

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ ЦИФРОВОЙ ОБРАБОТКИ ИНФОРМАЦИИ ПРИ ЕЁ ГОЛОСОВОМ ВВОДЕ

И. Ш. Невлюдов, И. Д. Перепелица, С. С. Милютина

Харьковский национальный университет радиоэлектроники
просп. Ленина, 14, г. Харьков, Украина. E-mail: tapr@kture.kharkov.ua

Д. В. Мосьпан

Кременчугский национальный университет имени Михаила Остроградского
ул. Первомайская, 20, г. Кременчуг, 39600, Украина.

Предложена программная реализация метода голосового ввода информации при управлении роботизированным производством. Рассматривается структура подсистемы голосового ввода информации в САПР управляющих программ робота РМ-01. Описаны ключевые классы, реализующие голосовой ввод информации в форме управляющих голосовых команд. Рассматриваются функции, обеспечивающие ввод информации, её обработку, проверку и выполнение. Представлен интерфейс работы пользователя, а именно, главное окно и окно распознавания. Приведен состав языка управления роботом и структура библиотеки возможных для введения слов. Разработанная прикладная программа позволяет повысить скорость проектирования технологических процессов сборки роботизированного производства путём интерактивного формирования прикладных программ сборочного робота, сократить время переналадки оборудования и снизить стоимость проектирования упомянутых технологических процессов.

Ключевые слова: голосовое управление, робот, библиотека, лексема.

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ЦИФРОВОЇ ОБРОБКИ ІНФОРМАЦІЇ ПРИ ЇЇ ГОЛОСОВОМУ ВВЕДЕННІ

І. Ш. Невлюдов, І. Д. Перепелиця, С. С. Милютіна

Харківський національний університет радіоелектроніки
просп. Леніна, 14, м. Харків, Україна. E-mail: tapr@kture.kharkov.ua

Д. В. Мосьпан

Кременчуцький національний університет імені Михайла Остроградського
вул. Першотравнева, 20, м. Кременчук, 39600, Україна.

Пропонується програмна реалізація методу голосового введення інформації під управлінням роботизованим виробництвом. Розглядається структура підсистеми голосового введення інформації в САПР керуючих програм робота РМ-01. Описано ключові класи, що реалізують голосове введення інформації в формі керуючих голосових команд. Розглядаються функції, які забезпечують введення інформації, її обробку, перевірку та виконання. Також наведено інтерфейс роботи користувача, а саме, головне вікно та вікно розпізнавання. Наведено склад мови керування роботом і структура бібліотеки можливих для введення слів. Розроблена прикладна програма дозволяє підвищити швидкість проектування технологічних процесів зборки роботизованого виробництва шляхом інтерактивного формування прикладних програм зборочного робота, скоротити час переналадки обладнання та знизити вартість проектування зазначених технологічних процесів.

Ключові слова: голосове керування, робот, бібліотека, лексема.

АКТУАЛЬНОСТЬ РАБОТЫ. Современное производство характеризуется большой гибкостью и необходимостью постоянной переналадки оборудования.

Существующие приёмы обеспечения гибкости производства в настоящее время не достаточно эффективны. Поэтому программная реализация метода голосового ввода информации подсистемы ввода информации САПР технологических процессов, который имеет целью упростить процесс проектирования технологических процессов сборки роботизированного производства путём интерактивного формирования прикладных программ сборочного робота, является актуальной задачей.

Цель работы – разработка программного обеспечения для цифровой обработки информации при её голосовом вводе.

МАТЕРИАЛ И РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЙ. Экспериментальные исследования производилась на базе промышленного робота РМ-01 в рамках научной программы, связанной с разработкой интеллек-

туальной роботизированной системы.

В её состав входят:

- подсистема управления роботом;
- подсистема поддержки принятия решений;
- подсистема технического зрения;
- подсистема голосового ввода информации.

Схематично данная роботизированная система представлена на рис. 1.

Поэтому основными задачами практической части работы являются:

- анализ использования стандартных методов обработки аудиоинформации в операционной системе Microsoft Windows и разработка соответствующего программного обеспечения;
- анализ технологии, реализованной в библиотеке Microsoft Speech Engine 5.1 и реализация голосового ввода информации на её основе;
- описание разработки программного обеспечения, реализующего проектирование технологических переходов и операций при помощи голосового формирования управляющих команд.

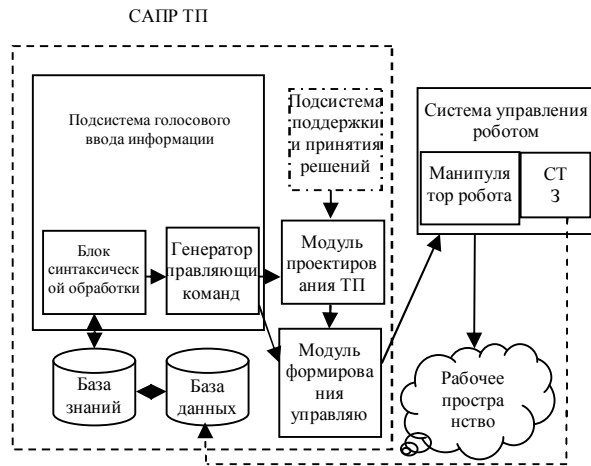


Рисунок 1 – Подсистема голосового вводу інформації в САПР управляючих програм робота РМ-01

При разработке программы, реализующей голосовой ввод информации в форме формирования управляющих команд, создан проект, в котором необходимо выделить следующие классы: CApp (класс приложения), SEasyCCRecognition (реализует непосредственное распознавание слов), CSDialog (обеспечивает необходимую проверку и трансляцию произнесенной фразы в команду), COpenGL (моделирование работы робота), CSerialException (класс, описывающий исключения при работе серийного порта), CSerialPort (класс, обеспечивающий передачу данных в серийный порт).

С точки зрения голосового ввода информации основные функции содержат классы SEasyCCRecognition и CSDialog, рассмотрим их.

Функции, содержащиеся в классе SEasyCCRecognition, приведены в табл. 1.

Таблица 1 – Функции класса SEasyCCRecognition

Название функции	Описание
PutStart	Переход в режим распознавания
GetStart	Проверка включения распознавания
GetIsStarted	Проверка состояния распознавания
PutIsListening	Установка режима ввода голосовых команд
GetIsListening	Проверка, включён ли режим ввода голосовых команд
PutXml	Определение словаря (библиотеки)
GetTextRecognized	Получение распознанного текста
PutRecognize	Установка распознавания

Функции, содержащиеся в классе CSDialog, приведены в табл. 2.

Функции get, word, index, number, prog, exam, newob, dvarb, dfunc, param, body, defin, gen реализуют работу транслятора.

Формирование голосовой команды, а также её выполнение начинается в функции OnTextRecognized,

рассмотрим её работу. Прежде всего, необходимо отметить, что в соответствии с принятыми правилами построения команды, до тех пор, пока не будет произнесено предупредительное слово «robot», система находится в состоянии бездействия. Все слова и словосочетания, произнесенные после этого слова, записываются в файл. После введения слова «end» запись прекращается, состояние ввода голосовых команд устанавливается равным false, и управление передаётся функции analys.

Таблица 2 – Функции класса CSDialog

Название функции	Описание
OnInitDialog	Инициализация диалогового окна приложения
RunJoint	Обеспечение перемещения звена
InitRobot	Инициализация робота
InitJointSpeed	Установка скорости поворота звена
ReadXml	Выбор файла, содержащего библиотеку
WordsXml	Подключение и чтение библиотеки допустимых слов
OnBnClickedToggle	Включение/выключение режима ввода голосовых команд
OnTextRecognized	Определение порядка действий после распознавания слов
RunVoiceCommand	Начало выполнения голосовой команды
analys	Анализ произнесенных и распознанных слов
get	Получение слова из файла, содержащего все распознанные слова
word	Проверка слова
index	Проверка символа
number	Проверка числа
prog	Запуск выполнения проверки слов
exam	Проверка правильности лексемы
param	Проверка параметров
body	Проверка соответствия лексем

В этой функции сначала выполняется функция get, которая считывает посимвольно слова из файла с произнесенными словами. Когда считывание доходит до конца файла, управление возвращается в предыдущую функцию. Пока считываемые символы являются пробелами, функция последовательно переходит к следующим символам. Если символ оказывается числом, то управление передаётся функции number, которая считывает всё записанное число и устанавливает, что лексема является числом. Далее происходит возврат к функции get.

При нахожденні знака препинання програма заходить в функцію `index`. Здесь полностью считываются все идущие подряд знаки препинания, после чего происходит возврат к функции `get`. При считывании малой буквы программа заходит в функцию `word`. В начале этой функции содержится описание зарезервированных слов, которые будут являться ключевыми. Этими словами являются рабочие слова языка программирования («`if`», «`while`», «`return`», «`print`», «`read`», «`int`», «`real`», «`const`», «`then`», «`do`»), предупредительное слово («`robot`»), слово окончания введения команды («`end`»), а также названия движений робота («`rotate`», «`move`») и его звеньев («`base`», «`shoulder`», «`elbow`», «`wrist`», «`gripper`»). Каждому из этих слов присваивается идентификатор. Ключевое слово устанавливается в `false`. После этого слово считывается целиком. Далее производится проверка, присутствует ли данное слово в зарезервированных, если да, то ключевое слово устанавливается в `true`, и лексеме присваивается соответствующее значение идентификатора. Если же слово отсутствует в списке в зарезервированных, лексеме присваивается значение соответствующее этому случаю.

Дополнительным компонентом разработанного программного обеспечения является подсистема, реализующая лексический и синтаксический анализаторы. В процессе лексического анализа определяются глобальные переменные `lex` – для распознанной лексемы, `lval` – значение лексемы, `nst` – номер строки программы, которая содержит лексемы. Различают лексемы: служебные слова «`if`», «`while`», «`return`», «`print`», «`read`», «`begin`», «`end`», «`int`», «`real`», «`const`», «`then`», «`do`», «`robot`», «`rotate`», «`base`», «`move`», которые кодируются соответственно константами `IFL`, `WHILEL`, `RETRL`, `PRITL`, `READL`, `BEGINL`, `ENDL`, `INTL`, `REALL`, `CONSTL`, `THENL`, `DOL`, `ROBOTL`, `ROTATEL`, `BASEL`, `MOVEL`, знаки операций и разделители, закодированные символическими константами `'(`, `)`, `';`, `':`, `';`, `'='`, `'+`, `'-`, `'*`, `'/'`, `'%`, `'.`, целое без знака с кодом `NUMB` и значением `lval`, идентификатор с кодом `IDEN` и значением, которое является показателем таблицы идентификаторов `TNM`.

Синтаксис языка описывается расширенными грамматиками с использованием символов [1, 2] в регулярных выражениях в правых частях правил. Если `r` – регулярное выражение, которое задаёт множество `R`, то `[r]` – регулярное выражение для множества $R \cup \{\varepsilon\}$, где `[ε]` – пустая цепочка. Терминальные символы расширенной грамматики – это лексемы, коды которых записываются маленькими буквами. Не терминалами являются последовательности букв `PROG`, `DCONST`, `DVARB`, `DFUNC`, `CONS`, `PARAM`, `BODY`, `STML`, `STAT`, `EXPR`, `FCTL`, которые обозначают синтаксические конструкции: программу, определение констант, переменные функций, константу, параметры, тело, список операторов, оператор, выражение, список выражений.

Данная грамматика допускает синтаксический LA-анализ. В разрабатываемом интерпретаторе синтаксический анализ ведётся до появления первой

ошибки, после чего печатается сообщение «ошибка синтаксиса», и работа завершается. В процессе анализа используется функция `exam(lx)`, которая проверяет, имеет ли код следующая лексема. Если да, то вводится следующая лексема, если нет – анализ завершается после выявления ошибки.

Обработка полученной в результате голосового формирования команды обеспечивается функцией `body`.

После выполнения функции `get`, функция `OnTextRecognized` передаёт управление функции `prog`. Эта функция проверяет вид лексемы и передаёт управление соответствующим ей функциям. С точки зрения формирования голосовой команды наибольший интерес представляет функция `dfunc`. Она вызывает выполнение функции `body`, в которой непосредственно определяется правильность задания порядка голосовой команды, для чего также используется функция `exam`, проверяющая соответствует ли данная лексема необходимой. Если команда, заданная голосом, соответствует правилам задания команд, то управление передаётся в функцию `RunJoint`, которая обеспечивает выполнение команды роботом.

При постановке задачи разработки программного обеспечения важное значение имеет знание механики работы манипулятора промышленного робота. Важным этапом проектирования манипулятора является выбор его параметров (мощность, передаточное число, оптимальное распределение передаточного числа по ступеням редуктора).

Проблема состоит в выборе электропривода с учётом следующих факторов:

- требуемые динамические свойства при пуске;
- торможение и изменение нагрузки;
- диапазон регулирования скорости;
- вид требуемой механической характеристики двигателя необходимой точности поддержания заданного режима;
- частота включения приводного механизма.

При этом необходимо произвести анализ динамических возможностей двигателя с учётом требуемых внешних нагрузок. Эта задача, как правило, неоднозначна, т.к. с одной стороны необходимо ограничивать величину ускорений на этапах разбега и выбега, а с другой – обеспечить оптимальное быстрое действие.

Анализ работ [3] показал, как правило, приведенные решения рассматривают применение, в качестве приводных, двигателей постоянного тока, т.к. асинхронные двигатели, из-за сложности и высокой стоимости электронных преобразователей, не имели широкого распространения. В [4] рассматривается вопрос выбора привода с асинхронным двигателем, с учётом его механической характеристики.

Задача решается при следующих допущениях:

- момент инерции движущихся масс I_c постоянный (=const);
- момент статических сопротивлений T_c постоянный (=const);
- передаточное число постоянно на всех режимах движения;

– угловое ускорение изменяется линейно.

Разработана методика определения параметров привода:

- угловой скорости установившегося движения вала манипулятора;
- определения мощности электродвигателя;
- передаточного отношения редуктора.

Использование голосового ввода информации может моделироваться не только на реальном оборудовании, но и на его модели. Наибольший интерес представляет моделирование перемещений робота и его звеньев. Модель робота удобно представить состоящей из простых геометрических поверхностей (примитивов). Математическое моделирование системы координат осуществляется таким образом, чтобы начальная точка моделирования примитива совпадала с началом координат. При этом необходимо учитывать преобразования системы координат.

Для выполнения видовых преобразований системы координат необходимо задать точку наблюдения и направление наблюдения в мировой системе координат. При этом необходимо учитывать, что из точки наблюдения можно видеть точки объектов только внутри некоторого конуса, вершина которого совпадает с точкой наблюдения, а её ось – с направлением наблюдения.

Все процедуры сдвига начала координат, поворота и т. п. достигаются с помощью аффинных преобразований – путём умножения вектора (матрицы-столбца), описывающего трёхмерное пространство, на матрицу поворота, сдвига и т. п.

Любое аффинное преобразование в трехмерном пространстве может быть представлено в виде суперпозиций вращений, растяжений, отражений и переносов. Следовательно, можно отдельно осуществить перенос системы координат, её поворот относительно необходимой оси или нескольких осей, растянуть или сжать, зеркально отразить относительно точки, оси, плоскости. Каждая точка пространства (кроме начальной) может быть задана четвёркой одновременно не равных нулю чисел (hx, hy, hz, h); эта четвёрка определена однозначно с точностью до множителя [5, 6]. Особенности моделирования описаны в [7].

Главное окно программы и подсистемы ввода информации приведено на рис. 2.

Кнопка “StartControl” осуществляет подключение робота к управлению от персонального компьютера. “StopControl” – отключает робот от персонального компьютера. Переключатели “Manual Ctrl” и “Angle Ctrl” позволяют выбрать либо пошаговое перемещение робота, либо перемещение по заданным углам соответственно. Угол поворота задаётся числом в специальном элементе управления.

Кнопки «Base», «Shoulder», «Elbow», «WristRotate», «WristDown» и «Gripper» позволяют выбрать звено робота, которое необходимо повернуть. Для инициализации движения или задержки существует кнопка «Move Init/ Move Delay».

В окне можно выбрать скорость перемещения робота с помощью элемента “SliderCtrl”, значение

скорости будет указываться в элементе “Edit” (в данном случае скорость равна семи). Кнопка “OpenGL” вызывает окно, в котором происходит моделирование работы робота. Кнопка “VoiceControl” вызывает окно распознавания, в котором происходит непосредственное задание команды и её проверка на соответствие существующим лексемам, оно приведено на рис. 3.

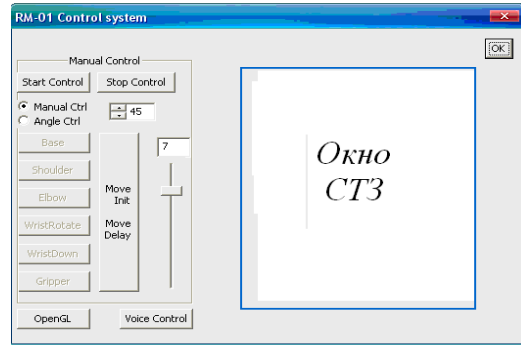


Рисунок 2 – Главное окно программы

При работе программы в режиме распознавания используются XML-файлы как стандартное средство описания грамматик для библиотеки Microsoft Speech API 5.1.

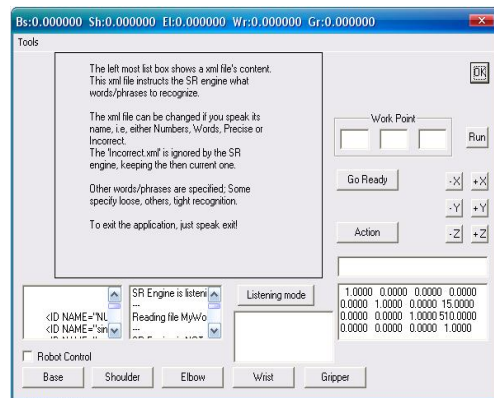


Рисунок 3 – Окно распознавания

Набор управляющих команд робота представляется набором грамматик, описанном в файле формата XML.

При составлении библиотеки слов, которые могут быть использованы в тексте программы работы робота, указаны следующие идентификаторы: «words» (ключевые слова), «switch» (переключение типа выполняемого действия), «joint» (название звена манипулятора), «NUMBERS» (числа), «SNUMBERS» (числа, произносимые одним словом), «DNUMBERS» (двухзначные числа), «HNUMBERS» (трёхзначные числа), «TNUMBERS» (четырёхзначные числа). Наличие такого количества разделов, описывающих числа, объясняется особенностями английского языка. Числа до двадцати являются однословными, то есть могут быть заданы посредством произнесения одного слова. Такие числа, как «двадцать один», «двадцать два» и так далее, в своём составе содер-

жат два слова, первое из которых обозначает десятки, а второе – число от одного до девяти. Трёхзначные числа, например «сто двадцать пять», в английском языке произносятся как «одна сотня двадцать пять». Таким образом, это число включает в себя на первом месте число от одного до девяти, показывающее количество сотен, за ним идёт непосредственно само слово «сотня» («hundred»), далее либо однословное число (от одного до двадцати), либо двухсловное число. Аналогично можно представить числа от тысячи до двадцати тысяч. Число «тысяча сто двадцать пять» (четыре слова в русском языке) в английском варианте будет звучать как «одна тысяча одна сотня двадцать пять». Здесь на первом месте стоит число обозначающее количество тысяч (от одного до двадцати), далее само слово «тысяча» («thousand») и затем всё аналогично представлению трёхзначного числа.

В разделе «words» указаны слова «robot» (начало команды), «end» (окончание команды), «exit» (выход из программы). В разделе «switch» представлены слова, обозначающие вид действия «rotate» (поворот), «move» (перемещение). Раздел «joint» включает названия звеньев «base» (основание), «shoulder» (плечо), «elbow» (локоть), «wristrotate» (поворот кисти), «wristdown» (подъём/опускание кисти) и «gripper».

Раздел «NUMBERS» является структурированным, он включает в себя разделы «SNUMBERS», «DNUMBERS», «HNUMBERS», «TNUMBERS», каждый из которых имеет свою собственную структуру. «SNUMBERS» включает в себя простые числа, «DNUMBERS» – десятки и простые числа («SNUMBERS»), «HNUMBERS» – сотни и «DNUMBERS», «TNUMBERS» – тысячи и «HNUMBERS»:

«SNUMBERS» = <однословные числа> = 0, 1, ..., 20;
 «DNUMBERS» = <двухсловные числа> = <десятки> + «SNUMBERS», <десятки> = 10, 20, ..., 90;
 «HNUMBERS» = <трёхзначные числа> = «SNUMBERS» + «hundred» + «DNUMBERS»
 «TNUMBERS» = <четырёхзначные числа> = «SNUMBERS» + «thousand» + «HNUMBERS».

INFORMATION VOICE INPUT DIGITAL PROCESSING SOFTWARE DEVELOPMENT

I. Nevlyudov, I. Perepelitsa, S. Milyutina

Kharkiv National University of Radio Electronics
 prosp. Lenina 14, Kharkiv, Ukraine. E-mail: tapr@kture.kharkov.ua

D. Mospan

Kremenchuk Mykhailo Ostrohradskyi National University
 vul. Pershotravneva 20, Kremenchuk, 39600, Ukraine.

In this article, the authors have suggested software introduction of the method of information voice input during robotic industrial control. The voice input information subsystem structure is considered in the general robot control CAD programs for RM-01. The key classes implementing voice input information in the form of control voice commands are described. The functions that provide input information, its processing, verification and implementation are considered. It is also presented the user interface, namely the main window and the recognition window. Language and structure of the robot control library for the possible words are given. The developed software conduces to design efficiency enhancement of technological assembly procedures in robotic production through the interactive creation of applied software for assembly robot, changeover time shorten, and cut cost of the mentioned processes design.

Key words: voice control, robot, library, lexeme.

ВЫВОДЫ. Таким образом, в данной статье рассмотрены главные положения создания программного обеспечения для реализации метода голосового ввода информации. Разработанная прикладная программа позволяет упростить процесс, а, следовательно, повысить скорость, проектирования технологических процессов сборки роботизированного производства путём интерактивного формирования прикладных программ сборочного робота. Это приводит к сокращению времени переналадки оборудования, повышению точности задания необходимы перемещений робота, снизить стоимость проектирования технологических процессов роботизированного производства.

Особую благодарность авторы выражают кафедре технологии автоматизации производства РЭС и ЭВС, на базе которой проводились исследования.

ЛИТЕРАТУРА

1. Проценко В.С., Чаленко П.Й., Ставровський А.Б. Техніка програмування мовою Сі. – К.: Либідь, 1993. – 224 с.
2. Шилдт Герберт. МFC: основы программирования. – К.: Издательская группа ВНУ, 1997. – 560 с.
3. Артоболевский И.И. Теория механизмов и машин. – М.: Наука, 1975. – 640 с.
4. Выбор параметров привода манипулятора / С.С. Милютина, Н.В. Милютина // Технология приборостроения. – 2003. – № 1. – С. 22–23.
5. Тихомиров Ю.В. OpenGL. Программирование трехмерной графики. – СПб.: БХВ-Петербург, 2002. – 304 с.
6. OpenGL: официальное руководство программиста / Мейсон Ву, Джеки Нейдер, Том Девис, Дейв Шрайнер.; пер. с англ. – СПб.: ООО «ДиаСофтЮП», 2002. – 592 с.
7. Разработка программного обеспечения для трёхмерного моделирования робота / И.Ш. Невлюдов, С.С. Милютина // Технология приборостроения. – 2004. – № 2. – С. 5–7.
8. Интеллектуальное проектирование технологии роботизированной сборки / И.Ш. Невлюдов, А.М. Цымбал, С.С. Милютина. – Харьков, 2010. – 207 с.

REFERENCES

1. Protsenko, V.S., Chalenko, P.I., Stavrovskiy, A.B. (1993), *Tekhnika programuvannia movoyu Si* [Programming technique on C language], Lybid, Kyiv, Ukraine.
 2. Shcildt, Gerbert (1997), *MFC: fundamentals of programming*, Publishing group BHV, Kyiv, Ukraine.
 3. Artobolevsky, I.I. (1975), *Teoriya mekhanizmov i mashin* [Theory of mechanisms and machines]nauka, Moscow, USSR.
 4. Miliutina, S.S., Milyutina, N.V. (2003), "Selection of manipulator drive parameters", *Tekhnologiya priborostroeniya*, no. 1, pp. 22–23.
 5. Tikhomirov, Yu.V. (2002), *OpenGL. 3D graphics programming*, BKhV-Peterburg, St.-Petersburg, Russia.
 6. Mason, Woo, Nader, Jacky, Davis, Tom, Shriner, Dave (2002), *OpenGL: programmer's official guideline*, DiaSoftYuP, LLC, St.-Petersburg, Russia.
 7. Nevliudov, I.Sh., Milyutina, S.S. (2004), "Software development for robot 3D modeling", *Tekhnologiya priborostroeniya*, no. 2, pp. 5–7.
- Nevliudov, I.Sh., Tsybal, A.M, Milyutina, S.S. (2010), *Intellektualnoe proektirovanie tekhnologii robotizirovannoi sborki* [Smart design of robotic assembly technology], KhNURE, Kharkiv, Ukraine.

Стаття надійшла 22.04.2014.