использованием регуляторов. Данное изучение позволило сделать ряд выводов.

Для проведения курса робототехники, нужен специально оборудованный кабинет, с соблюдением норм техники безопасности. А также уметь выстраивать урок, исходя из темы и правильно использовать методы организации урока.

# Заключение

С каждым днем все больше возрастает спрос на технические виды профессий, таких как, инженеры. Поэтому на сегодняшний день, преподавание робототехники в учреждениях дополнительного образования, набирает обороты. Из этого следует, что возрастает количество различных авторских рабочих программ и методик преподавания робототехники. Главными задачами данных курсов являются:

- развить навык программирования посредством управления роботом в зависимости от поставленных условий;
- развивать творческие способности и логическое мышление обучающихся;
- развивать умение выстраивать гипотезу и сопоставлять с полученным результатом;
- развивать умение применять знания из различных областей знаний;
- вовлечение детей и молодежи в научно-техническое творчество, ранняя профориентация;
- развить навыки командной работы;

В выпускной квалификационной работе было рассмотрено несколько учебных программ по курсу робототехники разных авторов и приведена сравнительная таблица этих программ.

Так же была рассмотрена тема «Регуляторы», которая включена в курс обучения по предмету «Робототехника». Выделены основные типы регуляторов, разобран принцип их работы и способы реализации.

Был проведен обзор конструкторов и сред программирования, применяемых в курсе робототехники, наиболее часто используемым конструктором является LEGO Mindstorms Education EV3 со своей средой программирования.

В практической части исследования были составлены методические рекомендации по организации урока по робототехнике.

Так же были рассмотрены несколько задач и примеры их решения, с использованием регуляторов. И проведена апробация разработанных материалов.

По результатам апробации, разработанные методические материалы соответствуют выбранному разделу дисциплины, а их практическое использование в учебном процессе будет интересным и эффективным. Исходя из этого, можно сделать вывод о том, что посредством успешного решения задач в ходе исследовательской работы, поставленная цель была достигнута.

## Библиографический список


1. LEGO Mindstorms Education EV3 – Обзор конструктора // Образование в кубе URL: https://educube.ru/news/news_detailed.php?ELEMENT_ID=1137 (дата обращения: 9.04.2017).
2. ROBOTC – язык программирования роботов // МБОУ "Лянторская Сош №4" URL: http://www.lschool4.ru/index.php?option=com_content&view=article&id=3019:robotc&catid=585:2012-10-15-15-25-06&Itemid=1008 (дата обращения: 15.04.2017).
3. VEX Robotics // Robot Geeks URL: http://robotgeeks.ru/collection/vex-iq (дата обращения: 9.04.2017).
4. Базовый набор LEGO MINDSTORMS Education EV3 // Lego Education URL URL: https://education.lego.com/ru-ru/product/mindstorms-ev3/core-set (дата обращения: 10.03.2017).
5. Ботов А. Перевод статьи «Просто о ПИД-алгоритмах» // URL: http://roboforum.ru/wiki
6. Бояркина Ю.А. Образовательная робототехника. Методическое пособие. - Тюмень: ТОГИРРО, 2013.
7. Гайсина И. Р. Развитие робототехники в школе. - Москва: Буки-Веди, 2012. - 107 с.
8. Дорф Р., Бишоп Р. Современные системы управления. Пер. с англ. Б. И. Копылова. – М.: Лаборатория Базовых Знаний. 2002. - 832 с.
9. Конструкторы программируемых роботов // ПрогХаус URL: http://proghouse.ru/article-box/26-robots (дата обращения: 17.04.2017).
10. Литвинов Ю.В., Кириленко Я.А. TRIK Studio: среда обучения программированию с применением роботов // V Всероссийская конференция «Современное технологическое обучение: от компьютера к роботу» (сборник тезисов). 2015.

11. Мордвинов Д. А., Литвинов Ю. В. Сравнение образовательных сред визуального программирования роботов. - 3-е изд. - СПб.: СПбГУ, ООО "Кибертех Лабс", 2016. - 49 с.
12. Обзор программы RoboLab для программирования NXT роботов // Муниципальное образовательное учреждение дополнительного образования "центр детского творчества" город Надым URL: http://www.cdt-nadym.edusite.ru/DswMedia/21obzorprogrammyrobolab.pdf (дата обращения: 15.04.2017).
13. Обзор робототехнического конструктора FISCHERTECHNIK ROBOTICS TXT Discovery set 28.07.2015 Динара Гагарина // Занимательная робототехника URL: http://edurobots.ru/2015/07/obzor-robototexnicheskogo-konstruktora-fischertechnik-robotics-txt/ (дата обращения: 14.04.2017).
14. Обзор робототехнических конструкторов HUNA-MRT // Занимательная робототехника URL: http://edurobots.ru/2014/07/obzor-robototexnicheskix-konstruktorov-huna-mrt/ (дата обращения: 5.04.2017).
15. Поляков К.Ю. Основы теории цифровых систем управления. - СПб: Наука, 2006.
16. Рабочая дополнительная общеобразовательная программа «робототехника» // Лангепасское городское муниципальное автономное общеобразовательное учреждение "Средняя общеобразовательная школа №2" URL: http://lgschool2.ru/Robot/jakovlev_programma_robottekhnika.pdf (дата обращения: 2.02.2017).
17. Рабочая программа курса «Образовательная робототехника и 3D моделирование» // Открытый урок URL: https://open-lesson.net/1946/. (дата обращения: 2.02.2017).

18. Робототехника-двигатель прогресса // Робототехника в образовании URL: http://Фгос-игра.рф (дата обращения: 17.03.2017).
19. Робототехнические конструкторы ROBOTIS // Занимательная робототехника URL: http://edurobots.ru/2016/05/robototexnicheskie-konstruktory-robotis/ (дата обращения: 7.04.2017).
20. Среда разработки RoboPlus для конструкторов OLLO и Bioloid // Robot Geeks URL: http://robotgeeks.ru/blogs/articles/sreda-razrabotki-roboplus-dlya-konstruktorov-ollo-i-bioloid (дата обращения: 17.04.2017).
21. ТРИК // Образовательная робототехника URL: http://robot.edu54.ru/constructors-description/222 (дата обращения: 5.04.2017).
22. Филиппов С.А. Робототехника для детей и родителей. – СПб: Наука, 2011. – 263 с.