

УДК 004.415.2, 004.588

РАЗРАБОТКА СИСТЕМЫ УПРАВЛЕНИЯ АВТОНОМНЫМ РОБОТОМ НА ПРИМЕРЕ ЗАДАЧИ СТАБИЛИЗАЦИИ В ШКОЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ РОБОТОТЕХНИКЕ

**Решетников Геннадий Павлович¹, Решетников Андрей Геннадиевич²,
Мошкова Анна Михайловна³**

¹Кандидат физико-математических наук, доцент;
ГБОУ ВО МО «Университет «Дубна»,
Институт системного анализа и управления;
141980, Московская обл., г. Дубна, ул. Университетская, 19;
e-mail: Reshetnikov@jinr.ru.

²Кандидат технических наук, доцент;
ГБОУ ВО МО «Университет «Дубна»,
Институт системного анализа и управления;
141980, Московская обл., г. Дубна, ул. Университетская, 19;
e-mail: agreshetnikov@gmail.com.

³Студент;
ГБОУ ВО МО «Университет «Дубна»,
Институт системного анализа и управления;
141980, Московская обл., г. Дубна, ул. Университетская, 19;
e-mail: vervonna@gmail.com.

Целью этой работы является разработка системы стабилизации движения робота по оптимальному курсу, на примере задачи «Движение по линии», как базис информационной технологии и образовательного процесса в области интеллектуальной робототехники. На примере системы автоматического управления (САУ) демонстрируются методы управления неустойчивым объектом и проводится сравнение различных регуляторов по установленным критериям качества. Основой для создания системы послужил классический ПИД-регулятор. В качестве объекта управления послужил робот, созданный на основе конструктора LEGO Mindstorm. Для реализации системы был использован язык Java.

Ключевые слова: робототехника, интеллектуальные системы управления, образование.

DEVELOPMENT OF A CONTROL SYSTEM FOR AN AUTONOMOUS ROBOT FOR SCHOOL EDUCATIONAL ROBOTICS

Reshetnikov Gennady¹, Reshetnikov Andrey², Moshkova Anna³

¹Candidate of Science in Physics and Mathematics, associate professor;
Dubna State University,
Institute of system analysis and management;
141980, Dubna, Moscow reg., Universitetskaya str., 19;
e-mail: Reshetnikov@jinr.ru.

²PhD, associate professor;
Dubna State University,
Institute of system analysis and management;
141980, Dubna, Moscow reg., Universitetskaya str., 19;
e-mail: agreshetnikov@gmail.com.

³Student;
Dubna State University,
Institute of system analysis and management;
141980, Dubna, Moscow reg., Universitetskaya str., 19;

e-mail: vervonna@gmail.com.

The purpose of this work was to develop a system for stabilizing the movement of the robot through an optimum path, to perform the task of "Line following," as a basis for information technology and the educational process in the field of intelligent robotics. The methods for managing an unstable object were demonstrated using the example of an automatic control system (ACS), and various regulators were compared according to the established quality criteria. The basis for creating the system was a classic PID controller. The controlled object was a robot, created on the basis of the LEGO Mindstorm construction blocks. System was developed using Java programming language.

Keywords: robotics, intelligent control systems, education.

Введение

Робототехника является одним из передовых и динамично развивающихся синергетических направлений науки и техники. В частности, робототехника – междисциплинарное направление, занимающаяся разработкой автоматизированных технических и производственных систем, является важнейшей технической основой интенсификации производства. Развитие робототехники требует одновременного развития нескольких направлений, в том числе и одно из ключевых направлений – технологий проектирования интеллектуальных систем управления (ИСУ) [1-3].

С конца 1970-х гг. для поддержки решения интеллектуальных задач, традиционно выполнявшихся людьми, на основе исследований в области искусственного интеллекта (ИИ) сформировалась новая отрасль компьютерной индустрии – разработка интеллектуальных систем. Среди множества направлений разработки интеллектуальных систем ведущее место занимают системы представление знаний и разработка систем, основанных на знаниях.

В настоящее время информационная среда наполнена различными высокотехнологичными информационными источниками, и в то же время курсы информатики и информационных технологий в школе несколько отстают от темпов развития и внедрения этих технологий в нашу жизнь. В передовых странах мира (Японии, Финляндии, и мн. др.), компании и корпорации, заинтересованные в высококвалифицированных специалистах, сотрудничают со школами и университетами с целью подготовки кадров в области информатики. Более того, обучающиеся целенаправленно проходят образовательные программы зная, что в дальнейшем эти знания и умения пригодятся в профессиональной деятельности. Важной отличительной особенностью является то, что образовательные программы основаны на самых современных информационных технологиях, а исследования учащихся являются средством развития этих технологий с дальнейшей апробацией в участвующих в таком сотрудничестве компаниях.

Зарубежное состояние развития образовательной робототехники показывает, что, например, в Японии, предмет «Программирование» включен в образовательную программу начальных классов. Программы средней школы включают подготовку на моделях автономных роботов, андроидов, манипуляторов и простых систем управления. Направления высших учебных заведений связано с развитием технологий проектирования и разработки программного обеспечения для новых роботизированных систем. Самыми популярными и представительными будут примеры Питтсбургского (*Pittsburgh*) и Стэнфордского (*Stanford*) университета. Результатами такой работы стали операционные системы для роботов (ROS) и всевозможные запатентованные алгоритмы и программные комплексы обработки информации, представляющие коммерческую привлекательность.

На рисунке 1 представлены учебно-исследовательские роботы типа «мобильный манипулятор» этих университетов (*HERB: Home Exploring Robotic Butler* [4] и *STAIR: Stanford Artificial Intelligence Robot* [5] и мн. др.), являющиеся пособиями для создания новых информационных технологий и программных продуктов.



Рис. 1. Роботы ведущих университетов

Очевидно, что для создания образовательных лабораторных роботов и программ передовые университеты не используют унифицированные конструкторы и комплектующие. Для этих целей приобретается самое современное оборудование, например, видео и тепло камеры, лазерные радары, манипуляторы, двигатели и платы, разрабатывается свое программное обеспечение. Изобилие самых современных сенсоров, вычислительных машин и исполнительных механизмов, позволяет создавать все новые и более совершенные алгоритмы обучения, адаптацию и самоорганизацию интеллектуальных систем. Каждая из таких технологий, закладываемая в конечное изделие, является объектом интеллектуальной собственности и подлежит тщательной авторской защите.

Результаты применения таких работ воплотились в появление специализированных программных инструментариев для проектирования систем управления. В свою очередь создание открытых библиотек на базах этих инструментариев позволило привлекать огромный класс специалистов и исследователей, для программно-аппаратного интерфейса, что дает толчок развитию архитектуры технологий, инструментариев и продуктов коммерческой собственности.

В средних и старших классах наступает этап подведения промежуточных профессиональных итогов, когда человек задумывается о перспективах и отвечает на многие важные для себя вопросы о будущем в профессии и желательном направлении дальнейшего развития. В ближайшее время навыки программирования будут необходимы практически для любой деятельности в сфере информационных технологий (ИТ), а одним из наиболее интереснейших способов получить такие навыки является образовательная робототехника, где на занятиях ученики работают, как с механическими, так и с электрическими устройствами, заставляя их выполнять написанные ими же программы.

Одним из способов подготовки людей к будущей профессиональной деятельности в сфере высоких технологий и уменьшения колоссального отставания в сфере информационных технологий – является развитие образовательной интеллектуальной робототехники.

В интернете представлено множество открытых примеров образовательных роботов [6], некоторые из них представлены на рисунке 2:



Рис. 2. Примеры образовательных роботов

В качестве среды программирования выступают *Arduino (C++)* для программирования микроконтроллеров, *ROS (C++, Python)* для программирования автономного поведения робота (навигация, распознавание, частично дистанционное управление), *Matlab* для математического моделирования динамики системы (движение робота). Отметим, что эти среды программирования включают в себя возможность подключения модуля для *LEGO Mindstorms NXT* (что показывает возможность масштабирования образовательного процесса и на конструкторы *LEGO*) [2, 3] для младшего и среднего возраста.

В образовательном процессе ученики знакомятся с методами и средствами разработки и реализации программных продуктов, учатся работать с подключаемыми библиотеками, знакомятся с типами данных, массивами, структурами, классами, объектами, формируют начальные навыки программирования.

Образовательный процесс включает в себя лекционные занятия, семинары и лабораторные практикумы, последовательно раскрывающих возможности программных инструментариев вычислительных систем. Компьютер рассматривается как инструмент в должной степени детализации архитектуры и операционных систем.

Так, целью образовательного процесса со школьниками стало:

- профессиональная ориентация и подготовка учеников;
- формирование и развитие навыков работы с языками программирования высокого уровня, работы с программными инструментариями моделирования и дизайна;
- стимулирование самоорганизации и самообразования в области информационных технологий;
- получение навыков для эффективного обучения в университете и ведение научной деятельности;
- ознакомление учеников с возможностями объектно-ориентированных языков программирования и развитие навыка программирования на языках высокого уровня (*C++, Java, C#*);
- развитие понимания базовых принципов теории автоматического управления и теории информации, электротехники и мехатроники на примере робототехники;
- ознакомление учащихся с современными методами и средствами проектирования программного обеспечения для роботов в том числе интеллектуальных систем;
- стимулирование интереса к технике, конструированию, программированию, новым информационным технологиям;
- формирование навыков подготовки презентационного материала, выступления на конференциях, написание публикаций (на русском и английском языке), участие в всероссийских и международных конкурсах и мероприятиях;
- развитие навыков коллективного труда и работы в команде, работы на результат и персональной ответственности;
- стимулирование саморазвития и самоорганизации учащихся.

При этом постепенное освоение материала наслаивается на более серьезные задачи из области робототехники:

- дистанционно управляемый робот;
- робот дистанционного присутствия;
- робот-манипулятор;
- обработка изображения и распознавание образов;
- мини-робот с манипулятором;
- автономный робот с навигационной системой;
- взаимодействие группы роботов.

В качестве базиса представленного образовательного процесса выступает мини-робот дистанционного управления.

Целью этой работы является разработка системы стабилизации движения робота по оптимальному курсу, на примере задачи «Движение по линии», как базис информационной технологии и образовательного процесса в области интеллектуальной робототехники. На примере системы автоматического

управления (САУ) демонстрируются методы управления неустойчивым объектом и проводится сравнение различных регуляторов по установленным критериям качества.

Система автоматического управления автономным роботом

Одной из центральных задач в робототехнике является задача навигации робота, то есть определение местоположения, планирование маршрута и построения карты. Часто в качестве аппаратного решения такой задачи используют навигационные лазерные дальномеры, стереопары и др. датчики. Программный уровень реализации навигационной системы называется *SLAM* [7] алгоритм или Метод одновременной локализации и построения карты (*SLAM* от англ. *Simultaneous Localization and Mapping*) – метод, используемый в мобильных автономных средствах для построения карты в неизвестном пространстве или для обновления карты в заранее известном пространстве с одновременным контролем текущего местоположения и пройденного пути.

На рисунке 3 представлены «сцены мира» получаемых с применением данных устройств, алгоритмов и технологий.

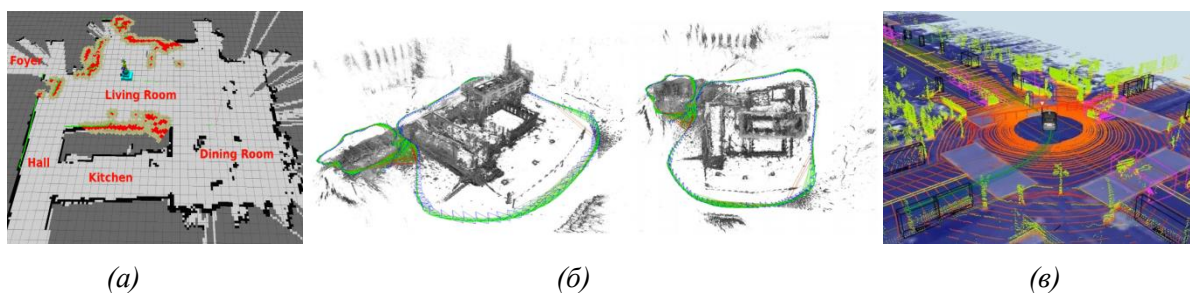


Рис. 3. Сцены мира навигационных систем. А – сцена мира в помещении с применением LIDAR A1. Б – сцена мира, построенная на базе автономного летательного беспилотника [8]. В – сцена мира автотопилота

В частности, на нижнем уровне управления, для выработки управляющих воздействий обычно применяются так называемые ПИД-регуляторы, о которых речь пойдет далее.

ПИД-регулятор в контуре обратной связи

Объектом исследований в данной работе, стал классический пропорционально-интегрально-дифференциальный (ПИД) регулятор в цепи управления с обратной связью. На рисунке 4 представлена схема традиционной системы управления на основе ПИД-регулятора. В качестве объекта управления (ОУ) выступает робот *Educator Vehicle*.

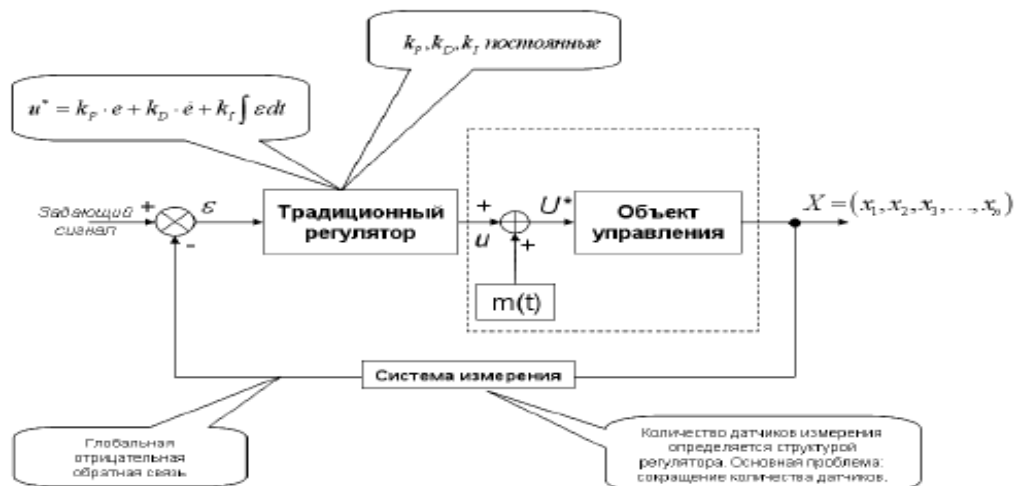


Рис. 4. Схема ПИД-регулятора

Традиционная система управления представляет собой совокупность одного или нескольких ОУ и управляющей системы. В общем случае, система управления состоит из звена регулирования, ОУ, а также измерительной системы (ИС) в цепи обратной связи. Для обеспечения заданных динамических показателей в системах управления находят применение различные типы регуляторов. Наиболее широко распространены пропорционально-интегрально-дифференциальные (ПИД) регуляторы.

Интегральная составляющая регулятора позволяет устранить в системе статическую ошибку, а дифференциальная составляющая – улучшить динамические показатели, форсируя переходный процесс.

ПИД-регулятор был изобретен еще в 1910 году, позже, в 1942 г., Зиглер и Никольс разработали методику его настройки, а после появления микропроцессоров в 80-х годах развитие ПИД-регуляторов происходит нарастающими темпами. Общее количество публикаций по ПИД-регуляторам за 9 лет с 1973 по 1982 г. составило 14 шт., с 1983 по 1992 г. – 111 шт., а за период с 1998 по 2002 год (за 4 года) – 225 шт. На одном только семинаре *IFAC (International Federation of Automatic Control)* в 2000 г. было представлено около 90 докладов, посвященных ПИД-регуляторам. Количество патентов по этой теме, содержащихся в патентной базе данных в январе 2011 г. составило 284 шт.

ПИД-регулятор относится к наиболее распространенному типу регуляторов. Около 90...95% регуляторов, находящихся в настоящее время в эксплуатации, используют ПИД алгоритм. Причиной столь высокой популярности является простота построения и промышленного использования, ясность функционирования, пригодность для решения большинства практических задач и низкая стоимость. Среди ПИД-регуляторов 64% занимают одноконтурные регуляторы и 36% – многоконтурные. Контроллеры с обратной связью охватывают 85% всех приложений, контроллеры с прямой связью – 6%, контроллеры, соединенные каскадно – 9%.

Пропорциональный регулятор является основным, где задание скорости прямо пропорционально ошибке. Если используется только пропорциональный регулятор, то в системе всегда будет ошибка. Низкие значения пропорционального коэффициента регулятора приводят к «вялости» системы, а слишком высокие приводят к неустойчивости и колебаниям в системе.

Интегральный регулятор используется для исключения ошибки. Скорость будет расти до момента исключения ошибки (или уменьшаться при отрицательной ошибке). Малые значения интегральной составляющей значительно влияют на работу регулятора в целом. Если установлено слишком высокое значение – система промахнется, и будет работать с перерегулированием.

Дифференциальный регулятор, оценивая скорость изменения ошибки, используется для увеличения быстродействия системы. Он и повышает быстродействие регулятора в целом. Однако с повышением быстродействия регулятора также увеличивается и его перерегулирование, что приводит к неустойчивости системы. В большинстве случаев, дифференциальная составляющая устанавливается нулевой или близкой к некоторому очень низкому значению для предотвращения этого. Однако она может быть очень полезной в системах позиционирования. На рисунке 5 показана зависимость системы от подобранного коэффициента.



Рис. 5. Графики переходных процессов для ПИД-регулятора

Основной проблемой управления является настройка ПИД-регулятора, то есть подбор коэффициентов усиления каждой компоненты. В общем случае необходимо найти коэффициенты ПИД-регулятора, при которых обеспечивается желаемый характер движения робота. Для определения коэффициентов аналитическими методами, необходимы первоначальные знания о системе управления и характере поведения ОУ [9, 10], т.е. корректное определение коэффициентов ПИД-регулятора возможно с привлечением эксперта.

Для достижения цели управления используется ПИД-регулятор, который вырабатывает закон управления вида:

$$u(t) = k_p e + k_d \dot{e} + k_i \int e dt, \quad (1)$$

где $u(t)$ – управляющее воздействие, а k_p, k_d, k_i – постоянные коэффициенты, определяющие величины пропорциональной, дифференциальной и интегральной компонент в управляющем сигнале. Зависимость (1.1) называется изодромным управлением с введением первой производной. Такое управление включает в себе высокую точность интегрального управления (k_i), большое быстродействие пропорционального управления (k_p) и упреждение ошибки, благодаря производной составляющей (k_d).

Основной проблемой управления является настройка ПИД-регулятора, то есть подбор коэффициентов усиления k_p, k_d, k_i каждой компоненты. В рамках данной работы рассмотрена возможность применения интеллектуальных технологий управления для настройки ПИД-регулятора – подбор коэффициентов усиления и формирования базы знаний нечеткого контроллера (НР) с использованием генетических алгоритмов.

В соответствии с законами автоматического регулирования и теории управления соотношение между сигналом на выходе динамического ОУ и задающим сигналом определяется следующим уравнением:

$$X(i\omega) = \frac{1}{\underbrace{1 + \Phi_1(i\omega)\Phi_2(i\omega)}_{\text{Устойчивость}}} \bullet \underbrace{\Phi_1(i\omega)\Phi_2(i\omega) \bullet Y(i\omega)}_{\text{Управляемость}}, \quad (2)$$

где $\Phi_1(i\omega)$ – передаточная функция регулятора, $\Phi_2(i\omega)$ – передаточная функция ОУ, Y – задающий сигнал, X – выходной сигнал ОУ. Нетрудно видеть, что эта формула, описывающая отношение вход/выход в структуре САУ, требует исследования одновременно двух критериев качества управления: устойчивость и управляемость [2]. Устойчивость – это способность системы возвращаться в состояние равновесия, при устранении причин, вызвавших отклонение от этого состояния равновесия [3]. Управляемость – возможность перевести систему из одного состояния в другое [3]. Следовательно, принцип ГООС содержит в себе два указанных выше критерия качества управления. Робастность рассматривается как третий независимый критерий качества управления, проектирование которого осуществляется после установления уровней устойчивости и управляемости. Под робастностью понимается способность системы противостоять неожиданным внешним воздействиям, при которых система ещё способна остаться в устойчивом состоянии [2].

Оптимальное управление – это такое управление, при котором принятый критерий оценки качества работы системы (критерий оптимальности) достигает экстремального значения [3].

Критерий качества системы автоматического управления основан на совокупности показателей, характеризующих качество работы САУ. К прямым показателям качества САУ относятся: время переходного процесса, время установления, глубина перерегулирования, число и частота колебаний регулируемой величины в течение времени переходного процесса, установившаяся ошибка системы (рис. 8). Прямые критерии качества определяются по кривой переходного процесса. Время переходного процесса – время, в течение которого отклонение регулируемой величины относительно установившегося значения становится (и в дальнейшем остается) по абсолютному значению меньше некоторой наперед заданной величины (обычно применяется 5%) [3]. Время установления – время, за которое выходная величина динамической системы в первый раз достигает своего установившегося значения при переходном процессе, возникающем вследствие подачи на вход системы возмущения (рис. 6).

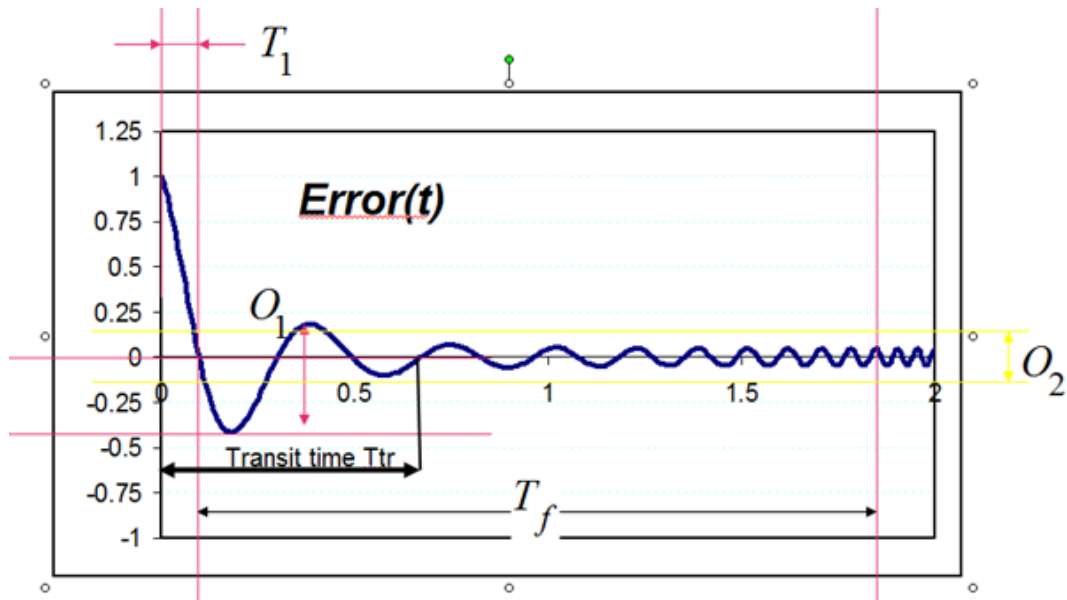


Рис. 6. Кривая переходного процесса T_1 – время установления, T_{tr} – время переходного процесса

Быстродействие переходного процесса – это время перехода системы из некоего начального состояния в требуемое конечное [3]. Быстродействие системы определяется следующим выражением:

$$T_m = 1 - \frac{T_1}{T_f}; T_f (const.) > T_1. \quad (3)$$

Еще один критерий качества переходного процесса – это максимальное отклонение регулируемой величины относительно ее установившегося значения, выраженное в процентах или в масштабировании на единицу, такой критерий называется перерегулированием и определяется как:

$$O_m = 1 - \frac{O_1}{O_m}; O_m (const.) > O_1. \quad (4)$$

Обычно, для стабильного управления принимается следующий критерий:

$$Q_m = 1 - \frac{Q_2}{O_m}; O_m (const.) > Q_2. \quad (5)$$

Показатель производительности включает в себя совокупность всех вышеперечисленных критериев. Производительность – показатель эффективного управления.

$$\text{Производительность} = \omega_1 \cdot P[T_m] + \omega_2 \cdot P[O_m] + \omega_3 \cdot P[Q_m]. \quad (6)$$

Возможны комплексные оценки качества управления, например время стабилизации (Т) принимается за основной критерий эффективности, а в качестве дополнительного критерия (таблица 1.1) используют так называемый интеграл квадратичной ошибки (ISE), а совместное использование этих критериев, есть целевая функция (ЦФ) для параметров ПИД-контроллера:

$$\text{ЦФ} = \alpha T + (1 - \alpha) \int_0^T t |e(t)| dt. \quad (7)$$

Таблица 1.1. Дополнительные критерии качества

Критерии	Определение
ISE	$ISE = \int_0^T e^2(t) dt$
IAE	$IAE = \int_0^T e(t) dt$
$ITAE$	$ITAE = \int_0^T t e(t) dt$
$ITSE$	$ITSE = \int_0^T te^2(t) dt$

Большинство систем управления, функционирующих в настоящий момент, построены опираясь на приведенные выше критерии качества. В тривиальных задачах управления традиционная система вполне пригодна. Но такая система не всегда справляется с задачей управления под влиянием многих факторов таких, как резкое изменение условий внешней среды, наличие времени задержки в каналах измерения и управления, резкое изменение параметров структуры самого объекта управления или резкое изменение задающего сигнала. Особенно важный фактор – это резкое изменение условий внешней среды и изменения параметров структуры, при котором данная традиционная система не обеспечивает робастность, и редко достигает цели управления, и заданного уровня качества.

Для решения данной проблемы существуют различные интеллектуальные системы управления. Большинство интеллектуальных систем управления обеспечивают частичную или полную робастность путем изменения коэффициентов ПИД-регулятора во времени. Классическая теория проектирования систем управления не имеет алгоритмов оптимального изменения коэффициентов во времени, поэтому привлекаются новые типы вычислений – интеллектуальные вычисления.

В общем виде к интеллектуальным вычислениям относятся:

- Мягкие вычисления и нечеткая логика.
- Искусственные и нечеткие нейронные сети.
- Алгоритмы оптимизации и генетические алгоритмы.

Однако, основное внимание данной работы посвящено именно созданию базиса образовательного процесса с применением инновационных информационных технологий разрабатываемых в лаборатории Интеллектуальных систем управления университета Дубна.

В соответствии с поставленными задачами был спроектирован, разработан и запрограммирован робот на базе конструктора *LEGO Mindstorm*.

Модель и макет робота

При проектировании сложных систем и их подсистем возникают задачи, требующие оценки количественных и качественных закономерностей процессов функционирования таких систем, проведения структурного алгоритмического и параметрического их синтеза [11-12]. Но сложные системы относятся к классу больших систем, этапы проектирования, внедрения, эксплуатации и эволюции которых в настоящее время невозможны без использования различных видов моделирования. Ресурсы современной информационно-вычислительной техники дают возможность ставить и решать математические задачи такой сложности, которые в недавнем прошлом казались нереализуемыми, например моделирование больших систем.

Ограниченность возможностей экспериментального исследования больших систем делает актуальной разработку методики их моделирования, которая позволила бы в соответствующей форме представить процессы функционирования систем, описание протекания этих процессов с помощью математических моделей, получение результатов экспериментов с моделями по оценке характеристики исследуемых объектов. Причем на разных этапах создания и использования перечисленных систем для

всего многообразия входящих в них подсистем, применение метода моделирования преследует конкретные цели, а эффективность метода зависит от того, насколько грамотно разработчик использует возможности моделирования.

Выбор метода моделирования и необходимая детализация моделей существенно зависят от этапа разработки сложной системы [13]. На этапах обследования объекта управления, например промышленного предприятия, и разработки технического задания на проектирование автоматизированной системы управления модели в основном носят описательный характер и преследуют цель наиболее полно представить в компактной форме информацию об объекте, необходимую разработчику системы.

С помощью непосредственного интегрирования можно решить множество важных задач о движении частицы (или точки, имеющей определенную массу), если известны силы, действующие на частицу [14]. Движение частицы по прямой линии (ось x) описывается ее положением: $x = f(t)$, т.е. функцией, дающей ее координату x в момент времени t . Скорость частицы определяется равенством:

$$v(t) = f'(t), \text{ т. е. } v = dx/dt.$$

Ускорение тела $a(t)$ равно $a(t) = v'(t) = x''(t)$ или, в обозначениях Лейбница:

$$a = \frac{dv}{dt} = \frac{d^2x}{dt^2}$$

Второй закон Ньютона (закон движения тела под воздействием силы) гласит, что если сила $F(t)$ действующая на частицу, направлена вдоль ее линии движения, то:

$$ma(t) = F(t); \text{ т. е. } F = ma,$$

где m – масса частицы.

Если сила F известна, то уравнение $x''(t) = F(t)/m$ можно проинтегрировать дважды. Этим будет найдена функция, дающая положение $x(t)$. Эта функция будет содержать две константы, появляющиеся после интегрирования. Эти две произвольные постоянные часто определяются по начальному положению $x_0 = x(0)$ и начальной скорости $v_0 = v(0)$ частицы.

Предположим теперь, что сила F , а значит и ускорение $a = F/m$, являются постоянными. Начнем с уравнения:

$$a = \frac{dv}{dt} \quad (a - \text{константа}).$$

Проинтегрировав обе части, получим:

$$v(t) = \int a dt = at + C_1.$$

Мы знаем, что $v = v_0$, когда $t = 0$, и подстановка этой информации в предыдущее уравнение дает $C_1 = v_0$. Так что

$$v(t) = \frac{dx}{dt} = at + v_0. \quad (8)$$

Второе интегрирование дает:

$$x(t) = \int v dt = \int (at + v_0) dt = \frac{1}{2} at^2 + v_0 t + C_2,$$

а подстановка $t = 0$, $x = x_0$ дает $C_2 = x_0$. Поэтому:

$$x(t) = \frac{1}{2} at^2 + v_0 t + x_0. \quad (9)$$

Таким образом, используя уравнение (8), мы можем найти скорость, а с помощью уравнения (9) положение частицы в любой момент времени t , если известно ее постоянное ускорение a , ее начальная скорость v_0 и ее начальное положение x_0 [15].

Для демонстрации работы ПИД-контроллера мы создали робота, выполняющего одну из наиболее распространенных задач в робототехнике: движения робота по линии. Ее распространенность связана с ее простотой и с огромным числом способов реализовать ее решение.

Макет автономного робота

Робот *Educator Vehicle* (рис. 7) – стандартная модель для базового образовательного набора *LEGO Mindstorms EV3 Education*. Используя робота можно решать общепринятые задачи образовательной робототехники: движение по линии, избегание препятствий, движения вдоль стены и другие. Для этих целей на робота можно установить датчики: цвета, расстояния, касания и другие.

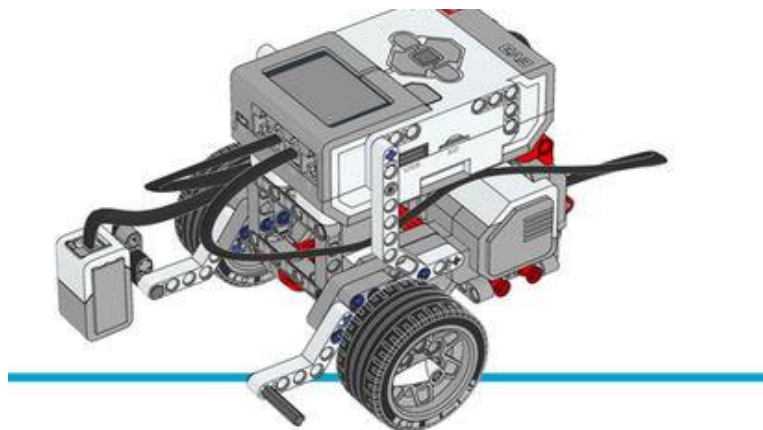


Рис. 7. 3D-модель робота

Для демонстрации работы ПИД-контроллера мы создали робота (рис.8), выполняющего одну из наиболее распространенных задач в робототехнике: движения робота по линии. Ее распространенность связана с ее простотой и с огромным числом способов реализовать ее решение.

Для создания робота использовался базовый набор *LEGO MINDSTORMS Education EV3*. Вместе с входящим в комплект поставки программным обеспечением этот набор образует одно из самых мощных и универсальных интерактивных образовательных решений для основной школы. Сердцем набора является микрокомпьютер *EV3*, управляющий моторами и датчиками. Он также обеспечивает связь микрокомпьютера *EV3* и персонального компьютера или, а также способен регистрировать экспериментальные данные. Микрокомпьютер *EV3* также имеет программный интерфейс, позволяющий создавать программы и настраивать регистрации данных непосредственно на микрокомпьютере *EV3*. Микрокомпьютер совместим с мобильными устройствами и питается батареями типа АА или аккумуляторной батареей *EV3*.

Также в состав набора входят 3 сервомотора различной мощности (2 больших и 1 средний), 5 датчиков (гироскопический и ультразвуковой датчики, датчик света/цвета и два датчика касания), перезаряжаемая аккумуляторная батарея и соединительные провода [16].



Рис. 8. Макет робота

Для создания были использованы детали набора:

- Программируемый блок *EV3*.
- Два больших сервомотора.
- Два датчика освещенности
- Соединительные кабеля.
- Детали *LEGO Technic*.

Информационная технология образовательного процесса.

В качестве информационной технологии (ИТ) образовательного процесса был выбран открытый программный продукт зарубежных разработчиков *leJOS* [17]. Это было сделано не случайно, обучающий конструкторов *LEGO Mindstorm EV3* с наличием встроенного графического интерфейса, не может в полной мере охватить рабочие программы учебных учреждений по информатике, математики, физики и т.д. *leJOS* – это виртуальная *Java*-машина, которая в 2013 году была адаптирована для работы с системой *LEGO Mindstorms EV3*.

Преимущества *leJOS* перед *LEGO*:

- Благодаря использованию *leJOS* мы получаем возможность использовать объектно-ориентированный язык *Java*, со всеми его преимуществами, в том числе массивы (многомерные массивы тоже), рекурсию, синхронизацию, исключения, большинство *Java*-классов из библиотек *java.lang*, *java.util*, *java.io* и т.д.
- Возможность разрабатывать программы при помощи специального плагина в продвинутой среде разработки *Eclipse* с возможностью полноценной отладки кода.
- Возможность создавать приложения, работающие не только в модуле *EV3*, но и приложения для смартфонов, планшетов и компьютеров для удалённого взаимодействия с роботом, в том числе приложения для модулей *EV3*, которые смогут удалённо взаимодействовать с другими модулями *EV3* и *NXT*.
- При помощи веб-камеры, подключенной к модулю *EV3*, можно захватывать изображения или видео и отображать их на дисплее или удалённо передавать их на компьютер или мобильное устройство. Камеру можно использовать, например, вместо датчика цвета.
- Наличие утилит для постройки карт помещения и передвижения по ним с огибанием препятствий (*EV3 Map Command* и *EV3 MCL Command*) и удобным центром управления с просмотром в реальном времени, что отображается на дисплее *EV3 (EV3 Control Center)*.
- Поддержка некоторых датчиков *NXT*, *RCX*, *HiTechnic*, *Mindsensors*, *Dexter* и *Cruizcore Gyro*. Из моторов поддерживаются моторы *NXT*, *RCX*, *PF* и *Tetrix*. Также поддерживаются подключенные к *USB-порту GPS-приёмники* (с компасом и без него).
- В будущей версии (после **0.9.0-beta**) намечен выход классов *Chassis* (шасси), позволяющих с лёгкостью управлять колёсным роботом любой конструкции, в том числе с *Omni-колёсами* или с рулевым управлением, просто указывая направление движения робота [18].

LeJOS распространяется бесплатно, ее последнюю версию можно скачать с официального сайта [19]. Для установки *leJOS* на блок управления *LEGO* сначала необходимо установить дистрибутив на компьютер, и затем при помощи встроенной утилиты создать загрузочную *SD* карту (рис. 9).

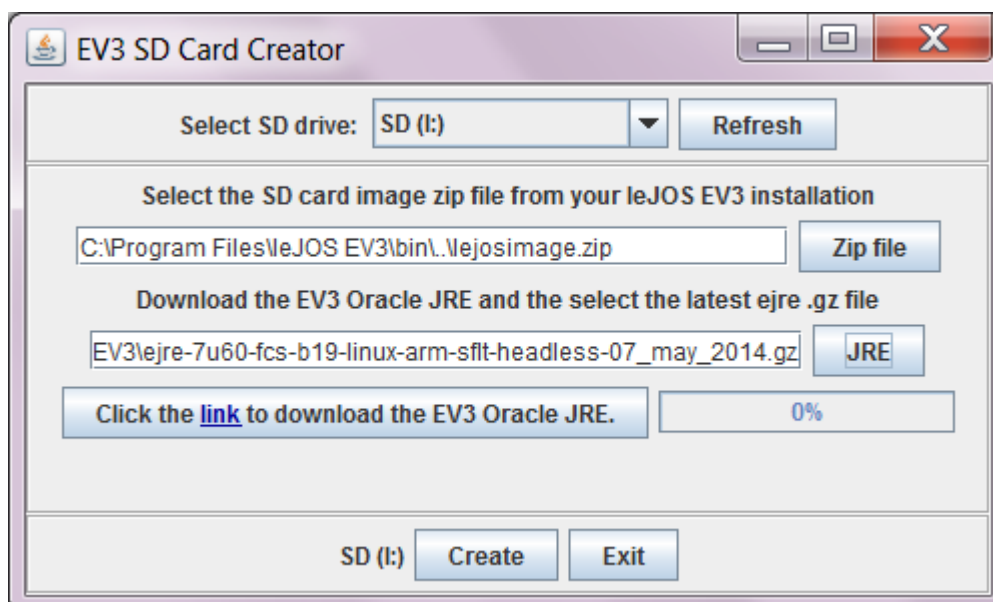


Рис. 9. Создание загрузочной SD-карты

Полученная карта вставляется в блок управления и загружается. Стандартное ПО *LEGO* при этом не удаляется, после извлечения загрузочной карты leJOS и повторного включения устройства снова загрузится *LEGO Mindstorm*.

Для разработки на *Java* для *LEGO Mindstorm EV3* необходимо установить плагин *leJOS* для студии *Eclipse*.

Описание кода (рис. 10):

```
while(!Button.ESCAPE.isDown())
{
    count++;
    LCD.clear();
    s1.fetchSample(sample, 0);
    s2.fetchSample(sample2, 0);

    alf = (float)((c1-sample[0])+(c2+sample2[0]))/2;
    alf = alf * 100;
    alf_int +=alf;

    if(alf>0)
    {
        uval=100-(kp*alf+(alf-alf_old)*kd+alf_int*ki);
        mRight.setSpeed(100);
        mLeft.setSpeed(uval);
    }
    else
    {
        uval=100-(kp*alf+(alf-alf_old)*kd+alf_int*ki);
        mLeft.setSpeed(100);
        mRight.setSpeed(uval);
    }

    mRight.forward();
    mLeft.forward();
    alf_old=alf;
}
```

Рис. 10. Код, вычисляющий угол поворота и скорость моторов

Подбор коэффициентов

Для более точной и стабильной работы ПИД-контроллера необходимо подобрать наиболее оптимальные коэффициенты. Подбор коэффициентов осуществлялся при помощи метода рекурсивного спуска.

Подбор пропорционального коэффициента (рис. 11-12):

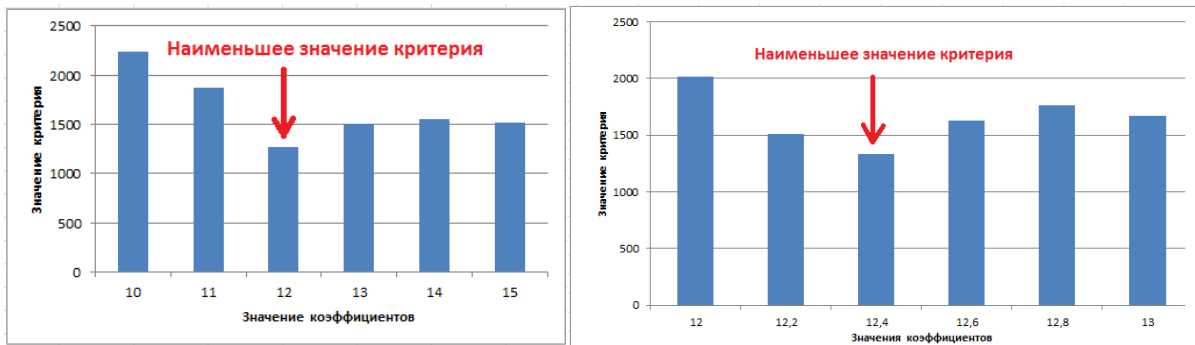


Рис. 11. Интеграл ошибки управления при различных k_p

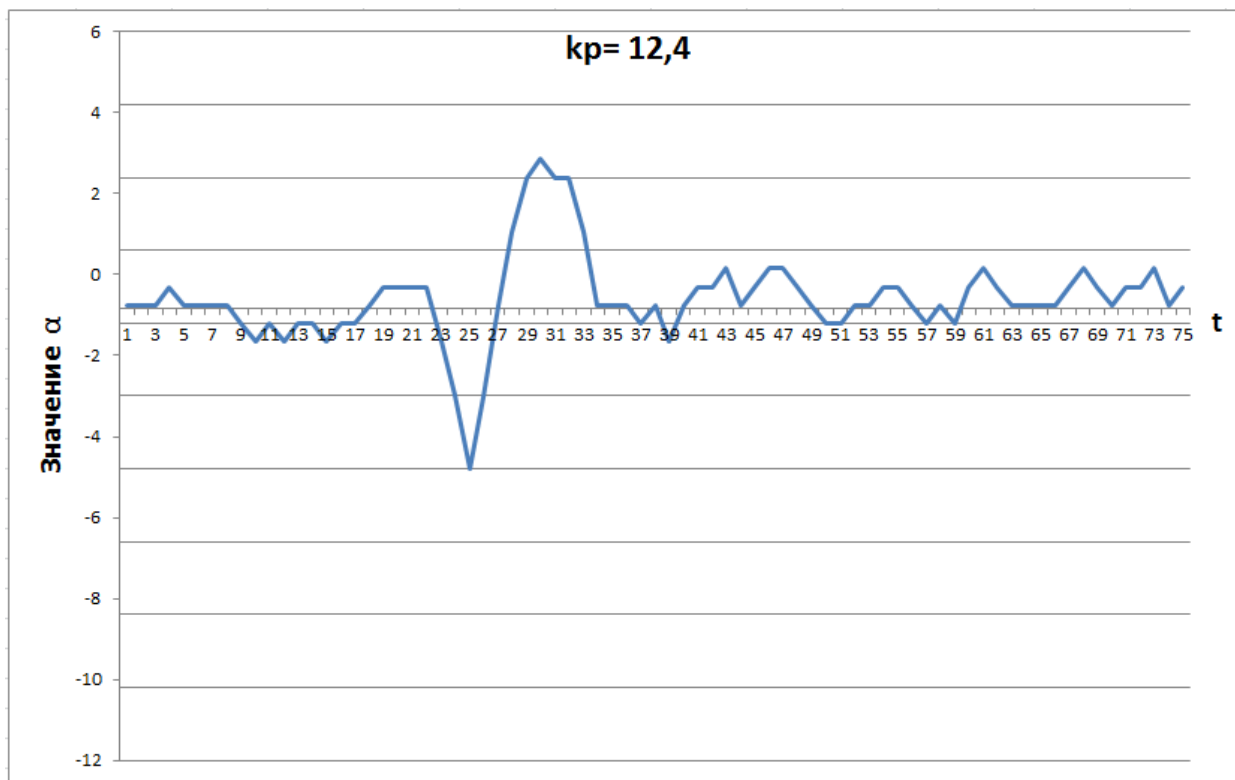


Рис. 12. Угол отклонения от целевого значения при $k_p = 12,4$

Подбор дифференциального коэффициента (рис. 13-14):

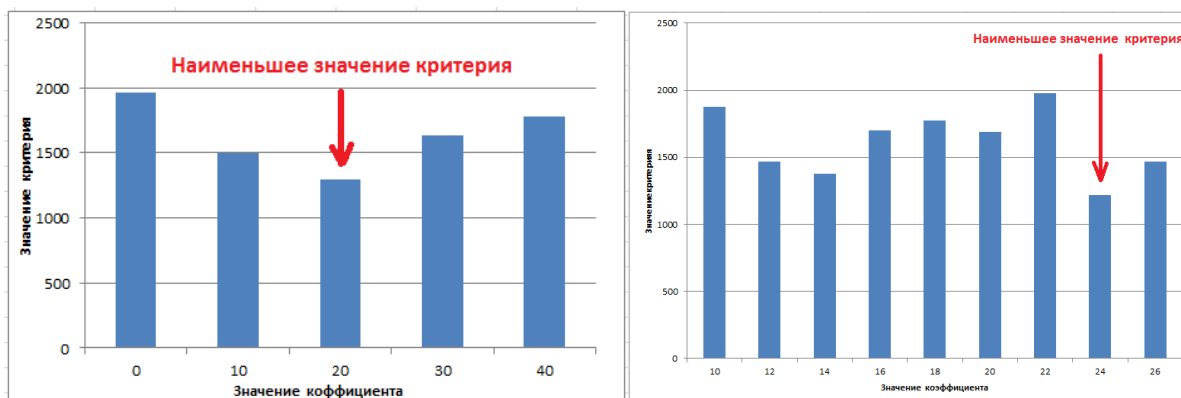


Рис. 13. Интеграл ошибки управления при различных k_d

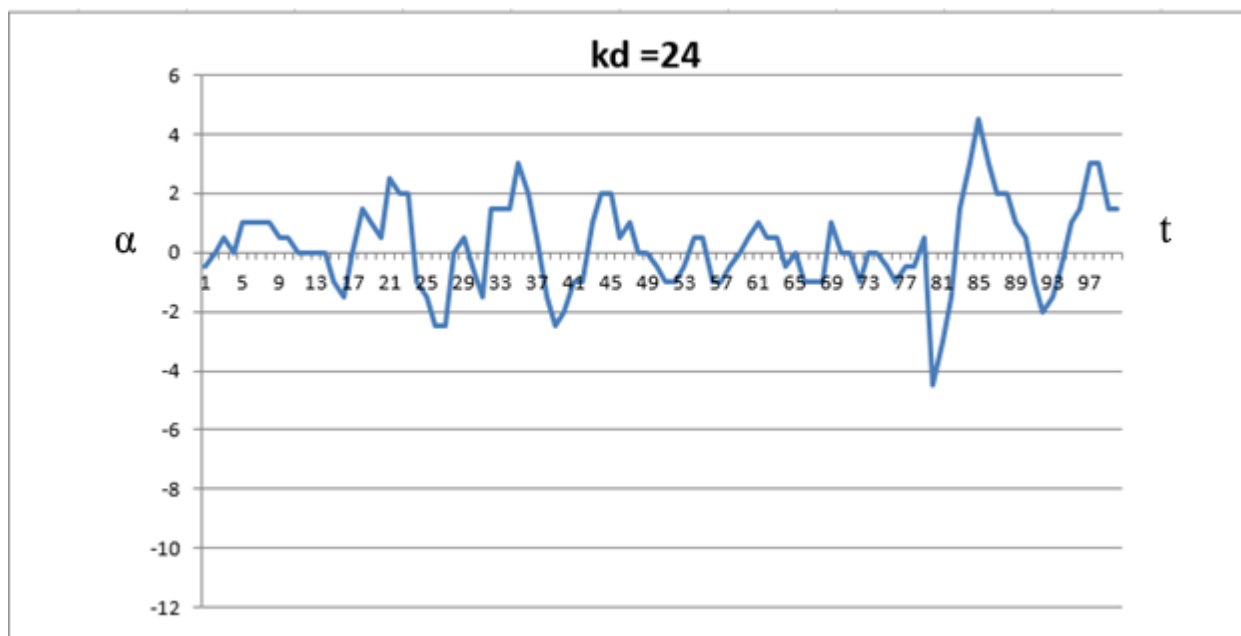


Рис. 14. Угол отклонения от целевого значения при $kd = 24$

Подбор интегрирующего коэффициента (рис.15):

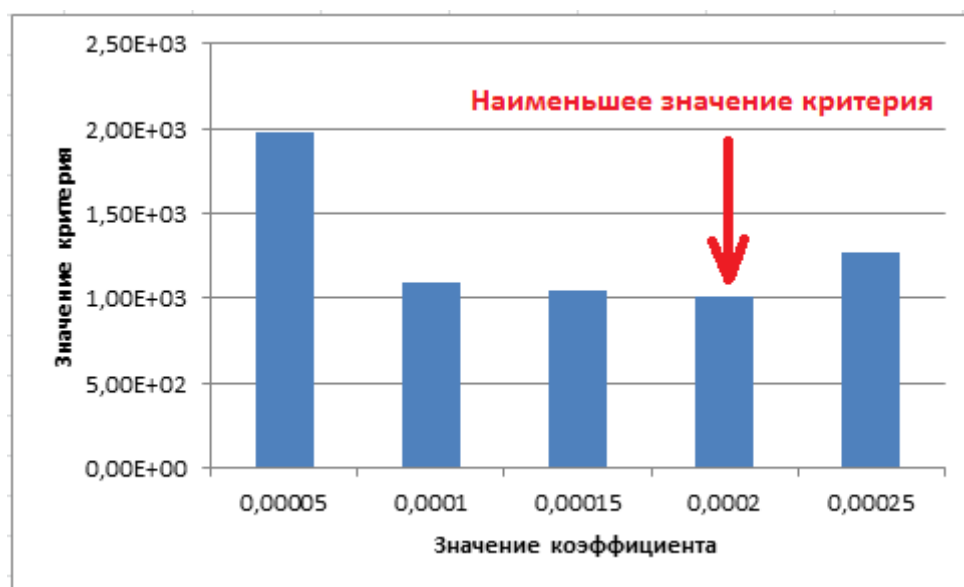


Рис. 15. Интеграл ошибки управления при различных ki

Проведенные тесты показывают, что САУ работает устойчиво, позволяет управлять объектом при движении с умеренной скоростью ~ 20 км/ч.

САУ работает так, что средний угол отклонения не превышает одного градуса.

Внедрение в учебную деятельность

Данная работа показывает новые возможности для обучения нового поколения будущих специалистов, позволяя вовлечь школьников в процесс разработки и познакомить их с основами робототехники и программирования.

Образовательные наборы *LEGO Mindstorms EV3* являются международным стандартом для образовательной робототехники, так как ни один другой набор не обладает таким уровнем стандартизации,

простоты использования и глубины проработки. Выпущенное в 2013-м году третье поколение образовательного робототехнического набора от *LEGO, EV3* обладает поистине необъятной широтой возможностей, заложенных в программное обеспечение и аппаратную составляющую, а совместимость с любыми другими наборами *LEGO* даже 40-летней давности дает очевидную возможность использовать любые детали для строительства конструкций.

Использование конструктора *LEGO EV3* позволяет создать уникальную образовательную среду, которая способствует развитию инженерного, конструкторского мышления. В процессе работы с *LEGO EV3* обучающиеся приобретают опыт решения как типовых, так и нешаблонных задач по конструированию, программированию, сбору данных. Кроме того, работа в команде способствует формированию умения взаимодействовать с соучениками, формулировать, анализировать, критически оценивать, отстаивать свои идеи. При дальнейшем освоении *LEGO EV3* становится возможным выполнение серьезных проектов, развитие самостоятельного технического творчества.

Конструктор *LEGO EV3* обеспечивает простоту при сборке начальных моделей, что позволяет школьникам получить результат в пределах одной пары уроков. Использование оболочки *leJOS* позволяет увеличить возможности в написании программ и создании проектов. Робот функционирует автономно, достаточно загрузить управляющую программу в специальный *LEGO*-компьютер и присоединить его к модели робота, *leJOS* работает независимо от настольного компьютера, на котором была написана управляющая программа; получая информацию от различных датчиков и обрабатывая ее, он управляет работой моторов.

Заключение

В результате данной работы была разработан программный продукт, устанавливаемый в систему управления неустойчивым объектом, выполняющий задачу управления роботом, движущимся по линии. Основой для создания системы послужил классический ПИД-регулятор. В качестве объекта управления послужил робот, созданный на основе конструктора *LEGO Mindstorm*. Реализация системы проведена на языке *Java*.

Представленная работа апробирована в виде лабораторного практикума в образовательной программе 5-8 классов. Полученные результаты являются основой для создания курса интеллектуальной робототехники для школьников средних и старших классов и переноса разрабатываемых современных информационных технологий в образовательные процессы школ.

Список литературы

1. Ульянов С. В. Интеллектуальная робототехника. Ч.1: Креативный когнитивный образовательный процесс – синергетический эффект альянса сотрудничества «колледж – университет – компания – рынок» // Системный анализ в науке и образовании: сетевое научное издание. — 2016. — №4. — [Электронный ресурс]. URL: <http://sanse.ru/download/274> (дата обращения: 18.03.2018).
2. Ульянов С. В., Решетников А. Г., Тятюшкина О. Ю. Интеллектуальная робототехника. Ч.2: Социально-экономическая техническая платформа когнитивного образовательного процесса // Системный анализ в науке и образовании: сетевое научное издание. — 2016. — №4. — [Электронный ресурс]. URL: <http://sanse.ru/download/277> (дата обращения: 04.02.2018).
3. Ульянов С. В., Решетников А. Г., Тятюшкина О. Ю. Интеллектуальная робототехника. Ч.3: Концепция когнитивного обучения – интеллектуальный роботренажер // Системный анализ в науке и образовании: сетевое научное издание. — 2017. — №1. — [Электронный ресурс]. URL: <http://sanse.ru/download/283> (дата обращения: 23.04.2018).
4. Srinivasa S.S., Ferguson D., Helfrich C.J., Berenson D. HERB: a home exploring robotic butler, Springer Science + Business Media, LLC 2009. — [Электронный ресурс]. URL: http://www.ri.cmu.edu/pub_files/2010/1/HERB09.pdf (дата обращения: 29.05.2018).
5. STAIR: Stanford artificial intelligence robot. — [Электронный ресурс]. URL: <http://stair.stanford.edu/> (дата обращения: 15.05.2018).

6. Карпутина А.Ю. Образовательная робототехника // Современные научные исследования и инновации. — 2016. — № 12. — [Электронный ресурс]. URL: <http://web.snauka.ru/issues/2016/12/74896> (дата обращения: 23.05.2018).
7. A. Eliazar, R. Parr. DP-SLAM. — [Электронный ресурс]. URL: <https://users.cs.duke.edu/~parr/dpslam/> (дата обращения: 19.04.2018).
8. Никляев. И. Навигация квадрокоптера с использованием монокулярного зрения. — [Электронный ресурс]. URL: <https://habr.com/company/singularis/blog/276595/> (дата обращения: 29.05.2018).
9. Антонов А. Теория и практика автоматического управления в робототехнике // Современная электроника. — 2015. — № 4. — Ср. 16-20.
10. Panda R. C. Introduction to PID Controllers – Theory, Tuning and Application to Frontier Areas. — Rijeka, Croatia : InTech, 2012. — P. 258.
11. Бусленко Н. П. Моделирование сложных систем. — М.: Наука, 1988. — С. 400.
12. Михайлов В. С. Теория управления. — К.: Выща школа, 1988. — С. 312.
13. Шеннон Р. Имитационное моделирование систем. Искусство и наука. — М.:Мир, 1978. — С. 418.
14. Маркеев А.П. Теоретическая механика: учебник для университетов. — Москва: ЧеРо, 1999. — С. 572.
15. Коробейников А.Г. Разработка и анализ математических моделей с использованием *MATLAB* и *MAPLE*: Учебное пособие. — СПб: СПбГУ ИТМО, 2010. — С. 145.
16. Базовый набор LEGO® MINDSTORMS® Education EV3. — [Электронный ресурс]. URL: <https://education.lego.com/ru-ru/product/mindstorms-ev3/core-set> (дата обращения: 16.05.2018).
17. Официальный сайт leJOS. — [Электронный ресурс]. URL: <http://lejos.org/> (дата обращения: 21.03.2018).
18. Програмируем работа LEGO Mindstorms EV3 на Java. — [Электронный ресурс]. URL: <http://www.proghouse.ru/article-box/55-lejos-ev3> (дата обращения: 22.04.2018).
19. Java for the LEGO EV3 Brick. — [Электронный ресурс]. URL: <https://sourceforge.net/projects/ev3.lejos.p/files/> (дата обращения: 06.05.2018).