

ведущей магистерской подготовке в Будапештском техническом университете.

Головными базовыми вузами были утверждены МГТУ "Станкин" и Будапештский технический университет, где организаторами обучения российских студентов стали профессора и заслуженные доктора МГТУ "Станкин" Матьяш Хорват и Пондло Янош.

Для обеспечения работ по программе Минобрнауки РФ на базе МГТУ "Станкин" был образован Международный учебно-научный центр "Мехатроника", руководителем которого был утвержден проф. Кулешов В. С. Одиннадцать выпускников специалистов, осуществленных Будапештским техническим университетом, позволили провести подготовку 170 магистров наук, ориентированных для последующего обучения в аспирантуре либо ведения научно-педагогической деятельности в вузах, направивших их на подобное обучение.

Активно развивались научные исследования и разработки, проводимые в области мехатроники рядом других кафедр МГТУ "Станкин". Так, кафедра д-ра техн. наук проф. Сосонкина В. Л. расширяла работы по созданию перспективных систем числового программного управления мехатронными системами, разрабатывая архитектуру систем типа PCNC.

На кафедре д-ра техн. наук проф. Митрофанова Г. Г. успешно решались проблемы создания компьютеризированных интегрированных производств, основанных на использовании мехатронных средств систем автоматизации технологических процессов в автоматизированных машиностроительных производствах.

Многие кафедры факультета "Механика и управление" возглавляемого проф. Подураевым Ю. В., совместно с отделом "Мехатронные системы и робототехника" Центра Физико-технологических исследований (ЦФТИ) МГТУ "Станкин", руководимого д-ром техн. наук проф. Григорьевым С. Н., подключились к выполнению перспективных проектов в рамках комплексных программ Минобрнауки РФ и Миннауки РФ.

Вышесказанное подтверждает важное значение, придаваемое в МГТУ "Станкин" становлению перспективного направления развития современной науки и техники, какой является "Мехатроника и робототехника". Очевидна также готовность университета в рамках данного направления к участию в научных исследованиях и Межвузовской программе подготовки инженерных и научных кадров совместно с институтами Российской академии наук.

Публикуемые ниже статьи в основном базируются на докладах, сделанных на IV конференции по Конструкторско-технологической информатике, проведенной в связи с 70-летием МГТУ "Станкин". При этом было решено опубликовать материалы, посвященные направлениям, способствующим созданию мехатронных систем, в частности разработкам в области CAD/CAM систем.

УДК 681.51:(621.01+621.38)

**В. Л. Сосонкин**, д-р техн. наук, проф.,  
**Г. М. Мартинов**, канд. техн. наук, доц.,  
МГТУ "СТАНКИН"

## **Концепция числового программного управления мехатронными системами: реализация логической задачи управления**

*Предложен новый подход к реализации логической задачи числового программного управления мехатронными системами. В рамках жизненного цикла логической задачи рассмотрены фазы программирования, интерпретации программы и исполнения. На первой фазе представляется важным применение визуальных средств программирования, которые дают оператору инструмент для графической диалоговой разработки программы управления электроавтоматикой, причем, эта программа интерпретируется в исполняемые C++ коды без компиляции в промежуточный язык. При этом изменяется (в сторону существенно большей эффективности) сама структура математического обеспечения системы логического управления.*

### **Введение**

Логическая задача, являясь разделом математического обеспечения систем управления цикловой электроавтоматикой, реализуется двойкой: программно в рамках системы ЧПУ или с помощью программируемого контроллера. Традиционный контроллер — это специализированный аппарат, оснащенный терминалом в виде персонального компьютера. Возрастание мощности и уровня сервиса персонального компьютера позволяет объединить терминал, программатор и собственно контроллер в рамках единой компьютерной архитектуры с дополнительным модулем ввода-вывода сигналов электроавтоматики. Существует прообраз контроллера, который называют системой ПСС (Personal Computer-Controller, персональный программируемый контроллер). Прогнозируя развитие концепции ПСС, можно отметить следующие ее особенности:

- использование однокомпьютерного варианта с операционной системой Windows NT и расширением реального времени;
- увеличение функций интерфейса оператора за счет многорежимного управления и применения встроенных инструментальных систем программирования и поддержания в реальном времени динамических графических моделей (мнемодиаграмм) управляемого объекта;
- построение терминальной части системы ПСС по типу "виртуального прибора" [1, 2];

- применение визуального программирования электроавтоматики (например, по типу графического языка HighGraph фирмы Siemens [3]) с генерацией C++ кодов исполняемого модуля;
- организация "многопоточного управления (Multi-thread)".

Нам представляется важным применение визуальных средств программирования, которые не только дают оператору инструмент для графической диалоговой разработки программы управления электроавтоматикой, но и порождают исполняемые C++ коды без компиляции в промежуточный язык. При этом изменяется (в сторону существенно большей эффективности) и сама структура математического обеспечения системы РСС. Дальнейшее изложение посвящено именно этой проблеме.

#### Формализм описания циклов электроавтоматики

Воспользуемся формализмом иерархических графов, который удобен для графического описания циклов, в том числе и с помощью инструментальных средств визуального программирования. Иерархический граф состоит из:

- простых вершин-состояний, изображаемых кружками; состояния могут быть статическими или динамическими (выход из статического состояния инициируется извне, тогда как выход из динамического состояния происходит после завершения процесса);
- сложных вершин-состояний, изображаемых двойными (с двойным бордюром) кружками; такие состояния сами по себе являются вложенными графами;
- дуг, отражающих переходы между состояниями любого типа;
- узлов, "разрезающих" дуги, изображаемых темными кружками; узлы фиксируют условия смены состояний любого типа (если дуга исходит из статической вершины-состояния, то узел может одновременно принадлежать другому графу того же или другого уровня иерархии).

Вершинам-состояниям приспаны "этикетки" — имена в прямоугольных рамках. Имя статического состояния имеет структуру СТАТУС\_⟨имя⟩. Имя динамического состояния имеет структуру ПРОЦ\_⟨имя⟩ или структуру ИНИЦ\_⟨имя⟩. Имя узла является признаком одного из следующих типов: команда, инициируемая с панели оператора; флаг завершения вычислительного процесса; сигнал окончания управляемой операции.

Методика описания цикла электроавтоматики включает следующие этапы:

- разработка "первичного автомата", т. е. автомата верхнего уровня иерархии, являющегося по сути диспетчером режимов;
- разработка режима нерегулярных ситуаций (внутреннего режима), который сохраняет корректность состояния управляемого объекта при любых переключениях основных режимов, а

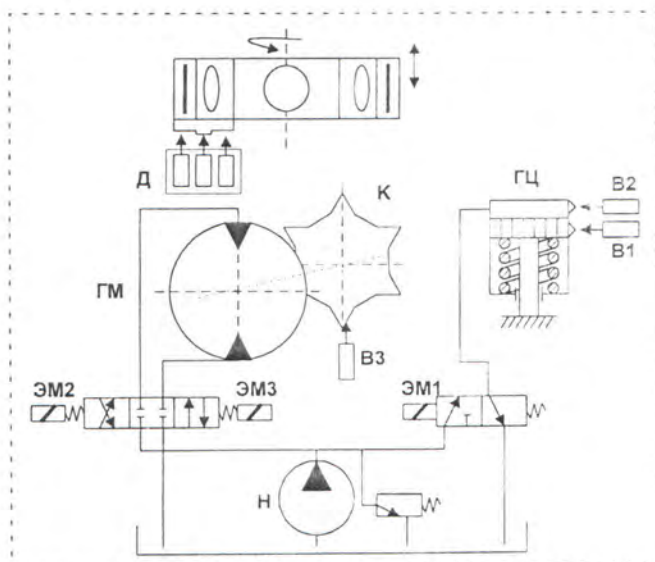


Рис. 1. Схема работы револьверной головки

также гарантирует неизменное состояние объекта, если цикл пассивен;

- выделение параллельно работающих автоматов, действующих в рамках цикла;
- разработка автоматов нижнего уровня иерархии.

Рассмотрим систему управления револьверной головкой токарного станка (рис. 1), которая поддерживает автоматический режим, в рамках которого обеспечивается вызов инструмента любой грани револьверной головки (переход на нужную грань происходит по кратчайшему пути), режим ручного управления (например, поворот револьверной головки на очередную ее грань) и режим нерегулярных ситуаций.

Двухпозиционный гидрораспределитель с электромагнитом ЭМ1 управляет гидроцилиндром ГЦ зажима-разжима револьверной головки; нижнее и верхнее положения контролируются выключателями В1—В2. Поворот головки осуществляется гидромотором ГМ, управляемым гидрораспределителем с электромагнитами ЭМ2—ЭМ3. Прохождение любой грани через возможное (по углу поворота) положение идентифицируется выключателем В3 и кулачком К, соосным с валом гидромотора. Код грани устанавливается многоразрядным датчиком Д.

Воспользуемся нашей методикой описания цикла. На рис. 2 приведен первичный граф (автомат) со всеми сложными состояниями, в том числе: СТАТУС\_НЕРЕГ\_РЕЖ (режим нерегулярных ситуаций), СТАТУС\_АВТОМ\_РЕЖ (автоматический режим), СТАТУС\_РУЧН\_РЕЖ (режим ручного управления). Первичный автомат инициируется узлом-условием (клавишей) начального пуска НАЧ\_ПУСК, а после этого управляет режимными переходами. Узлы-условия АВТОМАТ и РУЧН соответствуют активизации соответствующих режимных управляющих элементов панели оператора.

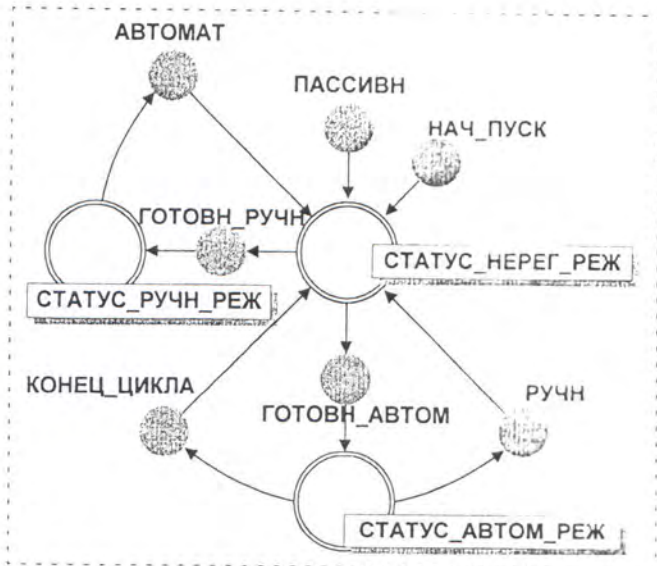


Рис. 2. Граф первичного автомата

На рис. 3 раскрыта сложная вершина-состояние СТАТУС\_НЕРЕГ\_РЕЖ первичного автомата, которая сама по себе является графом режима нерегулярных ситуаций. Вход в граф осуществляется через один из узлов-условий: НАЧ\_ПУСК, АВТОМАТ, РУЧН (см. рис. 2). Затем инициируется сложная динамическая вершина-состояние ПРОЦ\_ИДЕНТИФ, в рамках которой идентифицируется текущее состояние револьверной головки. Если головка вращается (узел-условие ВРАЩ), то вызывается алгоритм определения совпадения с корректным (по углу) положением грани (в динамическом состоянии ПРОЦ\_ОПРЕДЕЛ\_СОВПАД), в корректном положении формируется команда торможения (узел-условие КОМ\_ТОРМ). Если головка разжата или неподвижна (узлы-условия РАЗЖАТ и СТОИТ), то вновь вызывается тот же алгоритм в вершине-состоянии ПРОЦ\_ОПРЕДЕЛ\_СОВПАД, в зависимо-

сти от результата работы которого возбуждаются команды зажима (КОМ\_ЗАЖ) или вращения (КОМ\_ВРАЩ). Наконец, если в качестве результата идентификации установлено зажатое состояние головки (узел-условие ЗАЖАТ), то система переходит в состояние подготовки к управлению (ПРОЦ\_ГОТОВН). Подготовка завершается получением условия готовности ручного или автоматического режимов.

Рассмотрим параллельные процессы, в которых используются собственные двигатели: зажим-разжим головки, вращение-останов головки (принцип выделения состоит в наличии собственного двигателя). Процессы отображаются независимыми графами, показанными на рис. 4, 5. Они инициируются в сложном состоянии ПРОЦ\_ИДЕНТИФ графа режима нерегулярных ситуаций.

Инициация зажима-разжима осуществляется в состоянии ИНИЦ\_1 графа автомата зажима-разжима. Инициация устанавливает условия:

- корректный разжим (КОРР\_РАЗЖ) — разжатое состояние головки в корректном положении ее граней по углу поворота (подобное условие инициирует процесс зажима в состоянии ПРОЦ\_ЗАЖ);
- корректный зажим (КОРР\_ЗАЖ) — зажатое состояние головки в корректном положении ее граней по углу поворота (подобное условие инициирует переход в статическое состояние зажима СТАТУС\_ЗАЖ);
- некорректный зажим (НЕКОР\_ЗАЖ) — зажатое состояние головки в неправильном положении ее граней по углу поворота по причине сбоя (подобное условие инициирует процесс разжима головки в состоянии ПРОЦ\_РАЗЖ);
- некорректный разжим (НЕКОР\_РАЗЖ) — разжатое состояние головки в неправильном положении ее граней по углу поворота (подобное условие инициирует переход в статическое состояние разжима СТАТУС\_РАЗЖ).

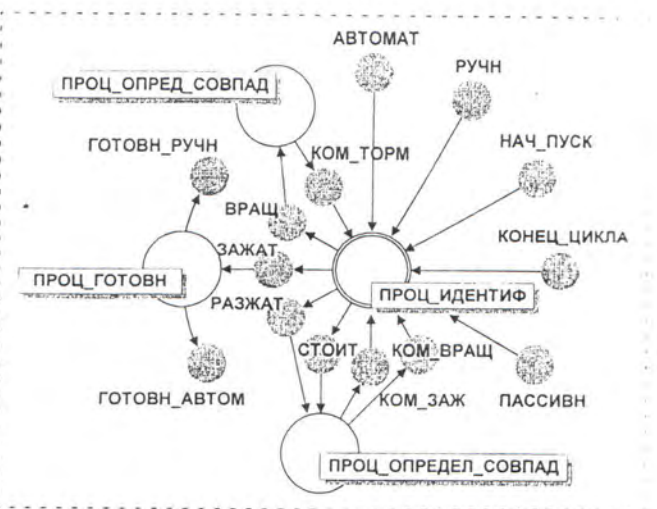


Рис. 3. Граф режима нерегулярных ситуаций

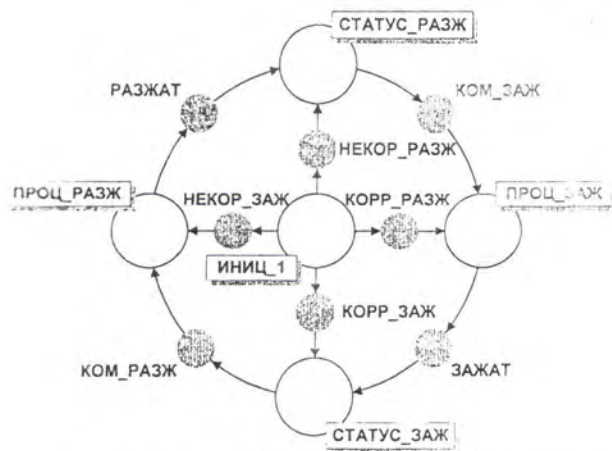


Рис. 4. Граф автомата зажима-разжима

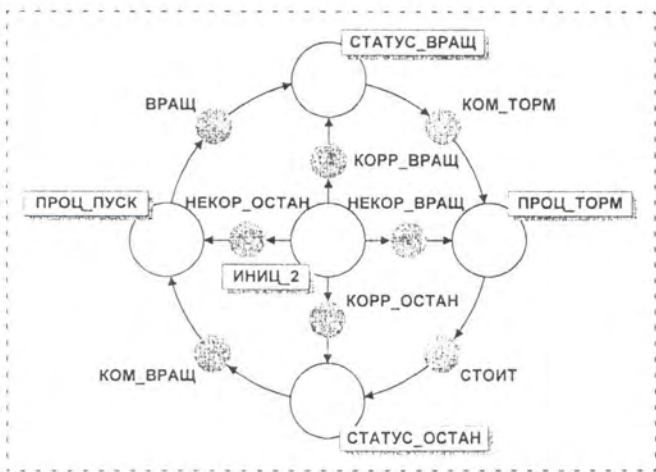


Рис. 5. Граф автомата вращения-останова

Инициация вращения-останова осуществляется в состоянии ИНИЦ\_2 графа автомата вращения-останова. Инициация устанавливает условия:

- некорректное вращение (НЕКОР\_ВРАЩ) — включенное состояние вращения гидромотора ГМ поворота головки при ее зажатом состоянии по причине сбоя (подобное условие инициирует процесс торможения головки в состоянии ПРОЦ\_ТОРМ);
- корректный останов (КОРР\_ОСТАН) — остановка головки в корректном положении ее граней по углу поворота (подобное условие определяет переход в статическое состояние СТАТУС\_ОСТАН);
- некорректный останов (НЕКОР\_ОСТАН) — остановка головки в неправильном положении ее граней по углу поворота (подобное условие инициирует процесс пуска вращения ПРОЦ\_ПУСК);
- корректное вращение (КОРР\_ВРАЩ) — нормальное вращение головки в ее разжатом состоянии (подобное условие определяет переход в статическое состояние СТАТУС\_ВРАЩ).

Совместное рассмотрение двух графов на рис. 4, 5 показывает, что система обеспечивает автоматический выход из любых некорректных ситуаций. Управляя условиями выхода из статических состояний, можно построить любой цикл автоматического или ручного управления.

#### Инструментальная поддержка визуального программирования циклов электроавтоматики

Однотипность скелета исполняемого кода циклов позволила нам разработать инструментальную систему визуального проектирования, генерирующую исполняемые C++ исходные файлы. Конкретный граф вводят с панели интерфейса программиста, которая предлагает набор графических примитивов: простую вершину-состояние, сложную вершину-состояние, дугу, узел дуги. Свойства примитивов (имена, типы вершин-состояний

и др.) задают в диалоговом режиме на "странице свойств" (property page). Функции визуального проектирования обеспечивают:

- многоуровневое вложение графов с работой на каждом уровне в отдельном окне;
- выполнение групповых операций (выделение фрагмента графа, удаление, копирование, перемещение фрагментов в разных позициях и на разных уровнях);
- сохранение-загрузку проекта или фрагмента;
- импорт одного проекта в другой;
- документирование проекта и генерацию отчетов;
- генерацию исходного кода для последующей компиляции;
- верификацию графа на уровне проектирования, моделирование и отладку циклов.

Применение инструмента визуального проектирования многократно повышает производительность разработчика, позволяет создавать сложные циклы электроавтоматики, реализация которых без инструментальной поддержки проблематична.

#### Генерация инструментальной системой C++ кодов исполняемых модулей циклов электроавтоматики

Диаграмма классов исполняемых модулей циклов электроавтоматики в нотации Booch [5] приведена на рис. 6. Диаграмма отражает состав и взаимоотношения классов. В соответствии с соглашением нотации каждый класс изображают в виде облака. Имя класса начинается с буквы "C" (например, CNcsState). Линии и стрелки показывают отношения между классами (а также структурами, типами и объединениями).

Прототипом простого или сложного состояния служит класс CNcsState. Этому классу сопоставлена "этикетка" типа CLabel, которая сохраняется в *m*-поле *m\_Label*; это показано отношением включения, в котором связь начинается закрашенным кружком и завершается закрашенным квадратом. Класс CNcsState хранит таблицу возможных пере-

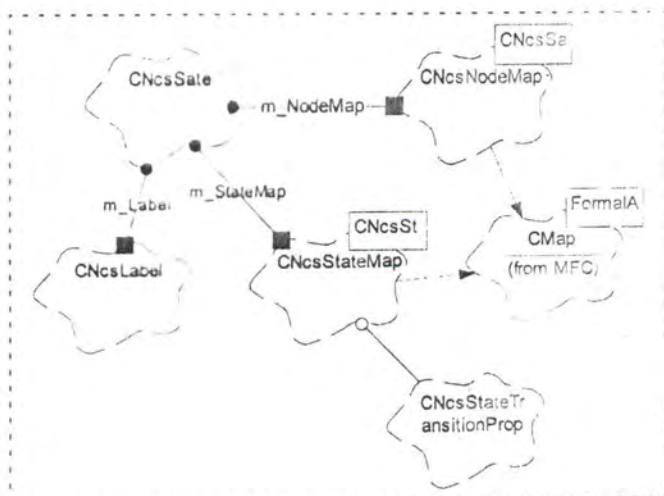


Рис. 6. Диаграмма классов в нотации Booch

ходов в  $m$ -поле `m_NcsStateMap`. Функциональные возможности таблицы отражены в реализации (`Instance`) `CNcsStateMap` класса шаблона `CMap` библиотеки MFC, что обозначено штриховой стрелкой. Объект класса `CNcsStateTransitionProp` устанавливает структуру, тип и условия переходов. Класс `CNcsState` содержит указатель на таблицу вложенных состояний в  $m$ -поле `m_NodeMap`. Эта таблица инициализируется и заполняется только для сложных состояний. Таблица вложенных состояний отражена в реализации (`instance`) `CNcsNodeMap` класса шаблона `CMap`.

### Заключение

Жизненный цикл логической задачи управления предполагает программирование, интерпретацию программы и ее исполнение. Современная тенденция характеризуется упрощением первой фазы за

счет визуального программирования, включая инструментальную поддержку, а также объектно-ориентированной реализацией второй фазы.

### Список литературы

1. Соломенцев Ю. М., Сосонкин В. Л., Мартинов Г. М. Построение персональных систем ЧПУ (PCNC) по принципу открытых систем // Информационные технологии и вычислительные системы. 1997, № 3. С. 68—75.
2. Перцовский М. Время распределять системы и время собирать системы // PC WEEK/RE. 1997, № 37 (111). С. 57—61.
3. Otto H.-P., Rath G. State diagrams. A new programming method for Programmable Logic Controllers / Software Engineering for Manufacturing Systems: Methods and CASE tools. Edited by A. Storr and D. Jarvis. 1996, London: Chapman&Hall. P. 27—37.
4. Сосонкин В. Л., Мартинов Г. М., Любимов А. Б. Интерпретация диалога в windows-интерфейсе систем управления // Приборы и системы управления, 1998, № 12. С. 10—13.
5. Selic B., Gullekson G., Ward P. Real-Time Object-Oriented Modelling. Computer Society, 1994. P. 560.

УДК 681.51:(621.01) + 621.38

Ю. В. Илюхин, канд. техн. наук, доц.,  
МГТУ "СТАНКИН"

## Совершенствование систем управления механообрабатывающих технологических роботов на основе концепций мехатроники

*С позиций мехатроники рассмотрены проблемы технологических достаточных робототехнических систем минимальной сложности: предложена стратегия построения систем управления технологических роботов СУ ТР, исследованы вопросы функционального синтеза перспективных вариантов структур управления, описана аппаратно-программная реализация СУ ТР*

### Постановка проблемы

Одной из актуальных проблем современного машиностроения является автоматизация ряда технологических процессов механической обработки деталей, обладающих большими размерами и сложной формой поверхности. Речь идет, прежде всего, об операциях зачистки кромок, удаления облоя и снятия заусенцев с поверхности отливок, полирования, фрезерования, сверления и механической резки. Помимо традиционного использования для этой цели специальных станков, перспективным представляется применение технологических роботов (ТР) [1]. Преимущество ТР, имеющих многостепенной манипулятор, состоит в их относи-

тельно небольшой металлоемкости, значительной маневренности и способности обрабатывать крупногабаритные детали со сложной формой поверхности, в том числе их труднодоступные участки при минимальном количестве установок. Однако манипулятор ТР обладает существенной упругой податливостью и относительно невысокими значениями частот собственных колебаний, что осложняет внедрение ТР на операциях, связанных с силовым взаимодействием инструмента и обрабатываемой детали. Возможность расширения области применения ТР в автоматизированном машиностроении, повышение их эффективности и конкурентоспособности связаны с необходимостью совершенствования методологии создания рациональных структур и алгоритмов современных систем компьютерного управления механическим движением. Такие системы должны удовлетворять требованиям конкурентоспособности благодаря высокой точности исполнительных движений при стабильности динамических показателей и ограниченной технической сложности.

### Стратегия построения систем управления технологических роботов с позиций мехатроники

При создании конкурентоспособных систем управления (СУ) исполнительными движениями ТР целесообразно основываться на идеях и принципах мехатроники, отражающих современные тенденции развития цифровых средств управления механическим движением.

Для увеличения производительности механообрабатывающего ТР, представляющего собой специфическую многокоординатную технологическую машину, его СУ должна обеспечивать пере-