

Образовательный кибернетический конструктор для использования в школах и вузах

доктор физ.-мат. наук, проф. Терехов Андрей Николаевич, Санкт-Петербургский государственный университет, зав. кафедрой системного программирования, Andrey.Terekhov@lanit-tercom.com;

Лучин Роман Михайлович, Санкт-Петербургский государственный университет, н.с. лаб. Теор. кибернетики мат.-мех. факультета, rommilu@gmail.com;

Филиппов Сергей Александрович, Физико-математический лицей №239 Центрального района Санкт-Петербурга, учитель, методист, safilippov@gmail.com.

Технический прогресс ставит новые задачи перед образованием и наукой, особенно по ряду технических направлений. Выпускники школ и вузов должны не только обладать базовыми знаниями в классических дисциплинах, но и уметь ориентироваться в новых реалиях, одной из которых является повсеместное распространение роботов и управляемых встраиваемых систем. Между тем, доступное российским школьникам и студентам учебное оборудование не отвечает современным требованиям. Идея создания и применения в обучении кибернетического конструктора нового поколения возникла и развивается на стыке школы и вуза в Санкт-Петербурге. В статье рассматривается комплекс решений для обучения программированию роботов, с помощью которого возможно обновление методов и процессов образования.

1. Введение

Элементы робототехники и кибернетики всё чаще применяются для обучения различным дисциплинам, как в школах, так и вузах. Возможность создать собственного робота и запрограммировать его вызывает живой интерес у молодых людей, начиная с младших классов. Стремительно растет количество соответствующих кружков, соревнования по робототехнике собирают большое число участников и зрителей. Подобный процесс исключительно необходим для формирования в обществе навыков использования новых технологий и развития технического творчества молодёжи.

Современные дети часто теряют интерес к учебному процессу, т. к. целиком захвачены технологиями развлечений, компьютерными играми, мобильными телефонами, телевидением и т. д. Чтобы противостоять этому процессу, необходимы технологически совершенные образовательные ресурсы, наглядные пособия, стенды и новые элементы в методике преподавания. В этом смысле большой интерес представляют робототехнические конструкторы, которые позволяют создавать автономные модели роботов и разнообразные обучающие стенды. Во многих странах мира подобные конструкторы успешно используются при работе со школьниками и студентами.

В последние годы большую популярность приобрёл конструктор LEGO® датского производства. Самые маленькие дети увлекаются сбором трансформеров и других «страшилок», говорят, что это даже полезно, так как развивает микромоторику. Для детей постарше эта же фирма выпустила конструктор, в состав которого входят моторчики, датчики и, самое главное, блок контроллера, размером с мыльницу. Оказалось, что с помощью конструктора LEGO® Mindstorms [1-3] хорошо обучать детей основам программирования.

Идею обучения программированию с помощью робота-черепашки (правда, виртуального, видимого только на экране) впервые высказал более 30 лет назад известный психолог Сеймур Пейперт [4-6]. Он обратил внимание на то что, детям важно сразу видеть результаты работы своей программы. Например, если при рисовании домика окно или дверь получаются вне дома, это сразу видно, и ребёнок, обучаемый программированию, сразу может приступить к поиску ошибки в своей программе.

Микроминиатюризация и удешевление электронных устройств позволили превратить виртуальных роботов в материальные, что ещё больше повысило интерес детей к их программированию. Во многих школах образовались кружки робототехники, проводится множество соревнований, вплоть до международных, словом, интерес к роботам огромный.

Важно заметить, что если 30 лет назад роботов можно было встретить только в фантастических романах или в виде громоздких примитивных (с современной точки зрения) станков с числовым программным управлением, то сегодня роботы широко используются в самых разных областях от промышленного производства до бытовых приборов. Соответственно проектированию роботов и методам управления роботами стали обучать во многих технических вузах, а в классических университетах в дополнение к традиционным кафедрам теоретической кибернетики, стали появляться кафедры прикладной кибернетики.

Естественно, многие преподаватели вузов попытались использовать роботы LEGO® Mindstorms для обучения студентов и даже добились определённых успехов, но только для самых простых задач. Даже школьники старших классов часто не могут реализовать разработанные ими алгоритмы управления на этих роботах. Глупо предъявлять какие-то претензии к фирме, роботы LEGO® Mindstorms и не были рассчитаны на сложные применения. Слабый процессор, маленькая память, датчики с весьма неточными результатами, разумеется, ограничивают область применения.

Значительная часть создаваемых с помощью конструктора устройств имеет в своей массе или программное управление без обратной связи или реализует достаточно простые релейные или пропорциональные регуляторы (реже – пропорционально-дифференциальные [7]). Примеров таких задач бесчисленное множество, вот некоторые из них: робот-манипулятор сортировщик, гоночный LEGOмобиль, установка сборки кубика Рубика ([8-10]).

Конечно, для первичного ознакомления с основами техники и теории управления такие задачи безусловно важны, но при обучении студентов старших курсов они уже не представляют интереса. В жизни часто можно заметить, что первоначальный бурный интерес учащихся к конструктору после знакомства с возможностями и техническими особенностями значительно спадает. Возможно, основной причиной является относительная теоретическая простота решаемых задач, так как основные трудности в упомянутых выше задачах составляют вопросы инженерно-программистского характера с типовыми, хорошо известными алгоритмами решения.

Особо нужно поговорить о программном обеспечении (ПО). Неэффективная операционная система, медленные драйверы вкуче с подключением датчиков и моторов через промежуточный процессор резко замедляют реакцию робота на внешние события. Интерпретативная реализация (через внутреннюю виртуальную машину) также существенно замедляет исполнение программы.

Большинство средств программирования этого конструктора ориентировано на решение задач, которые, может быть, и сложны технически, но с точки зрения теории достаточно просты. Часто так получается из-за огромного объема инженерно-программистской работы, которую приходится проделать в процессе решения задачи. И это, в первую очередь, обусловлено самими средствами программирования – концепции и парадигмы, лежащие в основе таких средств, не позволяют подниматься на высокий уровень абстракции. Это, в свою очередь, не дает возможности сосредоточиться непосредственно на уравнениях модели и управления.

Средства графического проектирования ПО рассчитаны на маленьких детей, но при этом довольно трудны для понимания. Чего стоят, например, 400 пиктограмм, часто мало отличающихся друг от друга. Кто их может запомнить и правильно применить? Ну и наконец, последнее по порядку, но не последнее по значению: отсутствует локализация для русскоговорящих пользователей (собственно, как и для всех других, кроме англоговорящих).

К этому надо добавить, что образовательных программ для школ по этой тематике не существует, а адаптация зарубежных разработок фактически эквивалентна созданию программы с нуля. На русском языке есть только одна книга, ориентированная на школу [7], т.е. ощущается нехватка методических рекомендаций и материалов.

Таким образом, назрела идея создания нового конструктора, существенно превышающего по возможностям LEGO® Mindstorms, который можно было бы применять сквозным образом, в цепочке: младшие школьники → старшие школьники → студенты техникумов и колледжей → студенты вузов → реальное производство. Предварительные расчёты показывают, что новый конструктор можно удержать в той же ценовой категории, что и LEGO® Mindstorms.

В качестве примеров емких задач, затрагивающих в том числе и не до конца исследованные (или оптимизированные) теоретические задачи, для которых нужен новый конструктор, могут послужить: стабилизация обратного маятника на тележке, управление колебаниями различных типов маятников, робот-велосипедист, робот на шаре (стабилизация конструкции сверху незакрепленного шара), система преследования «лев – антилопа» [11] и т. д. [12]. Важно отметить, что упомянутые задачи являются, как правило, частью более сложных задач в рамках проводимых исследований.

2. Модельно-ориентированный подход

В русском языке не существует однозначного перевода термина Model Based Design (MBD). Наиболее близким по смыслу выглядит термин модельно-ориентированный подход¹ (МОП). Оригинальный термин MDB можно определить, как технику автоматической генерации программ на основе моделей, используемых для имитационного моделирования. Практически стало стандартом при разработке систем управления создавать имитационные модели установки и регулятора в процессе проведения серий экспериментов для верификации и валидации алгоритмов управления.

¹ Надо заметить, что устоявшийся перевод термина Model Driven Architecture (MDA) тоже звучит как модельно-ориентированный подход.

Основная идея МОП — использовать модель регулятора не только для имитационного моделирования, но и для генерации кода программы, помещаемой на управляющее устройство. Если при этом есть возможность считывать данные с этого устройства, то их можно использовать для восстановления внутреннего состояния системы, которое было в реальном эксперименте, что, в свою очередь, можно использовать для дальнейшего улучшения модели регулятора. В итоге, в процессе разработки мы движемся по цепочке, которая выстраивается в спираль процесса разработки:

имитационное моделирование → генерирование кода → анализ результатов → имитационное моделирование → ...

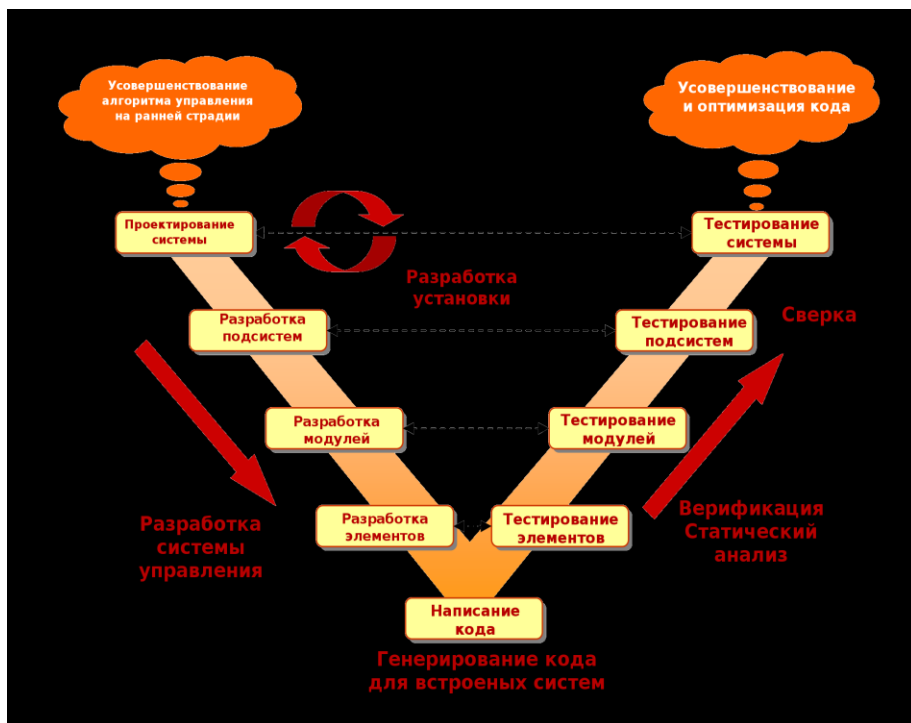


Рис. 1. V-модель разработки системы управления

При таком подходе получается, что исходный код генерируется автоматически по модели. Это позволяет в процессе разработки программы оставаться на очень высоком уровне абстракции – уровне модели (рис. 1), к тому же программа автоматически разрабатывается «сверху-вниз». Необходимо отметить и некоторые другие

преимущества подхода: раннее обнаружение ошибок проектирования систем управления, уменьшение нагрузки на устройства, проверка работоспособности путем имитационного моделирования, легче оптимизировать взаимодействие модулей, автоматическая генерация кода экономит время и избавляет от семантических ошибок.

3. Этапы обучения

Парадигма МОП позволяет предельно сконцентрировать внимание на математическом содержании программы, максимально устраняя (в пределах разумного) технические подробности процесса разработки. Это, конечно, существенный шаг вперед, 10 лет назад возможность появления таких средств казалась делом далекого будущего.

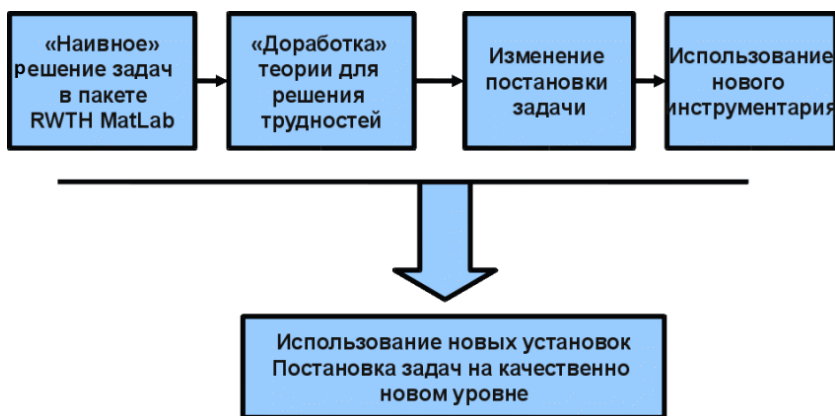


Рис. 2. Этапы обучения

Но на разных стадиях обучения имеет смысл как по-разному формулировать постановку задачи (рис. 2), так и предлагать различные инструментальные средства для их решения. Еще раз нужно подчеркнуть, что в начале процесса обучения в качестве задач предлагается брать «упрощенную» формулировку содержательной математически задачи. Отметим, что при знакомстве с программированием встроенных систем и основ теории управления все же не имеет смысла использовать такие средства программирования как NXT-G, NXC, NBC— они рассчитаны на школьников. Специально для знакомства с управлением робототехническими системами в Аахеновском Техническом Университете был разработан пакет RWTH Mindstorms NXT (около 360 студентов младших курсов каждый год проходят робототехнический практикум на LEGO® NXT [10]). Его удобно использовать для решения задач в «упрощенной» постановке.

Но рано или поздно возникнут технические трудности, для решения которых придется изменить постановку задачи, «доработать теорию» (использовать новый математический аппарат) и реализовать решение с помощью более технически совершенных средств (см. рис.2). К примеру, выясняется, что запаздывание в 0.1 секунды существенно при стабилизации обратного маятника, реализованного на LEGO® NXT, его нужно учитывать при конструировании регулятора и минимизировать при написании программы, сделать последнее в RWTH Toolbox практически невозможно. Следовательно, меняется и постановка задачи (учет запаздывания), и средство программирования (т.к. RWTH Toolbox уже не подходит).

Цель такого подхода - убедить учащихся практическим примером в необходимости использования более сложного математического аппарата. Показать, что начальные «наивные» решения не дают удовлетворительного результата. При этом еще одна цель - познакомить с универсальной технической базой программирования встраиваемых систем.

Схемы разработки приложений на основе МОП подобно пакету ECRobot используют в своих инструментальных средствах такие компании, как, например, National Instruments и dSpace, которые занимают лидирующие позиции на рынке производства встраиваемых систем управления. Появляются возможности получать обучающие стенды, выпускаемые этими компаниями. И следующим шагом процесса образования логично вытекает переход после ECRobot NXT к экспериментам с более технически совершенными стендами.

Приведенный подход опробован на мат-мехе СПбГУ при работе со студентами в рамках проекта «Киберфизическая лаборатория» (сам проект основан на сотрудничестве лица № 239, кафедра теоретической кибернетики и системного программирования математико-механического факультета СПбГУ на базе ИПМаш РАН) [12]. Участники проекта выступали на множестве конференций и имеют более 20 публикаций.

4. Реализация парадигмы

Надо заметить, что в настоящее время компанией ЗАО «Ланит-Терком» совместно с математико-механическим факультетом СПбГУ разработаны новые инструментальные средства QReal и Embox [13–15], с помощью которых можно программировать LEGO® NXT.

По сравнению с существующими инструментами QReal и Embox более универсальны, эффективны и реализуют сразу несколько подходов к программированию встроенных систем, в том числе и модельно-ориентированный. Вместе они представляют последовательную линейку решений для программирования и обучения

программированию роботов. С ее помощью возможно обновление методов и процессов образования, что, безусловно, поможет интеграции вуза и школы. Чем эффективней и проще будут инструменты, чем раньше будет возможность знакомства с ними, тем легче будет сосредоточиться на изучении задач управления, заняться задачами кибернетики (так, как это определял Н. Винер, основоположник современной кибернетики, еще в 1948 году [16]), что даёт возможность поднять образовательный процесс на качественно новый уровень.

5. Анализ сложившейся ситуации

В России и во всём мире значительно увеличиваются объёмы рынка роботов для дома и развлечения, по статистике в западных странах он увеличивается на 20-45% в год. По некоторым прогнозам в Японии этот рынок увеличится к 2013 году в 4-6 раз по сравнению с 2010 годом. Надо отметить, что одной из отличительных черт роботизированных систем является их узкая направленность на конкретный вид деятельности. Это относится и к абсолютному большинству недорогих робототехнических конструкторов, которые, как правило, позволяют собрать одну типовую конструкцию с весьма ограниченной функциональностью, ориентированной на выполнение узких задач. Немного усовершенствованные наборы позволяют собрать 3-5 моделей, но при этом, как правило, отсутствует возможность создавать полноценные программы, и всё управление роботом сводится к манипулированию дистанционным пультом.

Если говорить о LEGO® Mindstorms, то возможности конструирования в этом конструкторе ограничиваются особенностями крепления деталей (патентованная технология LEGO) и самой элементной базой (фирменные детали LEGO). Несмотря на то, что компания LEGO выпускает «Набор для обучения», направлен он прежде всего на младший школьный возраст, т. к. поставляемое с набором средство программирования NXT-G предназначено для детей 8-10 лет. Решение задач, которые могут быть интересны старшеклассникам, с помощью этого средства практически невозможно. Кроме того, использование набора предполагает, что будут использоваться только фирменные детали LEGO, что может значительно ограничивать творческие порывы детей. Надо заметить, что можно использовать только датчики и моторы, оснащённые фирменным разъёмом LEGO, что также сужает технические возможности создаваемых моделей.

Существует и другая проблема: в базовой поставке конструктора LEGO® Mindstorms отсутствуют многие датчики (гироскоп, датчик света, компас и т.д.). В магазины России они, как и большинство запасных деталей, не поставляются. При этом заказать на сайте их тоже невозможно, т.к. попросту отсутствует доставка в Россию.

Такая же ситуация наблюдается и у других роботизированных конструкторов: BIOLOID, VEX и др. Они в Россию или не поставляются, или наблюдаются большие проблемы с комплектующими. Кроме того, цена этих конструкторов весьма высока. Ситуацию усугубляет то, что при желании собрать роботизированное устройство самостоятельно комплектующие с порталов интернет-магазинов заказать невозможно, поскольку, отсутствует доставка в Россию, хотя есть доставка в страны бывшего СССР (Узбекистан, Туркмения и т.д.).

6. Анализ задачи

В настоящий момент возникла острая нужда в использовании в области образования оборудования и программного обеспечения российского производства (в частности, по причинам, описанным выше). Правильно выстроенный процесс обучения знакомству с высокотехнологичными инструментами, начиная с 4-5 класса, с продолжением в вузе или техникуме, позволит выпускать специалистов уже готовых к работе с высокотехнологичным оборудованием (разработке высокотехнологичного оборудования) российского производства. Причём навыки работы можно уже закладывать в школе. Чтобы построить такую цепочку, необходимо выполнить три фундаментальных условия.

1. Создать образовательную программу дополнительного образования детей, ориентированную на овладение современными технологиями и инновациями. При этом методика должна помогать раскрывать на практике знания, полученные на разных предметах, тем самым сохраняя фундаментальность образования.
2. Использовать открытое кроссплатформенное программное обеспечение, желательно, российского производства: важно предоставить возможность подстраивать ПО для образовательного процесса, а не наоборот, и очень важный момент - качественная локализация.
3. Создать образовательный кибернетический конструктор на российской элементной базе.

Последний пункт рассмотрим подробнее. Во-первых, конструктор должен быть предназначен для образовательного процесса, как школьников, так и студентов, т.е. чтобы можно было собрать как простые типовые модели для первых шагов в робототехнике и использования на уроках технологий и информатики, так и достаточно сложные модели, позволяющие студентам изучать современные технические задачи. Для этого конструктор должен быть легко масштабируемым – в его основу нужно закладывать большой потенциал.

Ключевую роль в этом имеет создание контроллера. Он должен обладать богатыми функциями, например, обработка аудио/видео сигналов, работа по сети wifi и т.д. Для этих целей могут подойти двухядерные микроконтроллеры, представленные на рынке, или решение, объединяющее два процессора на одной плате. Один процессор (ядро) занимается обработкой поступающих событий в режиме реального времени, второй - вычислениями (например, обработкой видеосигнала). Должна быть реализована возможность подключения новых периферийных устройств. При этом одной из главных целей должен быть переход на микросхемы российского производства. К сожалению, для быстрого достижения результата использовать их в самом начале не получится.

Новый конструктор должен быть основан на распространённой механической элементной базе. Для этих целей подходят металлические конструкторы, известные всем с детства. К этой элементной базе необходимо добавить наборы шестерёнок, передач и универсальные крепления для фиксации нестандартных элементов. Потенциал металлических наборов в нашей стране еще не раскрыт, а, между тем, он весьма велик. Конструкции получаются жёстче и легче, чем аналогичные из пластмассы.

По-настоящему кибернетическим конструктор может сделать возможность сосредоточить внимание на алгоритмах анализа и управления. Для этого необходимо, чтобы отсутствовали проблемы аппаратной части и чтобы были высокоуровневые средства программирования микроконтроллера.

Мы планируем перенести на новый контроллер уже упоминавшиеся операционную систему реального времени Embox и метасредство визуального программирования QReal. Оба этих продукта являются свободно распространяемыми и прекрасно подходят для задач обучения. Embox – это операционная система реального времени, которая применяется в промышленности, но, благодаря своей масштабируемости, легко переносится на разные архитектуры. Эта ОС обеспечивает базовую функциональность контроллера. Для создания средств визуального программирования с расчётом на разные уровни пользователей можно использовать простое и удобное средство QReal, основным назначением которого является создание языков визуального программирования. Это позволит создать спектр языков программирования, полностью обеспечивая образовательный процесс.

Завершающим этапом разработки является подготовка методических материалов, поставляемых вместе с конструктором, по которым учащийся будет иметь возможность самостоятельно освоить основные

элементы робототехники и теории управления, а преподаватель вести уроки технологии в школе или практические занятия в вузе.

С точки зрения образовательного процесса очень важно, чтобы школьный преподаватель сразу имел все необходимые материалы для "быстрого старта", т.е. чтобы была возможность начинать занятия со школьниками фактически на следующий день после получения конструктора.

7. Заключение

Надо отметить, что перечисленными в докладе техническими и образовательными свойствами не обладает ни один из известных роботизированных наборов. Единственным выходом является создание такого набора с нуля. Тем не менее, значительная часть работы уже проделана сотрудниками СПбГУ, и есть надежда увидеть опытные образцы уже к новому 2012 году. Сейчас ведётся разработка методических материалов на базе ЛГУ им. А. С. Пушкина совместно с Петербургским городским центром робототехники. С текущими результатами работы можно ознакомиться на сайте роботека.ком.рф.

Очевидно, что конструктор может послужить базой для таких студенческих и школьных проектов, как создание бюджетных систем для автономного управления роботами-футболистами, мультиагентной сети беспилотных летательных аппаратов и т.п. Подобного рода проекты не только способствуют развитию технического творчества, но и являются прекрасной площадкой для исследований в области искусственного интеллекта, компьютерных наук и современных информационных технологий.

Литература

1. LEGO Technic Tora no Maki, ISOGAWA Yoshihito, Version 1.00 Isogawa Studio, Inc., 2007, электронный ресурс [<http://www.isogawastudio.co.jp/legostudio/toranomaki/en/>].
2. CONSTRUCTOPEDIA NXT Kit 9797, Beta Version 2.1, 2008. Center for Engineering Educational Outreach, Tufts University, // электронный ресурс [http://www.legoengineering.com/library/doc_download/150-nxt-constructopedia-beta-21.html].
3. Lego Mindstorms //Wikipedia, the free encyclopedia электронный ресурс [http://en.wikipedia.org/wiki/Lego_Mindstorms].
4. Papert S. Mindstorms: Children, Computers, and Powerful Ideas. Basic Books, 1980.
5. Пейперт С. Переворот в сознании: Дети, компьютеры и плодотворные идеи. Москва, Педагогика, 1989.
6. Papert S. The Children's Machine: Rethinking School in the Age of the Computer, 1992.

7. Филиппов С. А. Робототехника для детей и родителей. СПб.: Наука 2010.
8. Benedettelli, D.; Casini, M.; Garulli, A.; Giannitrapani, A.; Vicino, A. A LEGO Mindstorms experimental setup for multi-agent systems // Proc. IEEE Conf. Control Applications, & Intelligent Control, IEEE 2009, pp. 1230 – 1235.
9. Valera, A.; Valles, M.; Fernandez, A.; Albertos, P. Platform for the development of mechatronic practical works based on LEGO Mindstorms NXT robots // Control Applications & Intelligent Control, IEEE, 2009, pp.1224-1229.
10. Lehrstuhl für Bildverarbeitung. MATLAB MeetsLEGO Mindstorms электронный ресурс [<http://www.mindstorms.rwth-aachen.de>]
11. Якубович В. А. Об одной задаче самообучения целесообразному поведению // Автоматика и телемеханика. 1969. №8. С. 119–139.
12. Филиппов С. А., Фрадков А.Л. LEGO-роботы в обучении мехатронике и автоматизации в школе и вузе. // Тез. док. 7-й науч.-техн. конф. "Мехатроника, Автоматизация, Управление" (МАУ-2010), ГНЦ РФ ОАО "Концерн "ЦНИИ "Электроприбор", СПб, 2010. С. 49-52.
13. QReal // электронный ресурс [<http://se.math.spbu.ru/SE/qreal>].
14. EMBOX // WIKIPEDIA.ORG электронный ресурс [<http://ru.wikipedia.org/wiki/Embox>]
15. Терехов А. Н., Брыксин Т. А., Литвинов Ю. В. и др., Архитектура среды визуального моделирования QReal. // Системное программирование. Вып. 4. СПб.: Изд-во СПбГУ. 2009, С. 171-196.
16. Винер Н. Кибернетика, или управление и связь в животном и машине. М.: Наука, 1983.