

Е.В. Красильникъянц, А.А. Варков, В.В. Тютиков

Программное обеспечение систем управления промышленными роботами

Рассмотрены особенности архитектуры программного обеспечения системы управления манипуляционным роботом и определены подходы к ее реализации. Описаны возможности языка программирования, разработанного для использования в системе управления.

The peculiarities of the architecture of the software of a system of a control of manipulation robot are considered and the approaches to its implementation are defined. The possibilities of the programming language development for the use in a control system are described.

Розглянуто особливості архітектури програмного забезпечення системи управління маніпуляційним роботом і визначено підходи до її реалізації. Описано можливості мови програмування, розробленої для застосування в системі управління.

Введение. Характеристики современных технологических систем, в частности систем управления промышленными роботами (СУ ПР), в равной мере зависят как от аппаратных, так и от программных особенностей. Выбор аппаратной архитектуры определяет способы их взаимодействия с управляемым оборудованием, соответствие желаемым техническим характеристикам, а также возможности интеграции в более сложные производственные единицы, например ГПС. Для расширения возможностей внедрения разрабатываемых СУ предлагается использовать двухпроцессорную архитектуру с контроллером движения в качестве основного вычислительного элемента блока управления (БУ) [2].

В связи с тем, что выбранная аппаратная архитектура устанавливает ограничения на вычислительные ресурсы, доступные для использования программными компонентами, возникает проблема эффективного распределения ограниченных ресурсов для обеспечения максимальной функциональности и удобства в эксплуатации системных и пользовательских программных средств. Решение указанной проблемы требует анализа особенностей управляемого объекта и построения иерархии используемых программных средств.

К особенностям СУ ПР, отличающим их от других систем управления движением (СУД), относятся независимо управляемые механические оси, между которыми существуют кинематические и динамические связи, нелинейность кинематической схемы, необходимость согласованного управления отдельными приводами для выполнения перемещений.

Для эффективного управления оси манипулятора объединяются в координатные системы.

Тип координатной системы (КС) определяется кинематическими соотношениями между отдельными осями [1]. Это может быть декартова, цилиндрическая, сферическая или вращательная система координат. Использование различных координатных систем для управления ПР и написания управляющей программы (УП) требует наличия в составе программного обеспечения (ПО) средств преобразования между ними.

Согласно предложенной архитектуре построения СУ основными компонентами аппаратной части есть контроллер движения (КД) – основа блока управления и промышленный компьютер (ПК), на котором реализовано терминальное устройство (ТУ) [3]. Такое решение позволяет разделить системные и прикладные задачи и обеспечить возможность модификации отдельных компонентов СУ ПР при ее использовании с различными типами роботов. Таким образом в составе программного обеспечения можно выделить уровень ПО контроллера движения и уровень ПО терминального устройства (рис. 1).

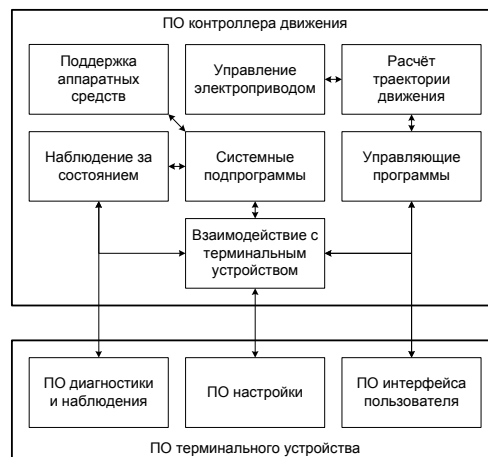


Рис. 1. Структура программного обеспечения СУ ПР

ПО терминального устройства работает независимо от КД и реализует задачу взаимодействия с пользователем. Оно предназначено для поддержки средств отображения информации о состоянии манипулятора, ввода/вывода информации, удаленного взаимодействия.

ПО контроллера движения обеспечивает решение основных системных задач и задач управления движением ПР. Оно должно выполнять следующие функции:

- поддержку аппаратных средств;
- управление электроприводом;
- расчет траектории движения;
- наблюдение за состоянием системы;
- взаимодействие с терминальным устройством;
- выполнение системных подпрограмм;
- выполнение управляющих программ пользователя.

Под **поддержкой аппаратных средств** понимается набор подпрограмм, обеспечивающих взаимодействие с электрооборудованием СУ и периферийными устройствами. Данные подпрограммы предназначены для реализации таких задач как опрос датчиков, организация работы интерфейсов передачи данных, опрос состояния ПР и других компонентов СУ. Кроме того, на рассматриваемом уровне решается задача синхронизации расчетных процессов, выполняемых на КД согласно разработанной таблице приоритетов.

Общие принципы построения ПО данного уровня заключаются в разработке модулей, работающих непосредственно в ядре системного ПО или, если структура КД не позволяет этого, – в виде задачи с наивысшим приоритетом. Взаимодействие с задачами более низкого уровня осуществляется посредством встроенных буферов данных для накапливаемой информации, таких как ввод/вывод через информационный интерфейс или специальных областей памяти для информации о состоянии системы.

Реализация упомянутых задач зависит прежде всего от аппаратных особенностей СУ ПР. Существующие КД поддерживают различные виды датчиков, форматов данных и информационных протоколов на уровне встроенного ПО, что

позволяет избежать необходимости написания собственных модулей для решения этих задач.

Таким образом, основное внимание при создании СУ ПР может быть уделено разработке средств синхронизации различных процессов и взаимодействия с платами расширения, разработанными непосредственно для СУ. Следует отметить, что основные операции, решаемые СУ, выполняются непосредственно на КД, в то время как периферийное оборудование служит для обработки получаемых данных, преобразования их в пригодную для дальнейшего использования форму. Поэтому распределения вычислительных процессов в данном случае не происходит, и проблемы согласования, связанные с параллельными вычислениями, не возникают. Для организации передачи и обмена данных используются выделенные области памяти, предназначенные для хранения текущих значений определенных системных переменных. Доступ к ним осуществляется на основе использования стандартных средств КД.

Управление электроприводом предусматривает опрос каналов обратной связи, реализацию координатных преобразований, расчет регуляторов, формирование управляющих сигналов, коммутацию силовых ключей. Поскольку предлагаемая аппаратная архитектура позволяет управлять различными типами двигателей, необходим набор программных средств, реализующий эти функции. К основным типам двигателей, используемых в ПР, следует отнести: шаговые, постоянного тока, вентильные. В перспективе возможно использование асинхронных двигателей как наиболее дешевых и надежных.

Современные КД поддерживают алгоритмы управления указанными типами двигателей. Информация о переменных состоянии двигателей и электромеханической системы в целом присутствует в системной памяти КД. Для минимизации вычислительных ошибок в качестве формата представления данных выбраны целочисленные значения и числа с фиксированной точкой.

Работа алгоритма управления электроприводом заключается в формировании управляющего сигнала на основании данных о перемен-

ных состояния двигателя, которые могут быть получены либо непосредственно через каналы обратной связи с электродвигателем (токи, скорость, положение) или вычислены в соответствующей подпрограмме (моменты).

Для обеспечения желаемого качества управления требуется синхронизация выполнения алгоритмов управления электроприводом с другими задачами, выполняемыми на КД. Для этого при разработке СУ ПР следует предусмотреть возможность разделения задач по приоритетам и задания частоты их выполнения. Современные высокопроизводительные КД позволяют проводить вычисления и выполнять подпрограммы, требуемые для управления электродвигателем, с частотой порядка 9–10 кГц без значительного снижения производительности при выполнении прочих задач.

Большинство задач управления электроприводом манипулятора может быть решено с использованием типового ПИД-регулятора. Тем не менее, в ряде задач, требующих точного и быстрого перемещения, необходимо использование модифицированного регулятора, позволяющего не только задавать координаты перемещения, но и учитывать желаемые физические характеристики движения, такие как скорость, ускорение и создаваемый момент. Для решения более сложных задач следует использовать возможности КД по созданию расширенных вариантов регуляторов. Прежде всего это относится к упругим системам, а также к системам, обладающим нелинейностями перемещений. Для управления ими требуются более сложные типы регуляторов, например такие как модальные регуляторы [4].

В связи с тем, что на работу ПР значительное влияние оказывают существующие в нем кинематические и динамические связи, при использовании робота в процессах, нуждающихся в максимально точном перемещении, необходимо создание адаптивной системы управления. Для ее реализации на КД используются встроенные средства написания пользовательских регуляторов и осуществления взаимодействия с манипулятором.

Для реализации алгоритма регулятора, используется либо язык ассемблера, либо язык,

предназначенный для написания системных компонентов КД. *Первый* подход обеспечивает более высокую производительность, но требует от разработчика знания особенностей процессора и средств взаимодействия с системными областями памяти. *Второй* – позволяет использовать специальные команды получения необходимых переменных из системной памяти КД. В современных высокопроизводительных КД применение языка высокого уровня оказывается эффективнее, чем использование ассемблера, так как позволяет проводить разработку при помощи отладочного и диагностического программного обеспечения [5].

Расчет регулятора должен быть синхронизирован с другими процессами, в первую очередь – с формированием управляющего сигнала и планировщиком траектории. Поэтому расчет контура тока должен выполняться с частотой порядка 9–10 кГц. Расчет контура положения в виду особенностей управления ПР (использование различных КС, влияние пространственной конфигурации) и необходимости согласования с планировщиком траектории, ограничен по быстродействию, поэтому в СУ он должен обновляться с частотой 2–2,5 кГц.

В состав **средств расчета траектории движения** должны входить такие компоненты, как модуль исполнения кадра управляющей программы, планировщик траектории (ПТ), программы расчета прямой и обратной кинематики, подпрограммы расчета динамических характеристик ПР.

Модуль исполнения кадра управляющей программы осуществляет выполнение представленного в виде набора команд КД текущего кадра управляющей программы и синхронизацию состояния системных и пользовательских переменных, а также различных флагов, определяющих состояние СУ. Обработанные команды движения передаются для дальнейшего выполнения в планировщик траектории.

Модуль исполнения кадра можно рассматривать как компонент виртуальной машины, реализованный на КД и преобразующий сформированный из управляющей программы псевдокод в набор инструкций контроллера движе-

ния. Набор команд и их семантическое и функциональное значение определяются базовым ПО КД. Оно предоставляет разработчику набор команд, предназначенных для указания типа, координат и характеристик перемещения, проведения математических вычислений, выполнения операций с системной и пользовательской памятью. Указанного набора команд достаточно для реализации основных операций, требуемых при управлении ПР.

Основная задача **планировщика траектории** – преобразование заданных в управляющей программе движений в последовательность задающих сигналов. Он поддерживает набор алгоритмов, обеспечивающий расчет основных видов траекторий, используемых в работе ПР: линейных, круговых, сплайновых. В связи с необходимостью преобразования координат между различными видами КС для их реализации используются интерполяторы, взаимодействующие с программами расчета кинематики ПР.

В СУ должна быть предусмотрена возможность расширения алгоритмов интерполяции разработчиком путем введения в них дополнительных операций и ограничений. В частности, разработчик может осуществлять проверки нахождение манипулятора в рабочей зоне, вводить ограничения по переменным состояния (скорость, ускорение, момент), что позволит включить в состав системы управления средства обеспечения безопасности при выполнении технологических операций.

Использование **программ расчета кинематики** необходимо для перевода заданных движений из координатной системы, применяемой в управляющей программе, в координатную систему, используемую в манипуляторе, и обратно. При этом кинематические характеристики определяются конфигурацией ПР. Решение задач прямой и обратной кинематики в многозвенных манипуляторах требует проведения ряда пространственных преобразований, основанных на матричных вычислениях. Большинство используемых ПР обладают пятью и более осями, что определяет высокий порядок матричных уравнений и требует от КД значительных вычислительных ресурсов. Для повышения про-

изводительности вычислений при разработке СУ была решена задача оптимизации вычислений электромеханической модели ПР [6].

В зависимости от пространственной конфигурации ПР в текущий отрезок времени изменяются его динамические характеристики. Для оценки их влияния на поведение ПР требуется решение уравнений динамики манипуляционного робота. Существует ряд методов их решения [1, 7]. Исходя из вычислительных возможностей КД был выбран подход, минимизирующий затраты вычислительных и временных ресурсов [6]. Поскольку КД, как правило, не имеют программных средств для выполнения операций с матрицами высокого порядка, для решения задач кинематики и динамики был выбран путь замены матричных операций вычислениями с использованием отдельных элементов. Такой подход позволил оптимизировать вычисления путем удаления элементов с малыми весовыми коэффициентами в итоговом значении.

Для использования СУ с различными видами ПР, отличающихся своими характеристиками, на этапе проектирования при расчетах уравнений кинематики и динамики реализована возможность конфигурирования основных параметров. Для этого представлены выделенные области памяти для хранения информации о количестве осей, кинематических и динамических параметрах ПР.

Средства наблюдения предназначены для осуществления обмена информацией о состоянии отдельных компонентов СУ, а также для защиты от возможных сбоев в работе аппаратного и программного обеспечения. К ним относятся сторожевые таймеры, внешние датчики и измерители физических величин, позволяющие определить возникновение различных проблемных ситуаций, а также программные счетчики и параметры, необходимые для первичной оценки поведения ПР и выполнения задач движения. Данный модуль программного обеспечения реализован в виде отдельной системной подпрограммы, выполняемой в реальном времени, но с более низким приоритетом, чем задачи управления движением.

Средства взаимодействия с терминальным устройством предназначены для обмена дан-

ными и управляющей информацией с пользователем посредством терминального устройства. К ним относятся различные буферы для передачи информации, средства интерфейса с переменными состояния системы.

Для выполнения **системных подпрограмм** существует уровень программного обеспечения, выполняемый на КД в фоновом режиме и осуществляющий обновление различных данных, наблюдение за характеристиками СУ, а также синхронизацию процессов в пользовательских программах. Критические задачи, требующие немедленной реакции, такие как обработка ошибок, выполняются с более высоким уровнем приоритета.

Управляющие программы пользователя, передаваемые для выполнения в КД, представляются в виде макрокоманд, которые должны быть преобразованы в систему команд КД для их выполнения при достижении соответствующего кадра. Набор макрокоманд определяется средствами ПО ТУ в первую очередь языком описания УП.

В связи с тем, что для работы СУ ПР необходимо обеспечивать выполнение различных задач в режиме реального времени, возникает вопрос о рациональном распределении между ними временного ресурса. На рис. 2 в виде блоков показано распределение выполняемых процессов по уровням приоритетов. Ширина блока показывает примерное соотношение частот их обновления в рамках одного приоритета.

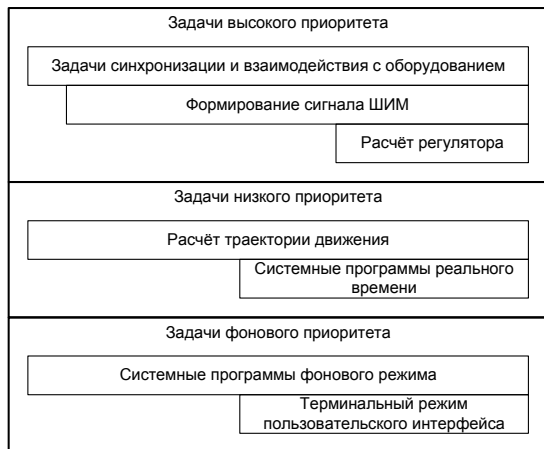


Рис. 2. Распределение задач ПО КД по уровням приоритетов

Основные задачи взаимодействия с оборудованием СУ и коммутации управляемых двигателей относятся к наиболее высокому приоритету. Такое решение гарантирует, что СУ находится в рабочем состоянии независимо от времени, которое потребуется для задач более низких уровней.

Задачи низкого приоритета ответственны за формирование траектории движения и обновление состояния СУ через системные программы. Предоставление им данного приоритета позволяет гарантировать, что они будут выполняться с определенной частотой.

Задачи фонового приоритета выполняются в свободное от решения других задач время. Это позволяет избежать блокировки выполнения важных задач в случае возникновения сбоев в работе системных программ и обмене данными с терминальным устройством.

Важная особенность СУ ПР, отличающая ее от других СУД, – наличие языка программирования высокого уровня, предназначенного для написания управляющих программ пользователя. В отличие от систем ЧПУ для СУ ПР не существует единого стандарта, описывающего особенности языка. В результате производители ПР реализуют собственные подходы к его разработке и созданию.

В связи с тем, что в предложенной архитектуре основные задачи управления решаются на КД, обладающем ограниченными вычислительными ресурсами, а промышленный компьютер служит в качестве терминального устройства, организующего человекомашинный интерфейс, на язык программирования накладывается ряд ограничений. В частности, нецелесообразно использование существующих сред программирования и языков высокого уровня, предоставляющих избыточную функциональность, а набор реализуемых возможностей языка определяется прежде всего особенностями ПР. Также следует отметить, что применение для написания управляющих программ языков высокого уровня требует от оператора расширенных знаний в области программирования. Поэтому при разработке СУ ПР был выбран подход, подразумевающий создание собственного языка, близ-

кого по функциональным возможностям широко используемым на рынке ПР решениям *RAPID (ABB), Karel (Fanuc), KRL (KUKA)*.

К основным принципам, использованным при разработке языка программирования для СУ ПР, следует отнести модульность и открытость. Модульность позволяет разработчику и пользователю СУ создавать собственные библиотеки подпрограмм и компонентов с применением наиболее часто используемых алгоритмов. Открытость предоставляет пользователю возможность написания собственных программ и доступа к основным конфигурационным параметрам для создания алгоритмов, взаимодействующих с системными компонентами СУ.

Управляющая программа, составленная на разработанном языке, должна передаваться в БУ, где она преобразуется в его внутреннее представление. Набор средств языка определяет перечень макрокоманд анализируемых КД при вводе программы для ее перевода в команды КД. Учитывая особенности управления ПР, к языку описания УП предъявляется ряд требований, связанных с использованием в составе программы различных координатных систем и управлением движением как отдельных осей, так и манипулятора в целом. К ним следует отнести:

- задание перемещений по требуемой траектории с выбранными параметрами (скорость, ускорение);
- преобразование координат точек траектории перемещения между различными видами координатных систем;
- взаимодействие с аппаратным обеспечением БУ и периферийными устройствами;
- организация сложных управляющих программ с элементами алгоритмического программирования;
- взаимодействие с библиотекой подпрограмм.

Как и в большинстве языков программирования высокого уровня для хранения данных предполагается использование переменных определенного типа. Основные типы данных при написании УП ПР следующие: координата, целое число, число с точкой, булево значение, текстовая строка. Для хранения данных и вы-

полнения различных алгоритмов переменные могут быть представлены не только в виде отдельных значений, но и в виде массивов.

Рассмотрим реализацию отдельных функциональных возможностей языка.

Для задания перемещений используются следующие команды:

- произвольное перемещение между двумя точками (*point, ptp*);
- перемещение между двумя точками по линейной траектории (*linear*);
- перемещение по окружности с заданием определенного набора точек (начальная координата, центр окружности, конечная координата или радиус) (*circle*);
- перемещение между точками по сплайну (*spline*);
- задание скорости перемещения (*speed*);
- задание максимального ускорения при перемещении (*accel*).

Перечисленные команды позволяют разработчику или пользователю задать желаемую траекторию перемещения ПР в рабочей области с указанием ее характеристик, таких как скорость и ускорение. Следует отметить, что они работают с учетом ограничений, накладываемых планировщиком траектории, при нарушении которых СУ автоматически прерывает выполнение программы.

Учитывая, что ПР может работать не в одной системе координат, в язык программирования включен набор средств, предназначенных для их преобразования. Наиболее удобным с точки зрения пользователя есть задание координат в виде структур, включающих в себя элементы как рабочей, так и собственной КС ПР. Данное представление позволяет использовать в одной УП координаты различных КС без необходимости пользователю следить за текущим режимом.

Представление структур основано на стандартах, существующих в языках высокого уровня общего назначения. Например, для обращения к координате в переменной, определяющей точку, разработчик должен использовать оператор обращения к компоненту структуры, за которым следует имя координаты. Такое реше-

ние позволит сделать программу удобной для написания и анализа, в сравнении с альтернативными вариантами, такими как применение массивов элементов. Тем не менее, при использовании СУ ПР с РТК, оснащенным дополнительными осями перемещения, может потребоваться введение дополнительных компонентов структуры, предназначенных для обращения к координатам дополнительных осей. В данном случае для них следует использовать массив, обращение к элементам которого, удобнее выполнить по номеру дополнительной оси.

Следует отметить, что при задании координат в рабочей КС может возникнуть неоднозначность при формировании траектории в связи с тем, что одному и тому же набору координат соответствует несколько конфигураций ПР в собственной КС. Для решения этой проблемы в состав средств языка введены команды, указывающие на требуемую конфигурацию ПР в пространстве. Поскольку различные манипуляторы могут различаться по конфигурации, данные команды должны иметь не жестко определенные значения, а учитывать особенности кинематической структуры ПР. Таким образом, значение их параметров должно определяться не описанием языка, а конкретной реализацией системного ПО СУ ПР, предназначенной для использования с манипулятором.

Для организации взаимодействия УП с системными переменными и периферийными устройствами наиболее эффективно определение массивов данных, отображающих в УП их состояние. Для удобства пользователя вводятся псевдонимы основных характеристик и параметров в описании языка и предоставляются пользователю возможности создания собственных псевдонимов и мнемонимен для других переменных, а также входов и выходов аппаратных средств. Такое решение добавляет уровень абстрактности к взаимодействию УП и СУ, что позволяет разделить программную и физическую реализацию взаимодействия с устройствами.

Для создания сложных программ язык программирования поддерживает набор средств алгоритмического программирования, позволяющих оптимизировать процесс разработки и вы-

полнения программы. По аналогии с языками высокого уровня к ним относятся подпрограммы, циклы, ветвления.

Подпрограммы предназначены для выделения отдельных блоков кода для многократного их использования в теле программы. Их можно использовать как для простого выполнения команд, так и для проведения вычислений либо иных операций, обеспечивающих получение результата (проверка значения системных переменных, чтение состояния системных устройств).

Циклы предназначены для организации повторяющихся операций. Следует учитывать, что в программе могут быть использованы циклы нескольких типов:

- безусловный цикл *loop* (без условия выхода);
- параметрический цикл *for* (с изменением значения переменной);
- цикл с предусловием *while* (с предварительным условием).

Кроме собственно циклов в языке предусмотрены команды для их досрочного прерывания и повторения (*break* и *continue* соответственно).

Команды ветвления предназначены для организации сложных программ, управляемых условными операторами. В коде программы они могут быть представлены в виде блоков условий *if-else-endif*, сопровождающихся наборами команд для отработки при выполнении/невыполнении условия.

Для создания библиотек алгоритмов разработчик или пользователь могут воспользоваться модульным принципом построения языка программирования. Отдельные модули загружаются в исполняемую управляющую программу посредством команд и директив.

Приведем фрагмент управляющей программы «укладывания гаек в ящик», написанной с использованием разработанного языка:

```
def PutObject(ObjNo)
  pos PyrPos ; Позиция гайки
  int Level ; Ярус
  int LevelNo ; Номер гайки в ярусе
  PyrPos.x = PyrCenter.x
  PyrPos.y = PyrCenter.y
  PyrPos.z = PyrCenter.z
```

```

PyrPos.a = PyrCenter.a
PyrPos.b = PyrCenter.b
PyrPos.c = PyrCenter.c
if (ObjNo < 8) then
; 1- и 2-й ярус
if ((ObjNo == 0) or
(ObjNo == 4)) then
PyrPos.x = PyrPos.x-50
else
if ((ObjNo == 1) or
(ObjNo == 5)) then
PyrPos.x = PyrPos.x+50
else
if ((ObjNo == 2) or
(ObjNo == 6)) then
PyrPos.y = PyrPos.y-50
PyrPos.a = 90
else
if ((ObjNo == 3) or
(ObjNo == 7)) then
PyrPos.y = PyrPos.y+50
PyrPos.a = 90
endif
endif
endif
endif
if (ObjNo > 3) then
PyrPos.z = PyrPos.z+ObjN
endif
endif
PyrPos.z = PyrPos.z+ApprHeight
PTP PyrPos
PyrPos.z = PyrPos.z-ApprHeight
PTP PyrPos
PyrPos.z = PyrPos.z+ApprHeight
PTP PyrPos
end

```

Заключение. Предложенный подход к созданию и разработке программного обеспечения систем управления промышленными роботами позволяет наиболее эффективно использовать возможности выбранной аппаратной архитектуры и разработать СУ, соответствующую современным технологическим требованиям. Он позволяет максимально использовать возмож-

ности КД и обеспечить гибкость и адаптацию СУ для использования с различными видами промышленных роботов при решении производственных задач.

Разработанный язык программирования имеет развитую систему команд, широкий набор функциональных возможностей и позволяет пользователю создавать сложные управляющие программы.

1. Фу К., Гонсалес Р., Лу К. Робототехника. – М.: Мир, 1989. – 624 с.
2. Красильникьянц Е.В., Бурков А.П., Иванков В.А. Применение контроллеров движения для систем управления электромеханическими объектами // Мехатроника, Автоматизация, Управление. – 2008. – 2. – С. 45–50.
3. Красильникьянц Е.В., Варков А.А., Тютиков В.В. Система управления манипуляционным роботом // Автоматизация в промышленности. – 2011. – 5. – С. 38–44.
4. Тарарыкин С.В., Тютиков В.В. Системы координирующего управления взаимосвязанными электроприводами. – Иваново: Изд-во Иван. гос. энерг. ун-та, 2000. – 212 с.
5. Варков А.А., Бурков А.П. Универсальный язык программирования для разработки систем управления. // Тез. докл. Тринадцатой междунар. науч.-техн. конф. студентов и аспирантов «Радиоэлектроника, электроника и энергетика»: В 3-х т. – М.: Изд. дом МЭИ, 2007. – Т. 1. – С. 37–38.
6. Tararykin S.V., Krasilnikyants E.V., Varkov A.A. Electromechanical model of manipulator robot // Proc. 8th Intern. Symp. «Topical problems in the Field of Electrical and Power Engineering», Pärnu. – 2010. – P. 291–295.
7. Armstrong B., Khatib O., Burdick J. The Explicit Dynamic Model and Inertial Parameters of the PUMA 560 Arm // Proc. 1986 IEEE Intern. Conf. on Robotics and Automation. – 1986. – P. 510–518.

Поступила 10.10.2011

E-mail: artem.varkov@gmail.com

© Е.В. Красильникьянц, А.А. Варков, В.В. Тютиков, 2011