___ В.С. Гусельников, А.Л. Комисаренко, М.М. Шальнов

Методические рекомендации по выполнению СРС

МОДЕЛИРОВАНИЕ ПРИБОРОВ, СИСТЕМ И ПРОИЗВОДСТВЕННЫХ ПРОЦЕССОВ

Под ред. к.т.н., доцента Е.И. Яблочникова

ПРИЛОЖЕНИЕ II

Санкт-Петербург 2008

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ



В.С. Гусельников, А.Л. Комисаренко, М.М. Шальнов

Методические рекомендации по выполнению СРС

МОДЕЛИРОВАНИЕ ПРИБОРОВ, СИСТЕМ И ПРОИЗВОДСТВЕННЫХ ПРОЦЕССОВ

Под ред. к.т.н., доцента Е.И. Яблочникова

ПРИЛОЖЕНИЕ II



Санкт-Петербург

2008

В.С. Гусельников, А.Л. Комисаренко, М.М. Шальнов. Моделирование приборов, систем и производственных процессов / Приложение II. Методические рекомендации по выполнению СРС. Под ред. к.т.н., доцента Е.И. Яблочникова – СПб: СПбГУИТМО, 2008. – 96 с.

Данные методические рекомендации служат для практического ознакомления студентов с современными средствами виртуального моделирования изделий и процессов в приборостроении и машиностроении. Приводятся краткие основные положения по планированию, анализу и оптимизации технологических операций сборки с помощью программного комплекса DELMIA DPM Assembly. Рассматриваются современные методики моделирования поведения конструкции в условиях эксплуатации в среде Nastran/Patran. Описываются характеристики системы IMSpost в области разработки постпроцессоров для станков с ЧПУ. Приводится описание системы Vericut в области построения виртуальных моделей станков с ЧПУ. Рассматривается реализация процесса проектирования технологических процессов в PDM-системе Smarteam.

Данные рекомендации предназначены для магистров, обучающихся по направлению подготовки 200100 – Приборостроение по магистерским программам «Технологическая подготовка производства приборов и систем», «Управление жизненным циклом приборов и систем» и «Проектирование интегрированных систем технической подготовки производства приборов и систем».

Рекомендовано УМО по образованию в области приборостроения и оптотехники в качестве учебного пособия для студентов высших учебных заведений, обучающихся по направлению подготовки 200100 – Приборостроение.

В 2007 году СПбГУ ИТМО стал победителем конкурса инновационных образовательных программ вузов России на 2007–2008 годы. Реализация инновационной образовательной программы «Инновационная система подготовки специалистов нового поколения в области информационных и опти-



ческих технологий» позволит выйти на качественно новый уровень подготовки выпускников и удовлетворить возрастающий спрос на специалистов в информационной, оптической и других высокотехнологичных отраслях экономики.

©Санкт-Петербургский государственный университет информационных технологий, механики и оптики, 2008

© В.С. Гусельников, А.Л. Комисаренко, М.М. Шальнов, Е.И. Яблочников.

Оглавление

Самостоятельная работа «Верификация технологических операций	
сборки»	.6
Цель работы	6
Введение	6
Задание	8
Этап 1. Формирование рабочего места	8
Этап 2. Создание библиотеки стандартных технологических	
сборочных операций1	2
Этап 3. Формирование перечня технологических операций	
(последовательности сборки)1	3
Этап 4. Задание траектории перемещения компонентов и анализ	
пересечений компонентов1	5
Этап 5.Формирование файла отчета1	7
Самостоятельная работа «Применение метода конечных элементов в	
системах MSC Nastran и Patran»1	9
Компьютерные технологии инженерного анализа и виртуального	
моделирования1	9
MSC.Patran - интегрированная среда моделирования, анализа и	
проектирования2	1
MSC.Nastran - расчет и оптимизация конструкций22	2
Задание2	3
Создание базы данных2	3
Импорт геометрии из CAD-системы24	4
Создание поверхности с сеткой конечных элементов24	4
Создание материала	7
Задание физических Свойств2	8
Моделирование нагрузок и граничных условий	9

Создание входного файла для MSC.Nastran	31
Передача в среду MSC.Patran результата решения задачи для	
обработки постпроцессором	34
Обработка результата постпроцессором	35
Самостоятельная работа «Построение виртуальных моделей станков	c
ЧПУ»	36
Введение	36
Система Vericut	37
Верификация (Verification)	38
Оптимизация (OptiPath)	39
Реалистичная имитация работы станка (Machine Simulation)	39
Общая структура системы Vericut	41
Начало работы	43
Интерфейс Vericut	43
Новый проект	44
Содержание отчёта	54
Самостоятельная работа «Разработка постпроцессоров для станков с	
ЧПУ»	54
Введение	54
Язык АРТ	54
Средства автоматизации разработки постпроцессоров. Система	
IMSpost	56
Диалоговые окна	60
Макро язык и макросы	61
Арифметические операции	63
Математические функции	63
Операторы присваивания	64
Классы переменных	65
Работа с макросами	67

Просмотр макросов	68
Редактирование макросов	68
Удаление макросов	69
Создание макроса	69
Импорт макроса	70
Поиск Макросов	72
Модель станка и её использование	73
Содержание отчёта	76
Самостоятельная работа «Разработка технологического процесса	
изготовления детали в PDM системе SMARTEAM»	76
Разработка технологических процессов в PDM-системе	76
Задание	81
Создание объекта «деталь»	83
Создание заготовки	84
Проектирование «расцеховок»	85
Проектирование «сквозного» ТП	86
Проектирование операций и переходов	86
Наполнение базы данных инструментов и технологического	
оборудования	89
Проектирование технологической документации	89
Отображение дерева ТП со всеми логическими связями	91
Приложение 1	92
Приложение 2	93
Список литературы	94

Самостоятельная работа «Верификация технологических операций сборки»

Цель работы

С помощью программного комплекса DELMIA DPM Assembly провести построение и верификацию технологических операций сборки изделия и ознакомиться со способами:

- задания типовых сборочных операций в системе DELMIA DPM Assembly;
- учета используемого оборудования и приспособлений;
- представления процессов сборки (схем сборки);

В результате выполнения самостоятельной работы студенты должны приобрести навыки разработки технологических процессов сборки в среде DELMIA DPM Assembly.

Введение

В настоящее время предприятия должны постоянно повышать производительность для того, чтобы успешно конкурировать на рынке. Для этого требуются сокращение сроков поставок, уменьшение текущих расходов, оптимальное использование производственных мощностей, оптимизация материальных и информационных потоков. В то же время полная автоматизация, определенная ранее как конечная цель, заменяется гибридной схемой, которая автоматизирована частично, вследствие чего она легко приспосабливаема к производственной системе.

Решением задач исследования, планирования и оптимизации процессов изготовления и обслуживания изделий на предприятии до начала их реального производства занимается система DELMIA.

Программный комплекс DELMIA решает следующие основные задачи:

- планирование производственных процессов;
- планирование производственных помещений;
- анализ эргономичности производственных процессов;
- оптимизация производственных процессов;
- контроль качества, нормирование;
- симуляция работы конвейеров;
- симуляция работы промышленных роботов и роботизированных ячеек;
- симуляция и верификация технологических процессов сборки и др.

В рамках данного пособия будет рассмотрен модуль DELMIA DPM Assembly, который позволяет моделировать технологические процессы сборки, проводить верификацию сборочного процесса, анализировать собираемость изделия, оценивать производственные ресурсы, используемое оборудование, инструмент и приспособления, рассчитывать временные характеристики технологических операций сборки.

Виртуальное моделирование позволяет оптимизировать используемые ресурсы и схему размещения оборудования в цехе или на производственном участке. Пользователь имеет возможность манипулировать как составом, так и взаимным расположением производственных компонент, быстро изменяя и совершенствуя схему их размещения. Сочетание DPM Assembly с другим модулем – DPM Work Instruction дает возможность генерировать так называемые электронные рабочие инструкции. Эти инструкции могут включать в себя симуляции различных этапов сборки и использоваться рабочими производственных цехов и участков.

Система DELMIA использует единую модель данных PPR (Process, Product, Resource), которая объединяет всю необходимую информацию об изделии (продукте), процессах изготовления или обслуживания, а так же о используемых ресурсах. В единой информационной среде объединяются данные, используемые инженерами-конструкторами, инженерами-технологами, технологами, инженерами по организации производства, специалистами по планированию производства. Любая единица информации,

важная в контексте интегрированного инжиниринга продукта и процесса, может быть найдена в модели данных PPR.

DELMIA полностью интегрируется с системой CATIA, предназначенной в первую очередь для проектирования изделия, а также с PDMсистемами (напр., ENOVIA Smarteam).

Задание

В данной работе студентам необходимо самостоятельно выполнить верификацию и оптимизацию технологических операции сборки. Для этого необходимо следовать следующему плану.

Этап 1. Формирование рабочего места.

До начала непосредственного задания технологического процесса сборки необходимо сформировать рабочее место и указать используемые ресурсы. Ресурсы – любые элементы, которые будут использоваться на конкретном рабочем месте при сборке изделия. Ресурсы могут храниться как в библиотеках, так и локально на компьютере пользователя (это определяется стандартами организации работ на конкретном предприятии).

1. Запустите систему DELMIA.

2. Перейдите к модулю DPM Assembly Process Simulation (меню Start / Digital Process for Manufacturing / Assembly Process Simulation), рис. 1.



Рис. 1. Запуск DPM Assembly Process Simulation

- 3. В данном задании ресурсы будут добавляться из каталога.
- 4. Щелкните на иконке *Catalog Browser* (рис. 2).



Рис. 2. Обращение к каталогу

Нажав на кнопку browse another catalog, выберите каталог .../Dassault Systemes / B17 / intel_a / startup / components / facility / furniture.catalog (рис. 3).

-9-44	A68
- ALEFT	ACMA
- S cons	ADEPT
- C CADIEN	A 0.005
C TANK	
- 3 m	Egy comau -
- S CONSACT	DATHEN
HOTOHAN	DUERR
- ST REWISSING	FANUC
- Col ens	(M) (2)F
чанана.	
Filter:	- A C

Рис. 3. Режим поиска каталога элементов

Из открывшегося каталога выберите стол work_bench 72x36x34.

Добавьте два таких стола и расположите их относительно друг друга, как показано на рисунке ниже.



Рис. 4. Подготовка виртуального рабочего места

Из каталога *My_Catalog.catalog* (находится в папке с данными для самостоятельной работы) добавьте стул (*chair*) и тележку для инструментов (*toolcart*).

Добавьте используемый инструмент – гаечный ключ (*wrench*) и отвертку (*screwdriver*). Расположите компоненты относительно друг друга, как показано на рис. 5.

Далее необходимо добавить изделие (продукт) в созданное окружение.

Для этого воспользуйтесь командой *Insert Product* в меню *Insert* или на панели инструментов.



Рис. 5. Выбор инструментов сборки

В открывшемся окне проводника Windows выберите файл 3.5 *HP* engine. *CATProduct* в папке с данными для самостоятельной работы (*Project Data/ Product*).

Расположите изделие (3.5 *HP engine*.*CATProduct*) на рабочем столе (рис. 6).



Puc. 6. 3.5 HP engine.CATProduct

Сохраните созданное рабочее место в папку, указанную преподавателем.

Этап 2. Создание библиотеки стандартных технологических сборочных операций

В процессе работы используется *Process Library* – библиотека стандартных операций, которая включает в себя описание всего множества операций, используемых в процессе замены компонента. Данная библиотека может формироваться как заранее, так и в процессе работы. Структура и иерархия, а также параметры конкретных операций должна определяться существующими на предприятии нормативными документами, стандартами и пр.

Создадим новую библиотеку операций. Меню File-New-Process Library.

Выберите команду *Create New Activity* assembly" (Сборка двигателя).



для создания

и назовите "Engine

Выберите команду Create New Activity Subtype подтипов операций.

Создайте следующие подтипы (операций):

- Intake Valve Assy (установка впускного клапана),
- Carburetor Assy (установка карбюратора),
- Air Filter Assy (установка воздушного фильтра),
- Crankshaft Assy (установка коленчатого вала),
- *Piston Assy* (установка поршня),
- Piston Rod Assy (установка поршневого штока),
- *Cam Assy* (установка кулачка),
- Crankcase Assy (установка картера),
- Flywheel Assy (установка маховика),
- *Cover Assy* (установка крышки).



Рис. 7. Технологический процесс сборки

Сохраните вашу библиотеку операций в папку, указанную преподавателем (например, .../ *Project Data / my process library*).

Далее ассоциируем созданную библиотеку с процессом сборки (*Process*). Для этого необходимо воспользоваться инструментом *Insert*

Activity Library и указать *Process* в дереве проекта, после чего выбрать ранее созданную библиотеку.

Это необходимо для того, чтобы при задании операций система предлагала нам список операций из библиотеки.

Этап 3. Формирование перечня технологических операций (последовательности сборки)

Следующим этапом работы является формирование последовательности сборки. Это делается путем добавления технологических операций (*Activity*) в дерево проекта (в ветвь *Process*).

Как правило, последовательность формируется в обратном порядке (т.е. производится разборка изделия), а после этого порядок операций меняется на противоположный.

Для добавления операций воспользуйтесь инструментом Insert Ac-

tivity , щелкните на *Process* и выберите *Engine Assembly* в появившемся диалоговом окне.

Далее с помощью того же инструмента добавьте *Block Assy*, выбрав тип вставки *Add As child* (вставить, как потомка).

Затем добавьте остальные операции, выбирая тип вставки Add As Successor (вставить, как последующий элемент). Последующий элемент (Sacsessor) должен быть связан с предыдущей операцией, а не с Engine Assembly.

Теперь необходимо проверить правильность последовательности операций и при необходимости изменить ее. Сделать это можно с помощью диаграммы планирования с использованием сетевого графика (*PERT chart*).



Щелкните на иконке Open Pert Chart

затем на *Process* в дереве проекта. Откроется сетевой график вашего процесса. Раскройте полностью график, нажав "+" в правом верхнем углу *Engine Assembly*.



Рис. 8. Задание последовательности операций сборки

Стрелки обозначают связи между операциями. Задайте последовательно связи между всеми операциями, используя инструмент *Link Selected Activities*

A	tivity	/ Mar	nager	ment	
-]	10×	I'r	0	5	**
_					

Ваш процесс должен выглядеть, как показано ниже.



Рис. 9. Отображение процесса сборки

С помощью диаграммы планирования можно задавать параллельные процессы, используя логические операции.

Теперь необходимо сопоставить операциям соответствующие детали, т.е. указать системе какая именно деталь (или сборка) устанавливается на данной операции. Для этого выделите подсборку *Block Assembly* в дереве проекта (ветвь *ProductList*), щелкните на иконке *Assign an item*

а затем на операции Block Sub Assembly (ветвь ProcessList).

Раскройте ветвь *Block Sub Assembly* в дереве, чтобы убедиться, что операция проведена успешно.

Этап 4. Задание траектории перемещения компонентов и анализ пересечений компонентов

Задание траектории перемещения компонентов является одним из ключевых моментов при создании технологических процессов сборки.

Поскольку мы работаем в виртуальной среде, то перемещение любого компонента может быть создано независимо от других компонентов. После чего может быть проведен анализ на столкновения и взаимные пересечения компонентов.

Зададим траектории перемещения для следующих компонентов:

- Cover (Кожух)
- *Air Filter* (Воздушный фильтр)

Выполните следующую последовательность действий:

- 1. Выделите операцию, для которой вы хотите задать перемещение.
- 2. Создайте перемещение с помощью инструмента Move Activity



- 📕 и выберите тип вставки as child.
- 3. Убедитесь, что в окне предварительного просмотра выбрана правильная деталь (сборка), нажмите *OK* в окне *Edit Shattle*.
- 4. Перетащите деталь в сторону от двигателя, используя компас.



5. Зафиксируйте положение, нажав кнопку Record

- 6. Повторяйте предыдущее действие, пока не уложите деталь на стол.
- 7. Фиксируйте каждое положение детали, нажав на кнопку *Record*.
- 8. Используйте панель инструментов *Player* для отмены и просмотра перемещений.
- 9. Когда убедитесь, что деталь имеет необходимое положение (лежит на столе), закройте диалоговое окно *Track* и повторите те же действия для других деталей.

-

После того, как траектории построены для необходимых деталей, мы можем посмотреть симуляцию всего процесса и определить места возникновения проблемных участков, представляющих собой столкновения компонентов и их взаимное пересечение.

	Включите	опцию Clas	sh Detection	(Stop)	🥦 , запус	стите симуля-
цию	процес	са, испо	льзуя ком	ланду	Compile	Simulation:
Sim	ulation					
13	2; 2; <u>1</u>	9, R (
	После че	го запусти	те симуляци	ию, исп	ользуя па	нель Player:
Play	er					
]	→ 1	Time	• • • •	н 4		н н 🚼

Система должна автоматически определить столкновения компонентов и остановить симуляцию в момент возникновения конфликта, подсветив красным цветом место столкновения (рис. 10).



Рис. 10. Визуализация столкновений в DELMIA

Этап 5.Формирование файла отчета

Формат файла отчета настраивается для каждого конкретного предприятия в соответствии с его стандартами и потребностями.

В рамках данной работы предлагается воспользоваться стандартной формой файла отчета, поставляемой с дистрибутивом системы DELMIA.

Вся необходимая документация генерируется системой на основе скриптов (CATScript, VBScript), что дает практически неограниченные возможности по заданию формы отчетов, формата (и выводу всех необходимых параметров). Для вывода файла документации воспользуйтесь

инструментом Generate Documentation

В появившемся диалоговом окне выберите скрипт, нажав на кнопку *Browse*. Путь к файлу (рис. 11):

../CATIA/B17/intel_a/code/command/DocumentGeneratorHTML_WNT_DPM WorkImages.CATScript.

Process docu	mentation		
Documentatio	n script	(
Process name		~	Browse
Process : Proce	55		•
Path :	C:\DOCUME~1\awb\LOCALS~1\Temp		Browse
Main File Name:			
-		OK Close	Help

Рис. 11. Генерация документации

В поле Path укажите путь для сохранения файла отчета и имя в поле *Main File Name*. Нажмите *ОК* для создания файла отчета. На следующем рисунке представлен пример такого отчета.

	Activ Gen	vity Detail : Engine Assembly.1 neration Date : 5/5/2005 at 12:18:42 PM			
	General Data		Specific Data		
Name : Engine Assembly.1 Type : Engine Assembly Description : No Description Cycle Time : 5 Calculated Cycle Time : 0 Begin Date : 0 End Date :			None		
	Items List]]	Resources List		
None		None			
		Related Activities			
<<<< Previous	Children		Next >>>>		
	 Block Sub Assy,1 Crankshaft Sub Assy,1 Intake Valve Sub Assy Ezhaust Valve Sub Assy Cran Sub Assy,1 Crankcase Cover Sub A Coil Sub Assy,1 Crankcase Cover Sub Assy,1 Crankcase Cover Sub Assy,1 Caduuretor Sub Assy,1 Caduuretor Sub Assy,1 Muffler Sub Assy,1 Air Filter Sub Assy,1 Cover Sub Assy,1 	1 <u>v.1</u> .es <u>v.1</u>			

Рис. 12. Форма отчета

Самостоятельная работа «Применение метода конечных элементов в системах MSC Nastran и Patran»

Компьютерные технологии инженерного анализа и виртуального моделирования

В настоящее время разработчики всех видов техники достигли значительных успехов в повышении производительности, скорости, грузоподъемности, безопасности, экологичности, комфортабельности при одновременном снижении материало- и энергоемкости своей продукции.

Однако, созданием хорошего изделия задача, стоящая перед предприятием, не исчерпывается. Как показывает мировой опыт, острая рыночная конкуренция вынуждает проводить разработку в кратчайшие сроки, с наименьшим объемом экспериментальной доводки, не снижая при этом качества и потребительских свойств своей продукции. Ведуцие производители видят выход в широком внедрении новых технологий в процесс разработки и подготовки производства сложной наукоемкой продукции. Одним из важнейших приоритетов в этом направлении является использование компьютерного моделирования при разработке изделий и технологических процессов, позволяющее значительно сократить сроки цикла «проектирование - производство», существенно снизить число опытных образцов и натурных испытаний. Практическая реализация такого подхода базируется на технологии проектирования, основанной на масштабном математическом моделировании и вычислительном эксперименте.

Современный уровень компьютерного моделирования позволяет оценить, исследовать и оптимизировать каждый компонент нового изделия в компьютерной, виртуальной среде, до изготовления дорогостоящих реальных прототипов. При этом проектирование сложных современных конструкций, как правило, требует совместного использования широкого набора различных систем инженерного анализа, ответственных за тот или иной этап расчетного сопровождения проекта (анализ движений, кинематики, моделирование систем контроля, огромный список задач от термо- и газовой динамики до анализа ударных и акустических процессов).

Обширный выбор систем инженерного анализа, представленный на рынке различными компаниями, ставит пользователей в трудное положение. Трудно обеспечить взаимодействие разрозненных систем друг с другом, сложно обеспечить их согласованное совместное использование в проекте. Поэтому, наряду с повышением функциональных качеств расчетных систем, актуальной темой последнего времени является обеспечение их взаимодействия и интеграции. Такая интегрированная среда, когда пользователь может легко применить нужные ему системы совместно, комплексно, является гораздо более ценной и эффективной, чем наличие набора разрозненных, совершенно разных, плохо совместимых программ специфического назначения.

Стратегия виртуальной разработки изделия (VPD – Virtual Product Development) – это интегрированная среда систем инженерного анализа. Системы VPD являются фактически проверенным международным стандартом для всех видов инженерного моделирования, включая мно-годисциплинарные приложения. Интегрированная среда систем инженерного анализа интенсивно используется ведущими предприятиями мировой индустрии, позволяя им выпускать лучшие изделия, сокращать сроки проектирования и получать максимум прибыли в жесткой конкурентной борьбе.

Действительно, сегодня каждая из систем VPD, взятая в отдельности, обеспечивает достаточную функциональность в тех или иных инженерных приложениях (например, системы анализа кинематики, различные конечно-элементные системы, пре- и постпроцессоры, системы управления процессами и т.д.). Каждая компонента, каждая система может использоваться индивидуально, изолированно от других продуктов. Но только комбинированное, комплексное использование систем может раскрыть их полный потенциал.

VPD системы обеспечивают гибкое управление и эффективное распределение информации, легкий обмен данными между системами, без промежуточных стадий конвертации и адаптации данных. Геометрическая информация оптимально передается от одного процесса к другому, и результаты одного расчета органично становятся исходными данными для следующего шага инженерного моделирования. Наряду с геометрией, обеспечивается передача данных по нагрузкам, всевозможным граничным условиям и т.д.

Каждая система имеет специфические, отличные от других систем инструменты и пользовательский интерфейс, однако общая среда, общность подходов, логичность, связность, легкость взаимодействия между системами, обеспечивают пользователю возможность работы с любой из них без длительного переучивания. Адаптивная, гибкая среда VPD систем позволяет легко переключаться на решение новых типов задач. Особо следует отметить удобство моделирования процессов, требующих использования не одной, а нескольких расчетных систем. Интегрированная среда VPD продуктов – идеальный инструмент для решения таких сложных, многодисциплинарных проблем. Кроме того, рассматриваемые системы дают возможность компьютерного моделирования изделия в сборе с учетом всех эксплуатационных нагрузочных параметров. Этот подход позволяет достоверно исследовать функционирование каждого компонента конструкции в составе всего изделия, а это в свою очередь позволяет осуществлять интеграцию информации не только на уровне подразделений предприятия-разработчика, но и обмен информацией на уровне предприятий-смежников.

MSC.Patran - интегрированная среда моделирования, анализа и проектирования

MSC.Patran – программный продукт компании MSC.Software, обеспечивающий интеграцию автоматизированных систем проектирования, моделирования, анализа и оценки результатов расчетов. Использование Patran в комбинации с другими программными продуктами позволяет достичь наибольшей эффективности в оценке работоспособности и оптимальности конструкции изделий при их разработке, производстве и эксплуатации еще до того, как начнутся изготовление и испытания опытных образцов.

Раtran располагает развитыми средствами генерации конечноэлементных сеток, что придает этому процессу гибкость и высокую степень автоматизации. Нагрузочные и граничные условия могут быть увязаны как с геометрическими, так и с конечно-элементными моделями. Развитые средства визуализации помогают ускорить и повысить качество анализа полученных результатов расчета.

При использовании Patran основой создаваемой конечноэлементной модели является, как правило, геометрическая модель. Поэтому Patran предоставляет прямой доступ результатам к геометрического моделирования в наиболее популярных в мире программных пакетах автоматизированного проектирования или к универсальным форматам обмена данными при отсутствии прямых интерфейсов. Patran имеет также собственные обширные возможности создания и модифицирования геометрических моделей, контроля CADгеометрии и преобразования ее перед построением конечно-элементной модели.

Patran используется не только для работы с расчетными системами компании MSC.Software, он на высоком уровне интегрируется и с системами анализа многих других разработчиков.

MSC.Nastran - расчет и оптимизация конструкций

Главный продукт компании MSC.Software - Nastran - это одна из лучших на рынке конечно-элементных систем. В сфере, где ненадежные результаты могут обернуться миллионами долларов дополнительных расходов на разработку, Nastran вот уже более 40 лет доказывает свою точность и эффективность. Постоянно развиваясь, он аккумулирует в себе достоинства новейших методик и алгоритмов конечно-элементного анализа.

MSC.Nastran обеспечивает практически полный набор расчетов, напряженно деформированного расчет состояния, включая собственных частот и форм колебаний, анализ устойчивости, решение задач теплопередачи, исследование установившихся И неустановившихся процессов, нелинейных статических процессов, нелинейных динамических переходных процессов, расчет критических частот и вибраций роторных машин, анализ частотных характеристик воздействии случайных нагрузок, спектральный при анализ И Предусмотрена исследование аэроупругости. возможность моделирования практически всех типов материалов, включая гиперупругие. Расширенные функции композитные И включают технологию суперэлементов (подконструкций) и макроязык DMAP для создания пользовательских приложений.

Наряду с расчетом конструкций Nastran может использоваться и для оптимизации проектов. Это делается путем вариации параметров формы, размеров и свойств проекта. Благодаря своей эффективности обрабатывают алгоритмы оптимизации неограниченное число проектных параметров и ограничений. Вес, напряжения, перемещения, собственные частоты многие другие характеристики И могут рассматриваться либо в качестве целевых функций проекта (в этом случае их можно минимизировать или максимизировать), либо в качестве ограничений. Алгоритмы анализа чувствительности позволяют исследовать влияние различных параметров на поведение целевой функции и управлять процессом поиска оптимального решения. Nastran применяется также и для планирования экспериментов (определения датчиков) полученных расположения оценки полноты мест И экспериментальных данных.

Задание

Студентам необходимо самостоятельно создать в CAD-системе 3-D модель детали, а также задать исходные данные, необходимые для выполнения расчета: свойства материала, величина и точки приложения нагрузки, граничные условия.

Для проведения инженерного расчета в MSC.Nastran и MSC.Patran необходимо выполнить следующие действия:

- 1. Создать новую базу данных.
- 2. Импортировать геометрию из САД-системы.
- 3. Создание поверхности с сеткой конечных элементов.
- 4. Задать нагрузку и граничные условия.
- 5. Создать материал.
- 6. Задать физические свойства.
- 7. Задать нагрузки и граничные условия.
- 8. Выполнить анализ в системе MSC.Nastran.
- 9. Обработка результата постпроцессором.

Создание базы данных

Для того чтобы начать работу в системе MSC.Patran необходимо создать базу данных.

- 1. В главном меню выбрать File > New. Появится панель New > Database.
- 2. Ввести в текстовое поле File Name имя базы данных. Нажать ОК.
- 3. Появится New Model Prference панель для указания общих параметров модели (рис. 13). В этой панели можно задать точность создания модели (Tolerance) или оставить значения по умолчанию (Default).
- 4. Необходимо убедиться, что к системе MSC.Patran подключена вычисляющая программа MSC.Nastran, которая указана в разделе **Analysis Code**. Если имеется лицензия можно подключить другие программы.

5. Выбрать в разделе Analysis Type > Structural и нажать Ok. База данных будет создана, но она будет пустая.

Model Preference for: strut.db
 Tolerance ● Based on Model ● Default
Approximate Maximum Model Dimension:
10.0
Analysis Code: MSC.Nastran 💌
Analysis Type: Structural 💌
OK Reset

Рис. 13 Панель для указания общих параметров модели

Импорт геометрии из САD-системы

На данном этапе выполнения работы необходимо импортировать 3-D модель детали в систему MSC.Patran. Для этого следует в главном меню выбрать **File > Import** и указать импортируемый файл модели.

Создание поверхности с сеткой конечных элементов

Для создания поверхности SMARTEAM сеткой конечных элементов необходимо выполнить следующие действия:

1. В главном меню указать кнопку Elements (рис. 14).



Рис. 14 Панель инструментов MSC. Patran

Откроется окно, в котором требуется задать свойства конечных элементов сетки (рис. 15).

- 2. В верхней части появившейся панели выбрать Action > Create, Object > Mesh, Type > Surface.
- Настроить или проверить установку кнопок Element Shape (форма элемента) > Quad, Mesher (тип генератора сетки)> IsoMesh, Topology > выбрать элемент.
- 4. Предварительно щёлкнув в окне Surface List (список поверхностей) указать мышью на экране последовательно разбиваемые поверхности. В окне должна появиться надпись со списком выбранных поверхностей.
- 5. Ввести в окне Value (величина) длина ребра элемента. Снять маркер параметра Automatic Calculation (автоматическое вычисление) и нажать Apply.
- 6. Поскольку конечные элементы не соединены вдоль геометрических границ, то необходимо «сшить» их вместе. Для этого на верхней части панели Finite Element выбрать Action > Equivalence, Object > All, Method > Tolerance Cube (область допусков представляется кубом с ребром 0,005 м). В пределах этой области все узлы будут объединяться в один. Нажать кнопку Apply. в строке сообщений внизу экрана будет указано сколько узлов было объединено.

Action: Create 🔻				
Object: Mesh 🔻				
Type: Solid 🔻				
Output ID List				
Node 1				
Element 1				
Elem Shape Tet 🔻				
Mesher TetMesh 💌				
Topology Tet10 🔻				
TetMesh Parameters				
Node Coordinate Frames				
Input List				
Global Edge Length Global Edge Length Global Edge Length				
Value 0.5				
Assembly Parameters				
Pron Name:None_				
Prop. Type: - N/A -				
Select Existing Prop				
Create New Property				
-Apply-				

Рис. 15. Панель свойств элементов сетки

Создание материала

Для того чтобы создать материал в MSC.Patran необходимо выполнить следующие действия:

- 1. В главном меню, нажать кнопку Materials (рис. 14). В появившейся панели выбрать Action > Create, Object > Isotropic, Method > Manual Input (ручной ввод параметров материала).
- 2. В окне Material необходимо указать название материала.
- 3. Нажать кнопку Input Properties (ввод свойств).
- Появится панель Input Options и в этой панели в строку Elastic Modulus ввести значение модуля упругости в H/м2; в строку Poisson Ratio (коэффициент Пуассона) (рис. 16). Остальные строки можно не заполнять, так как они заполняются при решении динамических задач.

Input Options	
Constitutive Model:	Linear Elastic 🔻
Property Name	Value
Elastic Modulus =	30e6
Poisson Ratio =	0.3
Shear Modulus =	
Density =	
Thermal Expan. Coeff =	
Structural Damping Coeff =	
Reference Temperature =	
Temperature Dep/Model Variable Fi	elds:
4	
Current Constitutive Medaler	
Current Constitutive Models.	
<u>र</u>	Þ
ОК	Clear Cancel

Рис. 16. Задание свойств материала

5. Нажать **Ok**, чтобы закрыть панель и **Apply** на панели **Materials**. При этом свойства материала только выбраны, но не назначены на конечные элементы.

Задание физических Свойств

- 1. В главном меню нажать кнопку Properties (рис. 14).
- 2. В появившейся панели Element Properties указать Action, Type.
- 3. В окно **Properties Set Name** (имя варианта свойств) ввести имя свойства.
- 4. Нажать на кнопку Input Properties.
- 5. На появившейся панели Input Properties, щёлкнуть мышкой в окне Material Name и выбрать материал заданный ранее в списке Material Properties Set.
- 6. Ввести в окно Thickness панели Input Properties величину толщины материала.
- 7. На панели Element Properties поместить курсор в окно Select Members (Выбрать элементы) и затем курсором выбрать все поверхности на экране.
- 8. Нажать кнопку Add (придать свойства), в окне Application Region должна повториться запись со списком выделенных поверхностей. После этого необходимо нажать кнопку Apply. При этом заданные свойства будут назначены указанным конечным элементам.

Input Properties Solid (CHEXA)			- 🗆 🗵
Property Name	Value	Value Type	
Material Name	Inisteel	Mat Prop Name	
[Mater, Orientation]	ni.steel	String V	
[Integration Network]		String	
[Integration Scheme]		String	
[Output Locations]		String	
	,		
Enter the Material Name or select a	spatial field with the icon.	<u> </u>	
ок	Clear	Cancel	

Рис. 17. Окно назначения материала

Моделирование нагрузок и граничных условий

Задание распределённых нагрузок.

- 1. В главном меню нажать кнопку Loads/BCs.
- 2. В появившейся панели Loads/Boundary Conditions выбрать Action>Create, Object>Distributed Load (распределённая нагрузка), Type>Element Uniform (равномерная по элементу) (рис. 18).
- 3. В окно New Set Name ввести имя.
- 4. Нажать кнопку Input Data (ввести данные).

- 5. В появившейся панели Input Data в окно Edge Distributed Load (нагрузка, распределённая по ребру) ввести, например: < , , -200>, что означает нагрузка задана против положительного направления оси z . Пробелы по направлениям осей x и y означают отсутствие нагрузок. Окно Edge Distributed Moment необходимо оставить пустым, т.е. нагрузка от моментов отсутствует. Нажать Ok.
- 6. В панели Loads/BC нажать кнопку Select Application Region (выбрать область приложения нагрузок), в появившейся панели с этим же названием под разделом Geometry Filter (геометрический фильтр) указать Geometry. Это означает, что нагрузки будут задаваться на геометрическом образе модели, а не на конечноэлементном.

Задавать нагрузки и граничные условия удобнее на геометрическом образе, т.к. конечных элементов на одном геометрическом элементе может быть очень много и задавать нагрузку на каждый элемент трудоёмко. MSC.Patran перед расчётом автоматически перенесёт все нагрузки и граничные условия на соответствующие конечные элементы.

- 7. В этой же панели поместить курсор в окно Select Surface Edge (выбрать ребро поверхности). Используя указатель мыши, выбрать четыре ребра средних поверхностей, затем нажать Add. Номера выбранных рёбер поверхностей должны появиться в окне Application Region. Нажать Ok.
- 8. Нажать кнопку **Apply** на панели **Loads/BC**. На экране должны появиться символы, обозначающие нагрузку (стрелки) и её величину (цифры возле стрелок).

В качестве граничного условия в задаче будет выступать **Displacement** (ограничение перемещений). Для задания граничных условий необходимо выполнить следующие действия:

1. В главном меню нажать кнопку Loads/BCs.

2. В появившейся панели выбрать Action > Create, Object > Displacement, Type > Nodal (узлы) (рис. 18).

3. В окно New Select Name ввести имя граничных условий pinned.

4. Нажать Input Data (ввести данные).

5. В появившейся панели в окно **Translations** (перемещения) ввести <0, 0, 0>, что означает запрет перемещений по осям x, y, z. Окно **Rotations** необходимо оставить пустым. Нажать **Ok**.

6. Нажать кнопку Selection Application Region в панели Loads/BCs, после этого появится панель с таким же именем, в которой необходимо указать Geometry в разделе Geometry Filter.

7. Поскольку в геометрической модели присутствую различные геометрические сущности (Entities): рёбра, поверхности, вершины, то необходимо точно указать какие сущности будут ограничены по перемещениям. Для этого в Select Menu (меню выбора) нажать кнопку Edge Option.

Это означает, что на экране курсором мыши будут выделяться только Edges (рёбра) геометрической модели.

8. Поместить в окно Select Geometric Entities курсор, и используя указатель мыши выбрать на экране четыре ребра на внешних поверхностях пластины. При выборе необходимо удерживать нажатой клавишу Shift, чтобы происходило суммирование выбора. Затем нажать Add и в окне Application Region должны появиться номера выбранных рёбер. Нажать Ok.

9. На панели Loads/BCs нажать Apply. С помощью кнопок просмотра изображений можно подобрать вид изображения для просмотра нагрузок и граничных условий. Граничные условия изображаются условными символами, около которых указываются номера ограниченных степеней свободы: 1, 2, 3, это означает, что ограничены перемещения по осям x, y, z. Вращательные степени 4, 5, 6 не указаны – значит, ограничений нет.

Создание входного файла для MSC.Nastran

Для каждой расчётной программы, которые могут быть использованы вместе с MSC.Patran (их тип задаётся в **Preference**), создаются свои входные файлы. В данном случае рассмотрим создание файла для программы MSC.Nastran.

1. В главном меню нажать кнопку Analysis.

2. На появившейся панели выбрать Action > Analyze, Object > Entire Model, Method > Analysis Deck (метод брать с рабочего стола или панели управления) (рис. 19).

3. Нажать кнопку Solution Туре (тип задачи).

4. В появившейся панели выбрать Linear Static (линейный статический расчёт), нажать Ok.

5. Нажать на кнопку **Translation Parameters**. В появившейся панели в разделе **Data Output** (вывод данных) нажать кнопку **XDB and Print** и в появившемся меню выбрать **XDB Only** (для сокращения размеров файла результата). Нажать **Ok**.

Object: Displacement ▲ Type: Nodal ▲ Current Load Case: △ Default ◄ Type: Static Existing Sets ▲	ABC Set Scale Factor
Type: Nodal	nslations <t1 t2="" t3=""> I 0 0> ations <r1 r2="" r3=""> ></r1></t1>
Current Load Case:	nslations <t1 t2="" t3=""> I 0 0> ations <r1 r2="" r3=""> ></r1></t1>
Existing Sets	
Spatia Spatia	al Fields
New Set Name	FEM Dependent Data alysis Coordinate Frame ford 0 OK Reset
Input Data Select Application Region	

Рис. 18. Окна задания нагрузок и граничных условий

6. Нажать кнопку **Apply**, чтобы запустить MSC.Nastran. На экране появляется окно MSC.Nastran с указанием размеров оперативной и дисковой памяти, занятой под задачу, а также сообщение, что MSC.Nastran запущен, время и число. Через некоторое время, после окончания расчёта, окно пропадает. В строке сообщений можно прочитать сообщение о том, что трансляция входного файла была завершена успешно.

Action: Analyze 🔻	MSC.Nastran
Object: Entire Model 🔻	Solution Type
Method: Full Run 🔻	Solution Type:
	C NONLINEAR STATIC
Code: MSC.Nastran	C NORMAL MODES
Type: Structural	C BUCKLING
	C COMPLEX EIGENVALUE
	C FREQUENCY RESPONSE
Available Jobs	C TRANSIENT RESPONSE
	O NONLINEAR TRANSIENT
	C IMPLICIT NONLINEAR
T P	C DDAM Solution
	Select ASET/OSET
Job Name	
strut	
Job Description	
MSC.Nastran job created on	Solution Parameters
12-Sep-05 at 10:44:07	
	Solution Sequence: 101
Translation Parameters	
Solution Type	OK
Direct Text Input	
Subcases	
Subcase Select	
Apply	

Рис. 19. Окно задания типа расчета

Передача в среду MSC.Patran результата решения задачи для обработки постпроцессором

- 1. В главном меню нажать кнопку Analysis.
- 2. На появившейся панели выбрать Action > Attach XDB, Object > Result Entities, Method > Local (рис. 20).
- 3. Нажать на кнопку Select Results File.
- 4. Нажать Apply на панели Analysis.

Action:	Access Results
Object:	Attach XDB 🔻
Method:	Result Entities 🛛 🔻
Code: M	SC.Nastran
Type: St	tructural
Available Jo	bs
strut	<u>م</u> ب ا
Job Name	
strut	
Job Descript	tion
MSC.Nastra 12-Sep-05	an job created on 🔺 at 11:06:39
s	elect Results File
Tra	nslation Parameters
	Apply

Рис. 20. Окно передачи результатов в среду MSC. Patran

Обработка результата постпроцессором

Создание Fringe-изображения и деформированного изображения (Deformation Plot)

1. В главном меню нажать кнопку Results.

2. На появившейся панели выбрать Action > Create, Object > Quick Plot (быстрое изображение).

3. В окне Select Result Cases панели Results выбрать Default, Static SubCase, в окне Select Fringe Result выбрать Displacement, Translational.

4. В окне Select Deformation Result панели Results выбрать Displacement, Translational.

5. Нажать кнопку **Apply**. На экране появится изображение панели с цветной шкалой уровней SMARTEAM указанием максимальных и минимальных величин перемещений (рис. 21).



Рис. 21. Окно результатов расчета
6. Для того, чтобы не мешало графическое изображение геометрической модели, необходимо убрать с экрана графику, нажав кнопку с изображением метлы (**Reset Graphics**), после этого нажать **Apply** на панели **Results**.

7. Можно просмотреть форму деформации, воспроизведя «оживление» модели. Для этого нужно в окне **Animate** поставить скобку и нажать

Apply. Появляется окно, в котором указаны параметры анимации (скорость анимации). Передвигая движок можно регулировать скорость анимации.

Самостоятельная работа «Построение виртуальных моделей станков с ЧПУ»

Введение

Сегодня на предприятиях приборо- и машиностроения наблюдается постоянное усложнение выпускаемой продукции. Усложняется и металлообрабатывающее оборудование, станки приобретают все большую универсальность, стойки ЧПУ оснащаются новыми функциями и вместе с тем усложняется и технология обработки на этом оборудовании.

Цена ошибки на современном производстве возрастает многократно, грозя порчей заготовок, выходом из строя инструмента и узлов станка, длительным простоем оборудования.

В связи с чем, при подготовке производства изделий на станках с ЧПУ необходимо использовать программные системы, основная задача которых - симулировать процессы, происходящие с оборудованием, инструментом, заготовкой. Вместе с тем, всё большую роль в успешности предприятия продолжает играть фактор времени, который выражается не только в сокращение сроков разработки изделия, но и в постоянном сокращении его времени жизни. Необходимо учитывать, также, и постоянно растущее влияние информационных технологий в технической подготовке производства, постоянно растущее значение кооперации предприятий и тенденции в области развития виртуальных предприятий. Это определенно внесет изменения в существующую методику разработки технологии производства изделий.

В такой ситуации единственно возможным путем обеспечения высоких темпов, гибкости и качества производства, является использование "виртуальных производственных комплексов". С помощью этих программных систем становится возможным осуществлять верификацию и оптимизацию управляющих программ для любых типов станков с ЧПУ, осуществлять контроль состояния инструмента и заготовки, повысить эффективность использования парка машин.

Система Vericut

Предприятия, желающие уменьшить затраты и увеличить эффективность производства зачастую вкладывают немалые средства в покупку современного оборудования с ЧПУ. Однако, опыт показывает – бесполезно односторонне развивать средства производства (оборудование), необходимо также постоянно развивать программное обеспечение, используемое в технологическом бюро. Имеется множество примеров, когда ничтожно малое время, потраченное на проверку управляющей программы до ее передачи в цех, может сберечь дорогостоящее оборудование, оснастку и инструмент. Одним из инструментов, разработанных специально для этой цели, является программное обеспечение проверки управляющий программ (УП) (верификаторы управляющих программ).

Многие из заводов теперь полагаются на программное обеспечение верификации для того, чтобы проверить собственные УП. Программное обеспечение может гарантировать, что первая полученная деталь будет правильной, таким образом, пробные прогоны УП на деревяшках или алюминии станут вещью из прошлого.

Кроме того, определенную выгоду приносит способность верификаторов УП оптимизировать скорости подачи. Время производственного цикла, качество детали и ресурс инструмента могут улучшаться в результате более совершенного контроля над режимами обработки. Такая УП с оптимизированными параметрами обработки становится проще с точки зрения ее последующего контроля оператором станка.

На сегодняшний день одним из основных верификаторов УП является VERICUT. Это признанная во всем мире система, ее используют ведущие предприятия Европы, Америки, России среди них: Airbus, Volvo Aero Norway, Dassault Aviation, Cosworth Racing, Prost Grand Prix, Renishaw, Siemens, Mazak, Sandvik, BMW Group.... Как видно, эта система представлена в различных сферах от производства инструмента, до автомобильной, авиа и аэрокосмической промышленности.

Vericut стоит на трех основных китах:



Верификация (Verification)

Верификация обеспечивает реалистичную имитацию и контроль УП для фрезерной, токарной, эрозионной и др. видов обработки, а также обнаружение таких ошибок, как неправильное построение траектории инструмента, недостаточная точность обработки, врезание на ускоренной подаче, столкновения инструмента или державки, столкновения с зажимным приспособлением и оснасткой, ошибки постпроцессора.

В любое время можно изменять вид и масштаб модели, выполнять сечения, устанавливать полупрозрачный режим. Можно измерять толщину стенок, объем, расстояния, углы и др.

Верификация управляющих программ выполняется для широкого спектра систем ЧПУ. При этом поддерживаются команды коррекции инструмента, сверлильные циклы, подпрограммы и т. д. Имеются средства для создания и редактирования описаний систем ЧПУ, а также для конвертирования УП.

Инструменты верификации предоставляют программисту средства для проверки траектории движения инструмента в УП на электронной модели станка до ее отправки в цех. Такой метод практически полностью исключает как возможность аварийного отказа станка, так и длительную доводку УП на станке.

Необходимо отметить, что верификация в Vericut, в отличие от встроенных модулей верификации CAD/CAM систем, ведется в кодах станка, т.е. идет проверка G-кода которых поступает на станок.

Оптимизация (OptiPath)

В случае использования опций оптимизации, программное обеспечение читает файл траектории движения инструмента (УП) и автоматически изменяет назначенные скорости подачи так, чтобы подобрать наиболее оптимальную подачу в зависимости от условий обработки и параметров режущего инструмента. Индивидуальный подбор режимов резания повышает эффективность процесса обработки и уменьшает время изготовления детали без потери качества. Оптимизация может применяться для черновой, получистовой, чистовой и высокоскоростной обработки. При оптимизации используется встроенная база знаний и математический аппарат.

При оптимизации Vericut читает файл траектории движения инструмента (G-код) и разбивает движение на ряд меньших фрагментов. Зная количество материала удаленного в каждом фрагменте, Vericut назначает лучшую скорость подачи для данного фрагмента траектории (рис. 22). Таким образом, в кадрах УП меняются только значения под адресом F (скорость подачи).

В процессе оптимизации учитывается: мощность и тип шпинделя; величина ускоренной подачи; жесткость фиксатора детали; характеристики фрезы, включая материал, длину и число зубьев.

Реалистичная имитация работы станка (Machine Simulation)

При многокоординатной обработке совместная работа линейных и поворотных исполнительных органов станка может привести к столкновениям и поломке инструмента, приспособлений или узлов станка. Реалистичная имитация работы станка, с контролем столкновений, помогает избежать таких ситуаций. Система контролирует столкновения инструмента, державки, шпинделя, зажимного приспособления, заготовки, исполнительных органов и узлов станка.



Рис. 22: верхний рисунок – траектория и скорость подачи, установленные САМ системой; нижний рисунок – оптимизированные в Vericut траектория и скорость подачи

Имеющаяся в системе библиотека станков содержит широкий спектр моделей многокоординатного оборудования. Можно создавать собственные описания станков или редактировать имеющиеся.

Наибольший эффект применения Vericut дает для 4- и 5координатных станков, так как при многокоординатных перемещениях наиболее вероятны не контролируемые САМ-пакетом столкновения инструмента с заготовкой, приспособлением, узлами станка. Кроме того, проверяя правильность УП, Vericut одновременно проверяет правильность работы постпроцессора.

Общая структура системы Vericut

Общая структура Vericut представлена на рис. 23, кратко об основных составляющих в таблице 1.



Рис. 23. Общая структура Vericut

Таблица 1: основные компоненты Vericut

Model Export	Экспорт модели обработанной заготовки в форматах STL и IGES.
Multi-Axis	Симуляция и верификация 4-х и 5-ти координатной фрезерной, сверлильной, токарной обработки и сме- шанной токарно-фрезеной обработки.
CNC Machine Probing	Создание измерительных циклов на любом этапе процесса обработки. Снижает вероятность ошибки благодаря симуляции измерительных операций.
Inspection Sequence	Позволяет сохранять проект «в процессе» (in- process), чтобы не отрабатывать уже оптимизирован- ные кадры управляющей программы.
EDM Die Sinking	Точное моделирование и контроль работы прошив- ных электроэрозионных станков. Определение заре- зов, пережога или недожога, объема удаленного ма- териала, областей контакта и перекрытия электрода (включая составные электроды).
AUTO- DIFF	анализ различия между конструкторской моделью и обработанной заготовкой в Vericut.

Cutter / Grinder Verification	Контроль обработки на многокоординатных режуще- шлифовальных станках. Элементы упрощенного ин- терфейса специально для шлифовальных станков.
Cutter / Grinder Machine Simulation	Моделирование обработки на режуще- шлифовальных станках и проверка на предмет столк- новений деталей и механизмов станка.
CAD/CAM Interfaces	Для систем подобных Vericut, принципиальную роль играет наличие интерфейсов передачи данных с CAD/CAM системами. Vericut имеет прямые интерфейсы со следующими системами: CATIA V4/V5, Unigraphics, Pro/E, hyperMILL и др. Наличие прямого интерфейса позволяет передавать данные (электронные модели станков, заготовок, де- талей и др.) из CAD/CAM системы в Vericut без ис- пользования нейтральных форматов STL и IGES. При обратной передаче – из Vericut в CAD/CAM систему данные передаются в «родных» форматах CAD/CAM системы, а также существуе возможность передать в CAD/CAM систему модель заготовки в процессе об- работки (Inproces) для проектирования онастки, ана- лиза и пр.
Mold & Die	Проверка, анализ, оптимизация траекторий фрезерной трехкоординатной обработки, обработки отверстий и электроэрозионной обработки. Проверка на предмет зарезов, а также столкновений заготовки и крепежной оснастки с режущими и нережущими частями инструмента

Начало работы

Интерфейс Vericut

Vericut имеет доступный интерфейс (рис. 24). В рабочем окне отображается модель заготовки и модель станка с заготовкой как вместе так и по отдельности, ниже движок управления скоростью симуляции и окно сообщений. В левой части окна отображается дерево проекта, в верхней части экрана расположены иконки управления проектом (рис. 25).

В дереве проекта: Project – имя пректа, Setup – установ, Control – вид стойки ЧПУ, Machine – имя станка из каталога, во вкладке Attach находятся модели приспособлений, заготовки и модель детали.



Рис. 24. общий вид интерфейса Vericutpuc 4: дерево проекта

В Vericut также существует дерево компонентов (Component tree) (рис. 25) в котором отображена кинематика станка и модели исполнительных механизмов.

Новый проект



Рис 25: Дерево компонентов, где: B,C – поворотные оси; X,Y,Z – оси перемещений компонентов станка; во вкладках Models – находятся соответствующие модели подвижных и статичных узлов станка

VERICUT может симулировать в одной сессии множество установок, используемых для создания детали. Файл проекта может содержать одну или несколько установок. Множественные установки могут иметь несколько NC операций, выполняемых на одном станке с деталью, перемещаемой для каждой операции. NC операции могут быть на разных станках с разными оснастками, инструментами и настройками. Каждая установка ссылается на свой собственный станок, систему управления, инструменты, NC программы, оснастки, настройки симулирования и т.д.

Установка может быть создана из эскиза, импортирована из другого файла проекта или из старой версии пользовательского файла.

Новый проект начинается с задания кинематики будущего станка. Vericut будет отрабатывать УП даже если не заданы модели узлов станка, но в этом случае он не будет отслеживать столкновения между узлами станка.

На рисунке 26 показана схема пяти координатного станка с ЧПУ, показана система координат станка (оси XwYwZw), оси движения (X, Y, Z), оси вращения (A, B) и основные компоненты, файлы управляющий программ обрабатывает система управления Fanuc 12M CNC.



Рис. 26: схема пяти координатного станка Cincinnati T30

Новый файл проекта File > New Project, при помощи иконки Working Directory 2 задается рабочая директория проекта, в которой находятся файлы проекта. Далее отображается система координат компоненты и модели Machine/Cut Stock в виде: кликнуть правой кнопкой мыши в рабочем окне VERICUT > Display Axes > Component, Model; кликнуть правой кнопкой мыши в рабочем окне VERICUT > View Type= Machine/Cut Stock.

Далее загружается система управления либо из рабочей директории, либо из имеющейся библиотеки: Configuration > Control > Open, Shortcut=Library. После этого действия в дереве проекта (Project Tree) во вкладке Project > Setup1 > CNC Machine > Control появляется имя системы управления.

Далее определяются компоненты на стороне инструмента станка с "Base" в "Tool" – **Base** > X > Y > A > Spindle > Tool, на этом же шаге будет задан цвет для моделей узлов станка. Отобразим дерево компонентов (Component tree) для этого в главном меню выбрать **Configuration** > **Component Tree**. Установим цвет для компонента "Base", для этого в дереве компонентов правой кнопкой мыши выбираем

вкладку **Base (0,0,0)**, нажмите правую кнопку мыши и выберите

Component Attributes, в открывшемся окне Modelling (рис. 27) на вкладке Component Attributes, указать цвет (color) и выбрать Apply. Окно Modeling оставьте открытым для установки цветов компонентов, которые будут добавлены позже.

Далее добавляются оси перемещений станка (X, Y, A), компонент «шпиндель» (spindle) и инструмент (tool). Чтобы добавить ось X в Base,

правой кнопкой мыши выберите в Component Tree **Base (0,0,0)** и выберите Append > X Linear, в окне Modeling укажите цвет (Color), нажмите Apply. Таким же образом вводятся оси "Y" в "X" и "A" в "Y". В случае с осью вращения A, в меню Append необходимо выбрать A Rotary.

Добавление шпинделя к оси вращения А. Правой кнопкой мыши

выберите в Component Tree 4 (0,0,0) и выберите Append > Spindle , в окне Modeling перейдите на вкладку Position и в графе Position (рис. 28) укажите смещение 0 0 -12.5, укажите цвет.

Modeling			x
Active Component Spindle 💌 Component Type : Spindle			
8	Selected 🔿 M	odel 💿 Compo	nent
Model Position Com	onent Attribut	es) Import	
Color 2:Aquan	narine	Visibility	Machine View 💌
- Machine			
Motion Axis	Z	Connect To	A
Disable Auto Direction	n 🗖	Rapid Rate	5000
Reverse Direction		SubSystem	1
Mixed Mode	Shade	Tool Index	1
			Toolchain Parameters
- Accel/Decel			
Accelerati	on (units/sec*:	sec) 125	
Decelerati	on (units/sec*	sec) 125	
Max Feed	Velocity (units)	(min) 5000	
Max Veloc	ity for Directior	n Change 600	
ок		Apply	Cancel

Рис. 27: вид вкладки Components Attributs окна Modelling

Добавление «Tool» в «Spindle». Правой кнопкой мыши выберите в

Component Tree Spindle (0,0,-12.5) и выберите Append > Tool, в окне Modeling установите Color=White, Apply.

Компонент Tool определяет, откуда будут загружены токарные инструменты. Компонент Tool должен быть определен перед обработкой файла траектории движения инструмента (УП). Ноль инструмента обычно размещается на пересечении оси инструмента с плоскостью шпинделя.

Modeling
Active Component Spindle 💌 Component Type : Spindle
Selected O Model 💿 Component
Model Position Component Attributes Import
Translate Rotate Assemble Matrix Csys Mirror
From 0 0 0 To 0 0 0 Move
Location Relative to Parent C Location Relative to Machine Csys
(Position 0 0 - 12,5) Undo
Angles 1000
OK Apply Cancel

Рис. 28. Вид вкладки Position окна Modeling

Следующие шаги определяют компоненты с "Base" в "Stock" – компоненты на стороне заготовки станка: Base > Z > B >Attach > Fixture > Stock > Design. Оси Z Linear и B Rotary добавляются аналогично осям X и A. Для добавления Fixture > Stock > Design к оси B, нужно вырезать (Cut) компонент Attach из Base и добавить его к оси B, для чего нужно правой кнопкой мыши выбрать —Attach (0,0,0) в дереве ком-

понентов, выбрать *f* Cut, затем правой кнопкой мыши выбрать

🤗 B (0,0,0) и выбрать 🖺 Paste.

В конце всех преобразований, дерево компонентов (Components tree) должно принять следующий вид (рис. 29).



Рис. 29. дерево компонентов до и после добавления кинематики станка

Ноль компоненты Fixture это место, откуда будут загружаться модели оснастки. Наличие оснастки в станке не оказывает влияния на то, как будет обработан файл траектории движения инструмента (УП), однако, это необходимо для обнаружения столкновений между оснасткой и другими компонентами станка.

Ноль компоненты Stock это место, в которое будет помещена заготовка. Этот тип компонента должен быть определен в каждом станке. Компонент Stock обычно привязан к компоненту Fixture, но в общем случае компонент Stock может быть привязан к любому другому компоненту. Единственное условие – компонент должен быть определен перед обработкой файла траектории движения инструмента (УП).

Сохраните проект в рабочей директории, как файл станка: в дереве компонентов (Components tree) File > Save As Machine.

Следующим шагом нужно задать модели элементов для составленной кинематики, для того, что бы Vericut мог отслеживать столкновения и опасные сближения между всеми элементами системы «элементы станка – приспособление – инструмент – заготовка».

Модель станка в Vericut может быть построена с любой степенью детализации. Для определения формы узлов станка, заготовки и технологической оснастки и Vericut организованы простейшие функции моделирования: построение цилиндров, конусов, блоков, профильных тел вращения. Элементы более сложной формы импортируются из CADпакетов в формате STL или IGES. Для точного позиционирования элементов модели друг относительно друга в системе имеются средства их переноса. Однако при создании элементов модели станка в CAD-пакете целесообразно вести их построение относительно одной системы координат, как правило, относительно «нуля» станка. В этом случае при передаче элементов моделей через CAD-интерфейс Vericut или при помощи форматов STL или IGES, элементы модели буду правильно позиционироваться в Vericut.

Теперь наполним проект моделями элементов станка. Добавим "Base" модель, в Component Tree правой кнопкой мыши выберите

Base (0,0,0), выберите Component Attributes, вкладка Model (рис. 30), Type=Model File, в строке Model File указать путь к директории в которой находится модель; задать цвет подели; Add.

Modeling
Active Component Base 🔄 Component Type : Base
Selected Model Component
Model Position Component Attributes Import
Type Model File
Model File Sketcher
Normals Outward
Units Millimeter
Color 9:Pale Green
Add Visible
Preserve Stock Transition
OK Apply Cancel

Рис. 30. Вид вкладки Model окна Modeling

Таким образом, добавляются все остальные модели. Если модель добавляется к компоненту, которому было присвоено смещение, то и модели нужно присвоить такое же смещение!

При размещении станка в «нуле», компоненты "Y" и "Z" попадают в состояние столкновения (рис. 31). Так как станки VERICUT сконструированы так, что все компоненты размещаются в соответствующих им нулях, столкновения во время определения станка будут общими.

Следующий шаг – определение таблицы начальных положений органов станка (Initial Machine Location table) для перемещения станка из состояния столкновения в начальную позицию. Таблицы размещения станка описывают, как будет позиционирован станок первоначально, как станок двигается при смене инструментов и шпинделей, а также размещение референтной точки станка.



Рис. 31. Все компоненты станка находятся в соответствующих им нуля.

Для задания Initial Machine Location table в главном меню Configuration > Machine Settings > вкладка Tables, выбрать Add/Modify, в строке Table Name указать *Initial Machine Location*, в строке Values указать значения 0 60 62, Add (рис. 32).

👽 Add/Modify Machine Table	×	🕜 Machine Settings 🛛 🔀
Table Name		☑ Machine Simulation On
SubSystem ID 1		Collision Detect Tables Travel Limits Axis Priority
Register: 1		- Machine Tables
SubRegister: 1		Hitial Machine Location Subsystem:1, Values:Y60 Z62
C Select From/To Locations		
Feature Name Offset		
From Component 💌 Base 💌 0 0 0	E.	
To Component 💌 Base 💌 000	E.	
🗖 Calculate when Used		
Enter Offset (or select 2 points)		
Values (XYZABCUVWABC): 0 60 62	The second secon	
Add Modify Close		Add/Modify] Delete
		OK Apply Cancel

Рис. 32. Определение таблицы начальных положений органов станка.

После проделанных манипуляций в окне Add/Modify Machine Table, в окне Machine Settings появится строка Initial Machine Location со значениями смещения > OK.. После введенных изменений необходимо сде-

лать сброс станка нажатием на иконку «Reset Model» , после чего модель станка должна приять начальное положение X0 Y60 Z62 A0 B0 (рис. 33).

Пересохраните проект.

Последний шаг в разработке проекта – добавление моделей заго-

товки и оснастки. Отобразите Project Tree нажатием на иконку **ГРЕ**. В Project Tree кликните правой кнопкой мыши на Setup 1 и выберите

Expand All Children, в Project Tree дважды кликните на **Fixture,** появится окно Modeling в нем выбрать вкладку Model (рис. 30) и ука-

зать путь к файлу модели. Таким же образом в ветку **Stock (0,0,0)** дерева проекта (**Project tree**) добавляется модель заготовки (рис. 34).

В Project tree кликните правой кнопкой мыши на **10** Tooling и

выберите *Open ...* и укажите путь путь к библиотеке инструмента.

Последний шаг – добавление УП. В Project Tree кликните правой

кнопкой мыши на NC Programs, выберите **I** Add/Modify NC Programs..., в окне TC Program выберите Add и укажите путь к УП.

Рис. 33. Правильное расположение модели собираемого станка.

Рис. 34. Вид собранной модели станка с заготовкой и оснасткой

После введенных изменений необходимо сделать сброс станка на-

жатием на иконку «Reset Model» — и произвести симуляцию Play to

End

Содержание отчёта

Результатом проделанной работы студента, является законченный проект виртуальной модели станка.

Отчет должен содержать в себе:

- Исходные данные;
- Изображение кинематической схемы в виде дерева проекта Vericut
- Описание этапов построения
- Изображение собранного станка

Самостоятельная работа «Разработка постпроцессоров для станков с ЧПУ»

Введение

Разработка постпроцессоров является традиционной задачей – еще до появления CAD/CAM-систем постпроцессоры входили в состав САП (Систем Автоматизации Программирования), выполнявших роль средств автоматизации разработки управляющих программ для станков с ЧПУ. Геометрия обрабатываемой детали и технология обработки описывались в САП в текстовой форме (например, на языке APT), после чего выполнялся расчет траектории движения инструмента. Для результирующего описания траектории обычно использовался стандартный формат CLDATA (Cutter Locations DATA). Эти данные обрабатывались постпроцессором, который формировал управляющую программу (УП) для конкретной модели станка с ЧПУ.

Язык АРТ

Имя языка программирования АПТ (APT) происходит от начальных букв полного названия "Automatically Programmed Tools", что в переводе означает "Автоматические программируемые инструменты".

АПТ - первый язык программирования для применения в специализированной области, первый язык программирования операций обработки деталей на станках с числовым программным управлением (ЧПУ). Он разрабатывался в Массачусетском технологическом институте (МТИ) по контракту с ВВС США. Группу разработчиков в 1956 году возглавлял математик Дуглас Т. Росс.

АПТ это не только язык станков с ЧПУ, но также и программа, которая выполняет расчеты по определению положений режущего инструмента в соответствии с операторами языка. Система АПТ предназначена для программирования операций в трех мерном пространстве, причем ее можно использовать для управления станками, которые имеют до пяти управляемых осей. Ее можно использовать для управления процессами разнообразных операций механической обработки.

Язык включает четыре типа операторов: геометрические операторы, операторы движения, операторы постпроцессора и вспомогательные операторы.

Геометрические операторы (их часто называют операторами определения) определяют геометрические элементы, которые составляют деталь.

Операторы движения применяются для описания траектории перемещения режущего инструмента.

Операторы постпроцессора используются применительно к конкретному станку и его системе управления для задания скорости подачи и резания и для учета других характеристик данного станка.

Вспомогательные операторы представляют собой различные команды, применяемые для определения типа деталей, инструментов, допусков и т. п.

Мощным средством языка АПТ является возможность использования макрооператоров (макросов). Они применяются в тех случаях, когда определенная последовательность движений повторяется многократно в процессе выполнения конкретной программы. Цель использования макрооператоров заключается в уменьшении числа операторов в записи всей АПТ-программы.

В 70-е и 80-е годы появились новые подходы к составлению управляющих программ для обработки деталей на станках с ЧПУ: применение в процессе программирования интерактивной машинной графики.

При использовании интерактивной машинной графики процедура программирования реализуется на графическом терминале, оснащенном CAD/CAM или CAM системами. Используя геометрические данные, определяющие деталь в процессе автоматизированного проектирования технолог-программист формирует траекторию движения инструмента.

Выходными данными такой процедуры программирования является текст АПТ-программы, т. е. фактически массив положений режущего

инструмента (Cutter Locations DATA, CLDATA), который затем пропускается через постпроцессор, с целью получения программы в кодах станка (G- код).

Средства автоматизации разработки постпроцессоров. Система IMSpost

Самым важным результатом работы САМ-систем является управляющая программа (УП) содержащая траекторию движения инструмента при обработке. Однако эту программу нельзя напрямую использовать на станке, так как каждый тип станков имеет уникальную кинематическую схему и предъявляет различные требования - например, может производить обработку с различным количеством одновременно управляемых осей, и т.д. Кроме того, конкретный станок управляется определенной системой ЧПУ, именно она принимает на входе файл траектории инструмента и напрямую руководит перемещениями инструмента и другими действиями станка.

Но не только каждый тип станков имеет свои уникальные характеристики. Программное обеспечение системы ЧПУ также обладают различными особенностями, поэтому перед выводом на станок траектория инструмента в обязательном порядке должна быть отформатирована под конкретную комбинацию «станок - система ЧПУ», т.е. обработана постпроцессором.

С точки зрения переработки информации постпроцессор можно считать специализированным транслятором, осуществляющим перевод с языка CLDATA на язык управляющей программы (G -код). Информация о траектории движения инструмента, содержащаяся в записях CLDATA, перерабатывается постпроцессором в команды, управляющие перемещением исполнительных органов станка.

Таким образом, постпроцессор – программа, транслирующая входящий CLDATA/APT файл, в файл управляющей программы в кодах используемой системы ЧПУ и с учетом особенностей самого оборудования.

Выполняемые постпроцессором функции весьма разнообразны и выходят за пределы простого транслирования. Типовыми функциями постпроцессора являются:

- считывание данных, подготовленных процессором;
- перевод координат траектории движения инструмента в систему координат станка;
- проверка по пределам перемещений исполнительных органов станка;

- линеарилизация траектории движения инструмента с учетом кинематической схемы станка;
- назначение подачи с учетом ограничений, связанных с характером движения, допустимым диапазоном подач;
- формирование и выдача в кадр дополнительных функций;
- формирование команд на перемещение с учетом значения единицы дискретности (цены деления) УЧПУ;
- кодирование и вывод в кадр значений подач и скоростей шпинделя;
- вывод команд на включение охлаждения, команд зажимов разжимов;
- формирование команд обеспечивающих цикл смены инструмента;
- формирование команд коррекции длинны и радиуса инструмента;
- синтаксический контроль правильности задания операторов управления постпроцессором и др.

Казалось, что развитие систем ЧПУ приведет к такой унификации форматов УП, что сделает постпроцессоры ненужными. Однако, по ряду причин этого не произошло. Основная из них – стремление изготовителей оборудования с ЧПУ обеспечить пользователя собственными средствами автоматизации "ручного" программирования, реализованными в системе ЧПУ. Унификация формата УП осталась на уровне стандарта ISO, который носит достаточно общий характер и не снимает необходимости в разработке постпроцессоров. Таким образом, и сегодня постпроцессоры входят в виде модулей в состав CAD/CAM-систем (или CAM-систем, далее мы не будем делать различий).

Потребность в разработке большого числа постпроцессоров и значительная трудоемкость их разработки привели к появлению средств автоматизации проектирования постпроцессоров. Эти средства прошли свой путь развития – от библиотек стандартных подпрограмм до специальных автоматизированных систем (генераторов построцессоров).

Сегодня практически любая CAD/CAM-система имеет в своем составе собственные генераторы для автоматизации разработки постпроцессоров. Эти генераторы используют во многом общие идеи, но отличаются "глубиной" их реализации. Проблема состоит в том, что для разработчиков CAD/CAM-систем автоматизация проектирования постпроцессоров является второстепенной задачей, и основные усилия разработчиков направлены на решение других задач, в большей степени влияющих на положение их продукта на рынке. Данная ситуация привела к появлению на рынке фирм, специализирующихся на разработке таких генераторов постпроцессоров, которые могли бы встраиваться в CAD/CAM-системы или использоваться в качестве автономных средств автоматизации. Понятно, что уровень автоматизации проектирования постпроцессоров, обеспечиваемый этими генераторами, должен быть существенно выше чем тот, который предлагается разработчиками CAD/CAM.

Наиболее известной из таких фирм является компания IMS Software, Inc. (США), выпустившая на рынок универсальный генератор постпроцессоров IMSpost. Можно сказать, что эта разработка наиболее полно реализует современные идеи в области проектирования постпроцессоров, а именно:

- действия постпроцессора по преобразованию траектории инструмента в управляющую программу описываются на специальном языке высокого уровня, в котором имеется возможность оперировать параметрами траектории инструмента и управляющей программы, как понятиями языка. Этим достигается максимальная гибкость проектирования при одновременной простоте и компактности программы действий постпроцессора;
- параметры, определяющие формат кадра, начала и конца УП, подготовительные и вспомогательные функции и другие характеристики управляющей программы, задаются в специальных настроечных таблицах (диалоговых окнах), что дополнительно упрощает проектирование и модификацию (редактирование) постпроцессора. Во многих случаях для разработки нового постпроцессора достаточно выполнить изменения в диалоговых окнах постпроцессора, взятого в качестве аналога;
- с помощью специальных таблиц можно описать геометрию и взаимное расположение исполнительных органов и узлов станка с ЧПУ. Это обеспечивает автоматический расчет значений линейных и поворотных координат станка для каждого текущего положения инструмента, чем облегчается разработка постпроцессоров для многокоординатного оборудования с ЧПУ. Описание станка можно просматривать и редактировать в графическом режиме;
- наиболее сложные алгоритмы постпроцессирования встроены в ядро IMSpost, что избавляет разработчика от необходимости решать такие задачи, как проблема нелинейности при многокоординатной обработке, замена серий "мелких" участков линейных перемещений на участки с круговой или сплайновой интерполяцией и др.

IMSpost позволяет быстро и эффективно создавать постпроцессоры для любых видов оборудования с ЧПУ – фрезерных обрабатывающих центров, многокоординатного оборудования, электроэрозионных и токарных станков. Генерируемые с помощью IMSpost постпроцессоры являются автономными системами и используют в качестве входной информации данные в формате CLDATA, которые подготавливаются САМ-системой. IMSpost адаптирован к форматам CLDATA многих CAD/CAM-систем, в числе которых: CATIA, Cimatron, Euclid, MasterCAM, PowerMill, Pro/Engineer, SurfCAM, Unigraphics.

В качестве входной информации IMSpost использует файлы с данными о перемещениях инструмента (CLDATA) и преобразует их в УП, готовые для выполнения на фрезерных, токарных и электроэрозионных станках, машинах лазерной резки, листопробивных прессах и других видах оборудования с ЧПУ.

При генерации УП для конкретного оборудования IMSpost использует файл проекта (project file) (рис. 35). Файл проекта содержит большое число настроечных таблиц, которые легко корректируются через Windows-интерфейс IMSpost. Для дополнительной детальной настройки используется макроязык IMS, который может поддерживать наиболее сложные особенности и возможности современного оборудования с ЧПУ.

Рис. 35. Схема работы IMSpost

Диалоговые окна

Эти окна являются средством параметрической настройки постпроцессора. Они вызываются выбором соответствующей опции меню или нажатием на соответствующую иконку инструментальной панели IMSpost. Диалоговые окна позволяют, в частности, настроить:

- вывод нужной информации в начале и конце УП;
- правила задания в УП команд смены инструмента, охлаждения, коррекции инструмента, подачи, оборотов шпинделя, сверлильных циклов;
- правила задания перемещений при линейной, круговой и сплайновой интерполяции;
- порядок нумерации кадров УП;
- правила разбиения УП на части, с учетом имеющегося размера памяти системы ЧПУ;
- состав адресов кадра УП, порядок их вывода в кадре и правила задания информации под каждым адресом;
- правила задания подпрограмм;
- состав и правила задания подготовительных и вспомогательных функций (G- и М-функций), их разбиение по группам;
- описание состава, характеристик и взаимного расположения исполнительных органов (осей) станка.

В качестве примера на рис. 36 приведено диалоговое окно IMSpost для настройки формирования постпроцессором команд круговой интерполяции. Здесь можно указать, какие G-функции используются для задания направления движения по и против часовой стрелки, способ задания начальной точки дуги (относительно конечной точки, относительно центра, с помощью радиуса), минимальное и максимально допустимое значение радиуса при круговой интерполяции (в противном случае выполняется линейная аппроксимация дуги), точность линейной аппроксимации, наличие или отсутствие ограничения системы ЧПУ на расположение дуги в одном квадранте (при наличии ограничения будет выполнена разбивка дуги по квадрантам), возможность задания круговой интерполяции в произвольной плоскости.

Диалоговые окна IMSpost отличаются большим числом различных параметров и продуманными наборами их вариантов, которые учитывают самые разнообразные способы задания команд УП в различных моделях оборудования с ЧПУ. Тем не менее, никакая параметрическая настройка не гарантирует того, что разрабатываемый постпроцессор сможет учесть все особенности конкретного оборудования с ЧПУ. Единственным гарантированным способом здесь является использования языка программирования.

C:\lms\IMSpost65b\PRJFILES\a_proj.lib	
<u>File Execute General Controller Machine F</u>	<u>R</u> eport User <u>H</u> elp
Motion Parameters	X
	1
Clockwise code used :	
Counter-clockwise code used : G3	(, ijk) (^{ijk} , ←)
Radius format	
C LUK : absolute center coordinates	
TUK : distance from circle start to circle center	
C LLK - unsigned distance from carble to circle start	
Protect and a standard from Center to start	
C Break into linear moves	
Linearization Parameters	
Maximum radius for circular interpolation	
Minimum radius for circular interpolation	
Break up circle on quadrant boundaries	
Helical interpolation supported	
Support circular interpolation in any plane	Help
	ОК Отмена

Рис. 36. Диалоговое окно IMSpost для настройки команд круговой интерполяции

Макро язык и макросы

В IMSpost используется высокоуровневый язык программирования (макроязык), с помощью которого описываются правила преобразования траектории движения инструмента, заданной в файле CLDATA, в управляющую программу. Программы на макроязыке (макросы) составляются так, что каждому виду оператора CLDATA соответствует свой макрос. При вызове постпроцессора он последовательно обрабатывает операторы файла CLDATA с помощью своих макросов и формирует команды УП. Операторы макроязыка IMSpost позволяют передавать параметры, использовать локальные и глобальные переменные, работать с внешними файлами, получать доступ к параметрам оборудования с ЧПУ (заданным в диалоговых окнах), выполнять арифметические и логические операции, использовать тригонометрические и другие стандартные математические функции, выполнять условные и безусловные переходы, работать с текстовыми строками и др.

Группа операторов макроязыка IMSpost представляет собой специальные функции, реализующие достаточно сложные вычислительные процедуры, например замену серий «мелких» участков линейных перемещений на участки с круговой или сплайновой интерполяцией.

При создании нового постпроцессора в него автоматически включается набор макросов по умолчанию. Эти макросы можно редактировать, удалять или заменять, импортируя макросы из других постпроцессоров. Отладка макросов производится с помощью отладчика (Debugger) IMSpost, который позволяет выполнять макросы в пошаговом режиме, просматривать текущие состояния локальных и глобальных переменных и т.д. В качестве теста для отладки может использоваться файл CLDATA, сформированный CAD/CAM-системой, или любой набор операторов CLDATA, набранный в текстовом редакторе.

Макросы устанавливают связь между входными операторами CLDATA и формируемой постпроцессором управляющей программой. Вы используете программирование макросов для определения внутренней логики вашего проекта, от которой зависят решения по формированию тех или иных последовательностей кадров, которые соответствуют требованиям оборудования с ЧПУ и особенностям его эксплуатации на предприятии.

Макрос IMS представляет собой последовательность операторов, которые задают для базы знаний IMS правила, как реагировать на полученную входную информацию. Каждый макрос начинается с имени макроса (оператора определения) и заканчивается оператором RETURN. Следующий пример иллюстрирует простейший макрос:

TEST/*

GLOBAL.VAR = 2.5 REGISTER.X.VALUE = GLOBAL.VAR OUTPUT(REGISTER.X, NEWLIN) RETURN

В данном примере макрос имеет имя TEST. Второй и третий операторы являются примерами использования арифметических выражений. Четвертый оператор – это пример макрооператора вывода OUTPUT. Макрос заканчивается оператором RETURN.

Арифметические операции

Порядок, в котором операции выполняются в выражении, зависит от их расположения и приоритета. Например, оба этих выражения:

2 * 3 + 4 4 + 2 * 3

дают в результате значение 10, так как операция умножения имеет более высокий приоритет и выполняется первой. Две операции с одинаковым приоритетом выполняются в порядке слева направо. Установлены следующие приоритеты:

*,/	высший
+,-	
AND	
OR	
= , < , >	низший

Для задания порядка выполнения операций могут использоваться круглые скобки (). Терм выражения, заключенный в скобки, вычисляется первым. Например, выражение (4 + 2) * 3 равно 18, а не 10. Используйте скобки в сложных выражениях, чтобы сделать порядок вычислений более очевидным.

Математические функции

Допускается использование в выражениях следующих математических функций:

Функция	Описание	
SIN(X)	Синус Х (Х в град)	
COS(X)	Косинус Х (Х в град)	
TAN(X)	Тангенс Х (Х в град)	
ASIN(X)	Арксинус Х	
ACOS(X)	Арккосинус Х	
ATAN(X)	Арктангенс Х	
ATAN2(X)	Обратный тангенс частного от деления ар-	
	гументов	
SQRT(X)	Квадратный корень от Х	
ABS(X)	Абсолютная величина Х	
INT(X)	Целая часть X (X может быть также сим-	
	вольной строкой)	
SINH(X)	Гиперболический синус Х	
COSH(X)	Гиперболический косинус Х	

TANH(X)	Гиперболический тангенс Х
EXP(X)	Экспонента
LOG(X)	Натуральный логарифм
LOG10(X)	Десятичный логарифм (X > 10)
POW(X,Y)	Х в степени Y; $A = POW(2,3)$ означает
	A=2*2*2
CEIL(X)	Наименьшее целое, не меньше чем Х
FLOOR(X)	Наибольшее целое, не больше чем Х
MIN(X1,,Xn)	Выбор наименьшего Х
MAX(X1,Xn)	Выбор наибольшего Х
UCASE(S)	Регистр прописных букв для строки S
LCASE(S)	Регистр строчных букв для строки S
STRLEN(S)	Длина строки S
SPRINT(Form	Отформатированная строка
at, V1,Vn)	
MOD(X,Y)	Остаток от деления Х/Ү со знаком Х

Примечание: вы не можете использовать в макросах имена переменных, совпадающие с именами математических функций. Эти имена зарезервированы в IMSpost.

Операторы присваивания

Операторы присваивания являются наиболее распространенным видом операторов, используемых в макросах. Эти операторы имеют следующий формат:

Переменная = Выражение

Где выражение может быть простым или сложным, лишь бы оно порождало правильный результат.

Макросы IMS используют два типа переменных: локальные и глобальные. Локальные переменные используются только внутри конкретного макроса; глобальные переменные доступны в любом макросе постпроцессора. Имена глобальных переменных должны начинаться со слова GLOBAL. Вы можете индексировать переменные любого из этих типов.

Примеры:

```
X = CLDATAN.1 GLOBAL.OX + GLOBAL.TX
MODE.COOLNT = "MIST"
PTR1 = LOCATE("G98")
a=1
tool = cldatan.1 + offset
```

global.rpm = cldatan.right.rpmdelta = sqrt ((X*X) + (Y*Y))global.tool[t] = cldatan.1

Не разрешается использовать символы точки (.), запятой (,), одинарных или двойных кавычек в именах локальных переменных. Система использует эти символы в качестве разделителей и для распознавания имен переменных и текстовых строк.

В операторе присваивания макроязыка вы можете задавать или изменять переменную так, что изменение будет действовать только в текущем макросе. Когда выполнение текущего макроса завершено, значение переменной будет возвращено к ее значению до "разового" изменения. Это "разовое" изменение достигается с помощью записи символа '!' перед переменной в операторе присваивания. Например:

TEST/* GLOBAL.A = 3.0Set variable GLOBAL.A *Set GLOBAL.A using "one-shot" !GLOBAL.A = 999.999 * Set VALUE_1 to equal -999.999 VALUE_1 = GLOBAL.A * (-1.0) RETURN

В данном примере переменная GLOBAL. А равна 999.999, пока она внутри макроса. После завершения работы макроса, она восстанавливает значение, равное 3.

Классы переменных

Для контроля переменных и управления ими в базе знаний, какие из них находятся на хранении, к каким есть доступ внутри макроса, а какие передаются из одного макроса в другой. Вам нужно понимать принципы использования переменных в макроязыке IMS. В нем существуют различные классы переменных:

SYSTEM	База знаний IMS создает и поддерживает сис-
	темные переменные. Вы получаете к ним доступ с
	использованием зарезервированного слова SYSTEM
	перед именем переменной (например,
	SYSTEM.DATE). К этим переменным можно полу-
	чить доступ всегда в любом макросе.
GLOBAL	Вы создаете и поддерживаете глобальные пере-
	менные. Для этого используйте зарезервированное

	слово GLOBAL перед именем переменной (напри- мер, GLOBAL.TEST). После того как вы определили глобальные переменные, их можно использовать в любом макросе.
MODE	Определяйте эти переменные с использованием слова МОДЕ для группы и подгруппы (например, MODE.MOTION or MODE.MOTION.LINEAR). В макросе вы получаете доступ к текущему состоянию переменной или ее функциональной подгруппы так же, как в случае глобальной переменной.
LOCAL	Вы можете создавать и поддерживать локальные переменные. Однако Вы можете использовать их только в том макросе, где Вы их определили. Для за- дания локальных переменных не используется заре- зервированное слово; вы просто задаете имя пере- менной (например, TEST1). После выполнения мак- роса, все локальные переменные считаются далее не определенными. Чтобы использовать их повторно, вы должны снова определить их в другом макросе или макросах.
CLDATA	Эти переменные являются разновидностью сис- темных переменных. Система устанавливает значе- ния этих переменных на время текущего выполнения макроса в зависимости от входных аргументов или информации CLDATA, передаваемой в макрос. Этот тип переменных поддерживается двумя зарезервиро- ванными словами: CLDATAN (числа) и CLDATAM (вспомогательные слова).
CLREAD	Эти переменные являются разновидностью сис- темных переменных. Система устанавливает значе- ния этих переменных на время текущего выполнения макроса в зависимости от входных аргументов или информации CLDATA, которая найдена оператором CLREAD при просмотре входной информации "впе- ред". Этот тип переменных поддерживается двумя зарезервированными словами: CLREADN (числа) и CLREADM (вспомогательные слова).
REGISTER	Эти переменные соответствуют адресам кадра, которые определяются в таблице Register Format ta- ble. Например, REGISTER.Z.MAXIMUM = 5000. Полный список этих переменных приведен в HELP файле

COMPONENT	Переменные, связанные с описанием узлов станка в диалоговом окне Machine Components. Полный список этих переменных приведен в приложении к данному руководству.
SOLUTION	Переменные, связанные с программами SOLUTION. Обратитесь к описанию макрооператора SOLUTION с примерами переменных (документация находится во вкладке Help).
FUNCTION	Переменные, связанные с описанием функций системы ЧПУ в диалоговом окне Function Definition. Эти переменные могут определять любое поле дан- ного диалогового окна, например: FUNCTION.G1.NAME="G1" FUNCTION.G1.USE="X,Y,Z,S,F,M" FUNCTION.G4.USE1="P" (USE1 означает один кадр) FUNCTION.G0.USE=FUNCTION.G0.USE+,"ААА" (добавляет строку "ААА" в поле USE)
MACHINE	Переменные, связанные с описанием осей стан- ка в диалоговом окне Machine Motion. Вы можете изменять любое из значений, например, MOTION.Z.HOME=3000

Работа с макросами

Каждый проект или постпроцессор включает свой собственный набор макросов. Когда вы начинаете новый проект, то он автоматически включает в себя набор макросов по умолчанию. Диалоговое окно Macro Manager отображает только те макросы, которые содержатся в открытом проекте.

В диалоговом окне Macro Manager вы можете импортировать макросы из другого проекта в открытый проект. У вас также имеется опция для добавления макроса, только в этом случае вы можете вставить его, если макрос существует в другом проекте.

Макросы IMS имеют синтаксис высокоуровневого языка программирования. Компоненты языка включают:

- Передачу параметров
- Определяемые пользователем процедуры (макросы)
- Доступ к внешним файлам или программам
- Локальные и глобальные переменные

- Условные операторы и операторы циклов
- Доступ к параметрам системы ЧПУ и станка
- Возможность обрабатывать входные операторы APT/CL (CLDATA)
- Функции для чтения и записи в файлы
- Функции предварительного просмотра
- Арифметические, тригонометрические и логарифмические функции

Просмотр макросов

- 1. Создайте новый или откройте существующий проект.
- 2. Выберите опцию Macro Manager из меню General для открытия диалогового окна Macro Manager. Вы можете также использовать

клавиши Ctrl M или нажать иконку Macros на инструментальной панели. Система отобразит на экране набор макросов, которые имеются в активном проекте (постпроцессоре).

Редактирование макросов

- 1. Создайте новый или откройте существующий проект.
- 2. Выберите опцию Macro Manager из меню General для открытия диалогового окна Macro Manager (рис. 37). Вы можете также исполь-

зовать клавиши Ctrl M или нажать иконку Macros на инструментальной панели.

- 3. Выберите макрос и нажмите кнопку Edit. Можно также дважды щелкнуть мышью на нужном макросе. Когда выбранный макрос появится на экране, он будет включать выбор команд File, Edit и Tab. Вы можете выбрать более одного макроса при нажатой клавише Shift или используя клавишу Ctrl.
 - File вы можете сохранить или отменить сделанные вами изменения
 - Edit доступны операции Undo, Cut, Copy и Paste
 - **Таb** автоматическое форматирование с целью упрощенного ввода. Нажатие Таb будет форматировать макрос с отступами для определенных операторов, таких как IF, WHILE и CASE.
- 4. Отредактируйте макрос или его описание и сохраните изменения

Рис. 37. Окно Macro Manager

5. с помощью выбора опции Save меню File окна редактирования макроса.

Удаление макросов

- 1. Создайте новый или откройте существующий проект.
- 2. Выберите опцию **Macro Manager** из меню **General** для открытия диалогового окна Macro Manager. Вы можете также использовать

клавиши Ctrl M или нажать иконку Macros на инструменталь-

- 3. Выберите удаляемый макрос и нажмите кнопку Delete
- 4. Подтвердите удаление нажатием ответа Yes.

Создание макроса

- 1. Создайте новый или откройте существующий проект.
- 2. Выберите опцию Macro Manager из меню General для открытия диалогового окна Macro Manager. Вы можете также использовать

клавиши Ctrl M или нажать иконку Macros на инструменталь-

3. Нажмите кнопку New

- 4. Введите имя макроса и нажмите ОК. Вам не нужно добавлять символ ·// к имени макроса. Система добавит его автоматически.
- 5. Введите операторы макроса в открывшееся окно редактирования макроса. Вы можете также вводить описание макроса. По завершении сохраните макрос с помощью выбора опции Save меню File окна редактирования макроса.

Примечание: по умолчанию, когда вы сохраняете новый макрос, система добавляет к имени групповой символ '*'. Этот символ инструктирует постпроцессор выбирать в файле CLDATA все главные слова с таким же именем. Например, PROBE /* принимает все операторы PROBE из файла CLDATA.

Вы можете сделать вызовы макроса более специальными. Например,

PROBE/# - должно иметься то же самое значение

PROBE/#,#,XOFF,# - значение, значение, вспомогательное слово XOFF, значение

PROBE/ZOFF,* - вспомогательное слово ZOFF с чем-либо, иначе следующий оператор.

Помните, что если какой-либо из операторов CLDATA не входит в число просматриваемых вашим макросом, то он будет игнорироваться.

Импорт макроса

- 1. Создайте новый или откройте существующий проект.
- 2. Нажмите кнопку Import для открытия окна Open Import File (рис. 38).
- 3. Выберите проект, который содержит макросы, которые вы хотите импортировать и нажмите Open или дважды щелкните мышью на имени файла.
- 4. В открывшемся диалоговом окне Import Macro (рис. 39) выполните любое из следующих действий:
 - Нажмите All для выбора всех макросов
 - Нажмите None для отмены выбора всех макросов
 - Нажмите View для просмотра выбранного макроса

• Отметьте окошко Verify для выдачи подтверждения в случае, если импортируемый макрос уже содержится в открытом проекте

Open Import I	File				? ×		
Look jn:	🔄 Prifiles	•	£	Ċ			
🔊 fagor800_lib.prj							
🖻 fanuc16c_lib.prj							
📄 hurco-isnc	🔊 hurco-isnc_lib.prj						
🛯 🔊 tiger6_lib.p	🗃 tiger6_lib.prj						
File <u>n</u> ame:					<u>O</u> pen		
Files of <u>type</u> :	Project Files (*.PRJ)		•		Cancel		
	Dpen as read-only			_			

Import Macro C:\IMS	\IMSpost60\PRJFILES	\milling_def.prj		×				
ACCURACY/*	FINDSETUP/*	OUTCUTCOM/*	SUBCALL/*	<u>o</u> k				
AUXFUN/*	FINI_CFG/*	PARTNO/*	SUBDEFINE/					
CIR2LIN/*	FROM/*	PATH_NAME_EXT/*	SUBMOVE/*	Cancel				
CIRCLE/*	GETPARTNO/*	PL_CHNG/*	SUBORIGIN/					
COMENT/*	GOTO/*	PLANE/*	SURFACE/*	Help				
COOLNT/*	INIT7*	PNT2VEC/*	TEST/*					
CUTCOM/*	INIT_CFG/*	PPRINT/*	TOOLHOME/					
CUTFILE_END/*	INSERT/*	PREFUN/*	TRANS/*					
CUTFILE_START/*	LINTOL/*	PRESELECT/*	UCS/*					
CUTLENGTH/*	LOADTL/*	PRESELECTION/*	UCS_MULTI/					
CUTTER/*	MATRIX/*	PROCEDURE/*	UCS_SAVE/*					
CUTWITH/*	MESSAGE/*	RAPID7	UNFORCE/*					
CYCLE/*	MODE/*	REWIND/*	UNIT/*					
CYCLMOTN/*	MOVEHOME/*	SEQNO/*	UNITREG/*					
CYCLMOVE/*	MULTAX/*	SEQNOON/*	USE1ADD/*					
CYCLPAR/*	NEXT_PROCEDURE/*	SETUNIT/*	USE1SET/*					
CYCLRET/*	NEXT_TOOLPATH/*	SETUP/*	XFORCE/*					
DELAY/*	NORMAL/*	SIMD07*	XSETVAR/*					
DISPLY/*	NURBS/*	SIMFILL/*						
END/*	OPSKIP/*	SIMINIT/*						
EXECBATCH/*	OPSTOP/*	SPINDL/*						
FEDRAT/*	ORIGIN/*	STOP/*						
•			F	Verify				
All	None View							

Рис. 39. Окно Import Macro
5. Нажмите ОК для импорта выбранных макросов.

Конкретные макросы, которые вы выбрали, копируются из старого проекта в текущий проект.

Примечание: При выборе более одного макроса нажмите клавишу Ctrl и выполняйте групповой выбор.

Поиск Макросов



Кнопка поиска Search дает вам возможность ввести символьную строку. Система отобразит все области в активном постпроцессоре, где будет найдена эта строка.

- 1. Создайте новый или откройте существующий проект.
- 2. Выберите опцию Macro Manager из меню General для открытия диалогового окна Macro Manager. Вы можете также использовать

клавиши Ctrl M или нажать иконку Macros на инструментальной панели.



- 3. Нажмите кнопку Search . В появившемся окне (рис. 40) поиска введите текст строки, которую вы хотите найти. Нажмите кнопку Find. Система выполнит просмотр всех разделов на предмет содержания данной строки и отобразит их в таблице. Вы можете последовательно просмотреть результаты поиска и перейти в конкретный макрос, сделав двойной щелчок мышью на выбранном пункте.
- 4. Отметьте окошко Match Case, если вы хотите, чтобы система искала далее только в том списке, который был найден (выше или ниже).
- 5. Отметьте окошко Match complete word, если вы хотите, чтобы система использовала при поиске полную строку. Если нет, то система будет находить и отображать все места, где заданная строка была найдена.

Macro Search	×
global.tool	Find
Match Case	Match complete word
Macro Lir	ne Text
COOLNT/* 17	/ IF ((GLOBAL.TOOLCHG_COOLON) AND (GLOBAL.T)
CUTLENGTH/* 16	REGISTER.[GLOBAL.TOOLCHG_LREG].VALUE
CUTTER/* 2	GLOBAL.TOOLCNT = GLOBAL.TOOLCNT + 1
CUTTER/* 2	GLOBAL.TOOLCNT = GLOBAL.TOOLCNT + 1
CUTTER/* 14	CALL(LOADTL/GLOBAL.TOOLCNT)
FINI_CFG/* 2	IF ((GLOBAL.TOOLCHG_REFIRST) AND (GLOBAL.F.
FINI_CFG/* 3	REGISTER.[GLOBAL.TOOLCHG_TREG].VALUE =
INIT7* 17	GLOBAL.TOOLCNT = 0
•	
42 occurrence(s)	have been found.

Рис. 40. Окно поиска Macro Search

Модель станка и её использование

Описание модели станка (состава, характеристик и взаимного расположения исполнительных органов) наиболее важно для многокоординатного оборудования, так как избавляет разработчика постпроцессора от необходимости решения ряда нетривиальных математических задач. Напомним, что при многокоординатной обработке текущее положение инструмента в траектории описывается в виде (x, y, z, i, j, k), где (x, y, z) – координаты центра торца инструмента, а (i, j, k) – орты, определяющие положение оси инструмента в системе координат детали. Постпроцессор должен (например, для 5-координатного станка с тремя линейными осями X, Y, Z и двумя поворотными осями B, C) выполнить преобразование $(x, y, z, i, j, k) \rightarrow (X, Y, Z, B, C)$, что требует от разработчика постпроцессора вывода соответствующих математических зависимостей. Те разработчики, которые сталкивались с подобной задачей, знают, что ее решение требует значительных усилий, особенно при отсутствии опыта.

В IMSpost эта задача решается принципиально иначе, благодаря наличию модели станка. Для создания модели необходимо описать в диалоговых окнах состав осей станка (окно Machine Motion), состав и параметры узлов станка (окно Machine Components) и взаимное расположение узлов станка (окно Machine Reference). На основе этих описаний IMSpost формирует пространственную модель станка, которую можно просматривать и редактировать в графическом режиме (рис. 41). При работе постпроцессора текущие координаты исполнительных органов станка определяются на основе заданного текущего положения инструмента автоматически, благодаря использованию модели. Для получения координат станка (в макросе обработки перемещений инструмента) используется специальный оператор макроязыка.

Модель станка позволяет автоматически решать еще одну сложную задачу – так называемую проблему нелинейности. Проблема состоит в том, что последовательные положения в траектории инструмента рассчитываются в CAD/CAM из предположения линейности перемещения инструмента между этими положениями. Однако, используемый в системах ЧПУ режим линейной интерполяции при отработке кадра приводит к тому, что инструмент движется по криволинейной траектории (рис. 42,а). В результате этого возможны зарезания и брак обрабатываемой детали.

Решение проблемы нелинейности является функцией постпроцессора и заключается в следующем. Если отклонение инструмента от линейной траектории превышает заданный допуск, то постпроцессор формирует дополнительное положение инструмента в середине текущего участка, удовлетворяющее условиям линейности. После этого рассчитывается отклонение для каждого из полученных участков (рис. 42,б). Если отклонение еще превышает допуск, то производится дополнительное деление и т.д., до тех пор, пока отклонение не окажется меньше допуска.



 $(x, y, z, i, j, k) \longrightarrow (X, Y, Z, B, C)$

Рис. 41. Схема формирования координат исполнительных органов станка в постпроцессоре IMSpost

Проблема нелинейности является значительно более сложной задачей, чем расчет координат исполнительных органов станка. Кроме того, при ее "прямом" решении путем написания соответствующей программы, возникают достаточно высокие требования к используемым средствам программирования. Далеко не все генераторы постпроцессоров имеют в своем распоряжении такие средства.



Рис. 42. Проблема нелинейности и схема ее решения

В генераторе IMSpost наличие модели станка позволяет решать проблему нелинейности автоматически. При этом, для получения массива дополнительных промежуточных точек, в макросе обработки перемещений инструмента используется специальный оператор макроязыка. Допуск отклонения от линейной траектории устанавливается в макросе инициализации, который выполняется в начале работы постпроцессора.

Конечно, освоение генератора IMSpost, как и любой серьезной системы, требует определенных усилий. Однако, эти усилия быстро окупятся за счет резкого сокращения времени проектирования постпроцессоров. Кроме того, с помощью IMSpost разработчик постпроцессоров получает возможность оперативно учитывать практически любые требования и пожелания цеховых технологов к поступающим управляющим программам.

Содержание отчёта

Результатом проделанной работы студента, является выполнение полученного задания.

Отчет должен содержать в себе:

- Исходные данные;
- Схему кинематики станка
- Описание выполнения
- Выходные данные

Самостоятельная работа «Разработка технологического процесса изготовления детали в PDM системе SMARTEAM»

Разработка технологических процессов в PDMсистеме

Современное производство использует самый широкий спектр технологий при изготовлении деталей изделий. При неавтоматизированной подготовке производства, технологические процессы разрабатываются непосредственно в виде комплектов технологической документации. При использовании автоматизированных систем ТПП, создаваемые описания технологических процессов размещаются в компьютерной базе данных, а соответствующая документация является лишь отображением внутреннего представления ТП во внешнюю сферу. Хранящиеся в компьютерной базе ТП являются основным источником информации для решения задач автоматизированного управления технологической подготовки производства. При этом разработка ТП выполняется с помощью специальных систем автоматизированного проектирования ТП (САПР).

Автоматизация проектирования технологических процессов обеспечивается путем разработки специальных приложений на основе использования инструментальных средств PLM. Эти приложения должны быть тесно интегрированы с PDM-системой с целью обеспечения работы технолога в среде ЕИП, а также с целью устранения дублирования баз данных по технологическому оборудованию, оснастке и инструменту. Такая интеграция, в частности, может быть достигнута за счет непосредственной реализации функций проектирования ТП в среде PDMсистемы средствами API. Практическая реализация такой САПР ТП выполнена средствами API PDM-системы SMARTEAM. При проектировании ТП в PDM-системе SMARTEAM используются следующие группы классов:



• Нормативно-технологическая документация

Проектируемый ТП формируется в виде некоторой информационной модели. Благодаря реализованному в РМD SMARTEAM объектноориентированному подходу информационная модель ТП не содержит в себе непосредственно тех данных, которые были «взяты» из базы при проектировании ТП (например, наименования операций и переходов, сведения о режущем инструменте, нормативно-справочная информация и др.). Она содержит в себе только ссылки – специальные указатели на местоположение этих данных. Это позволяет исключить ненужное дублирование информации. Любое изменение элемента базы данных (например, наименования приспособления или инструкции по технике безопасности) приводит к тому, что эта информация автоматически «заменяется» во всех разработанных ТП. Тем самым автоматически обеспечивается соответствие информации, содержащейся в ТП, информации различных разделов базы данных

Заполнение базы данных (сведениями по оборудованию, оснастке, инструменту, наименованиям операций и переходов) может осуществляться двумя способами:

- а) непосредственным занесением информации пользователем (в интерактивном режиме) в соответствующие разделы базы данных;
- b) автоматическим занесением информации в базу данных, в процессе разработки индивидуальных ТП.

Второй метод применяется тогда, когда технолог при проектировании ТП не нашел в базе данных необходимой информации, и вынужден ввести ее в ТП вручную. В этом случае он может дать команду на запись введенной информации в базу данных. Такое занесение избавляет пользователей от необходимости повторного ввода данных в интерактивном режиме и способствует постепенному автоматическому формированию базы данных.

Проектируемый в PDM SMARTEAM TП, представляется в виде дерева, основными узлами которого являются номера операций и номера переходов, в состав созданных объектов вводят наименования операций, их содержание, оборудование и другие компоненты выполняемых технологических операций и переходов.

Технологический процесс представляется в системе в виде древовидной структуры, узлами которой являются номера операций и номера переходов, в состав этих объектов вводят наименования операций, их содержание, оборудование и другие компоненты ТП. Проектирование ТП выполняется как стандартный процесс построения дерева проекта (рис. 43).



Рис. 43. Структура технологического процесса в PDM SMARTEAM

Можно добавлять, удалять или заменять узлы дерева, описывать их конкретные характеристики. При этом определяется последовательность операций в маршрутном ТП, последовательность переходов в операционных ТП, выбирается необходимое оборудование, приспособления и инструмент, указываются режимы обработки. Все эти действия осуществляются с использованием соответствующих разделов базы данных. Например, при добавлении в дерево ТП новой операции, технологи выбирает наименование операции из предлагаемого системой списка. При выборе, например, режущего инструмента для перехода «Точить поверхность», технолог просматривает списки имеющихся резцов и т.д.

Для проектирования ТП в PDM SMARTEAM разработан специальный графический интерфейс (рис. 44), использование которого позволяет:



Рис. 44. Общий вид графического интерфейса

- обеспечить наглядное представление структуры ТП в соответствии с требованиями стандартов ЕСТД;
- показать оптимальную последовательность описания ТП;
- упростить и ускорить процесс, за счет повышения эргономичности.

Средства настройки системы SMARTEAM позволяют реализовывать так называемый направленный поиск информации. Этот поиск используется при автоматическом формировании списков данных, которые предлагаются технологу при решении задач выбора оборудования, оснастки, технологических переходов и других элементов ТП. Направленный поиск позволяет получать такие списки, которые «не содержат ничего лишнего». Так при выборе оборудования в токарной операции предлагается только список токарных станков (но не фрезерных или электроэрозионных), при выборе очередного перехода в токарной операции предлагаются только возможные для данной операции переходы и т.д. пользователь может сам легко связывать те или иные группы объектов для обеспечения направленного поиска.

После того, как проектирование ТП завершено, выполняется его нормирование и ТП поступает на утверждение. Комплект технологических документов (ТД) размещается в классе «Технологическая документация» и включает в себя маршрутную карту (описание маршрута), операционные карты (описание операций), ведомость оснастки (перечень используемых средств технологического оснащения) и ряд других документов, оформленных в соответствии с ЕСТД. По запросу пользователя, комплект ТД или отдельные документы могут быть выведены на печать.

Вся необходимая текстовая и текстово-графическая технологическая документация по требуемым формам формируется автоматически на основании созданного ТП. По запросу технолога в БД создается объект, все атрибуты которого заполняются автоматически, после чего к нему прикрепляется файл созданного документа. Все технологические документы собираются в комплекты.

Некоторые технологические документы по своему содержанию являются не текстовыми, а текстово-графическими (графическими). К таким документам относятся карта эскизов, карта наладки и карта заказа оснастки.

Формирование графических документов ТП осуществляется в два этапа:

- а) Проектирование графического изображения (эскиза) в CADсистеме или его сканирование с готового чертежа.
- b) Формирование текстовой части документа на соответствующем макете бланка и включение графического изображения в документ в PDM-системе.

Для карт эскизов в системе должны быть созданы объекты с прикрепленными файлами эскизов, выполненными в одной из CAD-систем. При проектировании эскизов могут быть заимствованы модели или чертежи изделий, разработанные конструктором. Вывод ТД на печать возможен по одному документу или комплектами. Система поддерживает стандарты ЕСТД, а также допускает возможность настройки на пользовательские форматы документов. Если в ТП используется оборудование с ЧПУ, то технолог создает заказ на проектирование управляющих программ (УП) обработки деталей. Подразделение, занимающееся проектированием УП, может использовать геометрические модели деталей, разработанные конструктором, или модели с предыдущих операций данного ТП. Управляющие программы для оборудования с ЧПУ, разработанные в CAD/CAMсистеме, также являются "частью" ТП механообработки. А именно, каждая управляющая программа (УП) является "частью" соответствующего операционного ТП для станка с ЧПУ. Поэтому УП присутствуют в общей модели ТП и хранятся в классе "Технологические процессы". Кроме того, оформленные на бланках тексты УП хранятся в классе "Технологическая документация".

Технологические процессы по различным видам обработки данного изделия могут разрабатываться параллельно вне зависимости от общей технологической последовательности, но затем имеющийся в системе механизм поддержки "сквозных" технологических процессов позволяет объединить эти ТП в единый технологический маршрут и автоматически вывести общую маршрутную карту.

Так можно сформировать информационную модель изделия, выпускаемого предприятием. Фрагмент информационной модели изделия приведен на рис. 45.

Вся информация, полученная в ходе проектирования и технологической подготовки производства изделия, хранится в БД и в дальнейшем может быть передана в системы управления производством MRP/ERP через специальные интерфейсы, имеющиеся в PDM-системе SmarTeam.



Рис. 45. Информационная модель проекта

Задание

В данной работе студентам необходимо самостоятельно разработать технологический процесс изготовления детали при помощи средств PDM-системы SMARTEAM.

Для проектирования технологического процесса в PDM-системе SMARTEAM необходимо выполнить следующую последовательность действий:

- 1. Создать объект «деталь».
- 2. Создать заготовку.
- 3. Указать материал.
- 4. Спроектировать расцеховку.
- 5. Сформировать маршрут изготовления детали.
- 6. Описать операции и переходы.
- 7. Занести в базу PDM-системы SMARTEAM оборудование и инструменты, необходимые для изготовления детали.
- 8. Сформировать комплект технологической документации.

Результатом выполнения работы является технологический процесс, сформированный в PDM-системе SMARTEAM. Набор технологической документации: маршрутная карта, операционная карта. Каждая операционная карта должна содержать описание переходов и операционный эскиз.

Проектирование технологического процесса

Создание ТП осуществляется методом индивидуального проектирования, который заключается в ручной компоновке необходимых последовательностей операций и переходов с использованием имеющихся в системе библиотек. Они содержат списки наименований операций и переходов, перечни применяемого оборудования, приспособлений, режущего, вспомогательного и мерительного инструмента. Индивидуальное проектирование предполагает также возможность использования разработанных ранее ТП в качестве аналогов. При этом проектирование сводится, по существу, к редактированию уже имеющегося ТП.

Библиотеки наименований и содержания операций, оборудования, оснастки и инструмента, вспомогательных материалов и других компонентов ТП доступны по «горячим клавишам» в пределах всей базы данных. Каждый новый компонент ТП, введённый любым из технологов в "свой" ТП, автоматически добавляется в соответствующую библиотеку и может быть заимствован любым другим технологом для нового ТП. В общем случае проектирование технологического процесса в SMARTEAM можно представить состоящим из следующих этапов:

- 1. Создание проекта.
- 2. Создание детали.
- 3. Создание заготовки.
- 4. Разработка ТП.
- 5. Наполнение базы данных инструмента и технологического оборудования.
- 6. Проектирование технологической документации.

Отображение дерева ТП со всеми логическими связями.

Создание объекта «деталь»

Прежде, чем начать проектирование технологического процесса, необходимо в PDM-системе SMARTEAM создать объект «деталь». Затем передать данные (обозначение и наименование детали) в ТП, используя кнопку, находящуюся на странице паспорта детали

Передать данные в ТП . Это необходимо проделать для того, чтобы система присвоила уникальный номер данному ТП и его операциям, что позволит избежать дублирования данных во всей системе.

Создание заготовки

На данном этапе в состав детали включается заготовка при помощи

кнопки Создать заготовку, расположенной на титульном листе данной детали.

При создании объекта «Заготовка» автоматически устанавливается логическая связь этого объекта с «Проектом (изделием)», количество заготовок принимается равным единице и автоматически вводится на страницу **Composition**.

Далее требуется назначить материал, который используется для изготовления детали. Выбор всех материалов производится из библиотек материалов. Информацию о материалах заготовок вводят на закладке «Титульный лист-Расчет М на заготовку». Активизация кнопки «Выбрать материал» выводит на экран классификатор материалов. На рис. 46 приведено окно, содержащее перечень материалов SMARTEAM.

Для того чтобы начать проектировать ТП необходимо перейти в класс ТП созданной вами детали. Как уже говорилось ранее, для проектирования ТП разработан специальный графический интерфейс (рис.

44), расположенный на закладке «**Титульный лист** – **хТП**». Так как на первом этапе была осуществлена передача данных в ТП, в классе ТП автоматически сформировались два объекта:

- а) папка «Сумма технологий с обозначением проекта»;
- b) техпроцесс «Расцеховка».



Рис. 46. Перечень материалов

Проектирование «расцеховок»

Объект – «расцеховка» (Рц) на стадии технологической подготовки производства изделий создается автоматически в составе папок «Сумма технологий» в группе классов «Технологические процессы» при активизации программы «Передать данные в ТП», «кнопка» запуска которой расположена на ТЛ изделий. При помощи графического интерфейса для проектирования ТП, формируется «расцеховка» – последовательность цехов изготовителей, которая задается при помощи кнопки «№ этапа».

После создания этапов, используя кнопку

необходимо получить «Межцеховой маршрут». После выполнения данной операции «Межцеховой маршрут» появится в дереве ТП.

🌃 Расцеховка (МЦМ)

Проектирование «сквозного» ТП

В системе SMARTEAM существует два способа проектирования ТП:

- Проектирование «сквозных» ТП;
- Проектирование ТП по методам обработки.

«Сквозной» ТП - это ТП одного из методов обработки, в котором описана еще и общая технологическая последовательность обработки изделия по всем методам обработки.

Проектирование ТП по методам обработки – это ТП, в котором формируются последовательность операций и переходов всех методов обработки, входящих в состав данного ТП.

Технологический процесс формируется при помощи кнопки «Номер сквозного ТП или ТП метода обработки».

На данном этапе выполнения работы необходимо ввести следующие данные:

- Вид ТП по охвату.
- Вид ТП по организации.
- Вид ТП по методу обработки.

Проектирование операций и переходов

При помощи кнопки «№ операции» в дерево проекта добавляется операция, которой присваивается свой уникальный номер, сформированный из наименования и кода операции, согласно ГОСТ 3.1201. На рис. 47 приведен классификатор операций в PDM-системе SMARTEAM.

Затем устанавливаются необходимые логические связи операции со следующими объектами:

- *А наименование операции;
- ***Б** оборудование из группа классов «Оборудование» (ОБ);
- *И инструкция из группы классов «Нормативнотехнологическая документация» (НТД);
- *ИОТ инструкция по охране труда из группы классов НТД;
- "ТТ" техническое требование в начале ТП операции;
- О содержание формообразующей операции (перехода);
- О_в содержание вспомогательной операции (перехода);
- Оу вспомогательной операции (перехода) установки-снятия;
- К комплектация оригинальных изделий из группы классов «Проекты» (ПР)

- Кпи комплектация покупных изделий из группы классов «Покупные изделия» (ПИ);
- М материал из группы классов «Материалы» (М);
- Р режим обработки из группы классов «Режимы обработки» (PO);
- ТТ техническое требование в конце ТП операции;
- ЦУ примечание (указание) в конце операции;
- **Т** инструменты (оснастка) покупные из группы классов «Инструмент, оснастка» (ИО).

♥ Классификатор операций: Сумма технологий	
A Concel	
🖃 🔂 Сумма технологий	Attribute Link Comment Revision Viewer
🕀 🗋 🧰 А_Сборка_(Assembles)	
🔄 🗋 🛅 Б_Формообразование_из_полимерных_материалов,_керамики,_стекла_и_резины	
😐 🗋 🧰 В_Получение_лакокрасочных_покрытий	
🔄 🗋 🧰 Г_Получение_покрытий_металлических_и_неметаллических_неорганичных	
🗄 🗋 🧰 Д_Обработка_давлением	
🔄 🗋 🧰 Е_Порошковая_металлургия	
🕂 🗋 🧰 И_Испытания	
🗄 🗋 🧰 К_Контроль	
🗄 🗋 🧰 Л_Литье_металлов	
🗄 🗋 🧰 М_Монтаж,_электромонтаж	
🕂 🗋 🧰 Н_Наплавка	
🔄 🗋 🧰 0_0бщего_назначения	
🕂 🗋 🔲 П_Пайка	
🗄 🗋 💼 Р_Обработка_резанием	
🕂 🗋 🧰 С_Сварка	
🗄 🗋 🧰 Т_Термообработка	
🕂 🗋 🧰 У_Упаковка	
🕂 🗋 🧰 Ф_Фотохимико-физическая_обработка	
🗄 🗋 🧰 Х_Хонингование,_отделочные_операции	
🗄 🗋 🛄 Ш_Шлифование	
🗄 🗋 🧰 Щ_Перемещение	
🗄 🗋 🧰 Э_Электрофизическая_обработка	
1	

Рис. 47 Классификатор операций

Для создания переходов в SMARTEAM необходимо указать его содержание и тип. На рис. 48 приведен классификатор переходов.

T.	🛫 SQ_TPFindO: Строка описания TП									
	OK Cancel									
D	ata Chart									
Ĺ	Код в ТП (Тип строки описани	я) Обозначение	Наименование 🗸 🔺							
1	0	0->Расточить_фаску,_выдерживая_ра								
2	0	0->Зенкеровать_деталь,_выдерживая								
3	Оу	Оу->Установить_оправу_в_центра								
4	0	0->Точить_поверхность,_выдерживая								
5	0	0->Точить_резьбу,_выдерживая_разм								
6	0	0->Отрезать_деталь_выдерживая_раз	8							
7	Ов	Ов->Кантовать_на_рабочем_месте								
8	Оу	Оу->Переустановить_изделие								
9	0	0->Сверлить_отверстие,_выдерживая								
10	0	0->Проверить соответствие_марки_м	6							
11	Оу	Оу->Установить_и_закрепить_изделие								
12	Оу	Оу->Установить_и_закрепить_поводон	c							
13	0	0->Точить_канавку,_выдерживая_раз	•							
14	Ов	Ов->Уложить_детали_в_тару								
15	0	0->Зачистить_наплывы_и_выплески_м	х							
16	0	0->Точить_канавку_для_выхода_резы	6							
17	0	0->Сверлить, выдерживая размеры								
18	0	0->Притереть_до_неплоскостности								
19	0	0->Расточить_отверстие,_выдерживая								
20	0	0->Точить_поверхность								
21	0	0->Фрезеровать_отверстия								
22	0	0->Зацентровать деталь								
	-									

Рис. 48. Классификатор переходов

Возможные типы переходов:

- у установка-снятие;
- в вспомогательный;
- рг рабочий грубый (черновой);
- рпч рабочий получистовой;
- рч рабочий чистовой.

Примечание. 1. Объекты – строки описания, «привязанные» к «дереву» ТП иерархическими связями, отображаются **черным цветом**.

2. Объекты – строки описания логически «привязанные» к «дереву» ТП, отображаются зеленым цветом.

3. Символ (* – «звездочка») перед кодом строки **Б**, **И**, **ИОТ** предназначен для поддержки последовательности строк описания ТП в «дереве» ТП в соответствии с требованиями стандартов ЕСТД при сортировке «дерева» командами: «Сортировать \rightarrow По обозначениям \rightarrow По возрастанию».

Наполнение базы данных инструментов и технологического оборудования

Внешний вид окна, содержащего перечень инструмента и оснастки, представлен на рис. 49. В случае отсутствия в системе SMARTEAM необходимого технологического оборудования, его требуется создать самостоятельно. Для наполнения базы данных инструментов и оборудования следует зайти в классы с соответствующими названиями **Оборудо**-

вание и Инструменты и оснастка . Затем в дереве проектов создать необходимое оборудование (инструмент/оснастку). Это позволит значительно сократить время на разработку, так как заведенные объекты могут в дальнейшем использоваться для проектирования технологического процесса.

Data	Chart												
Cla	ss № ГОСТа/Код в ТП/Место хран.	Наимен.класса / Обозн.ИО 🛛 🗠	Код исполн. /Вид и 🔺	Attribute Link	Comment V	/iewer							
433 🔄	P01A6	Резцы_токарные_отрезные,_с_механиче											
434 🔄	P01A4	Резцы_токарные_отрезные,_с_пластина		Honor									
435 🔁	P01A4K3	Резцы_токарные_отрезные,_с_пластина	1	ттеречень инструмента, оснастки:									
436 🔄	P01A4X4	Резцы_токарные_отрезные,_с_пластина		- пап	ка пла і	информации и лок, учета ИО на склал	av						
437 🔄	P016	Резцы_токарные_подрезные		- напка для информации и док. учета ио на складах.									
438 📥	P0167	Резцы_токарные_подрезные,_с_механи		 папка для разделов классификатора ГОСТов и литературы 									
439 🔁	P0167C	Резцы_токарные_подрезные,_с_механи											
440 🔁	P0166	Резцы_токарные_подрезные,_с_механи	u i	наописи свеј	жу иноексни	ых полеи - это овозначение и∪ оля учета на складах, сниз	у - оля классифик						
441 🔁	P0156C	Резцы_токарные_подрезные,_с_механи	4				код исполнения						
442 🔁	P0164	Резцы_токарные_подрезные,_с_пластин	4	Код места		Обозначение ИО с № ГОСТа + № экземпляра (+01+99) и/или						
443 🔁	P018	Резцы_токарные_проходные		хранения ИО		Пример: Штангенциркуль ШЦ-II-250-0,05 ГОСТ 166-89 - экз.1	<u>код дубля (+Д1 +</u>						
444 🔄	P01B1	Резцы_токарные_проходные,_из_быстр		P0167C		Резцы_токарные_подрезные,_с_механич.креплением_п							
445 📥	P01B1×1	Резцы_токарные_проходные,_из_быстро		Для классифи	kamopa:								
446 🔄	P01B1×5	Резцы_токарные_проходные,_из_быстро		Код класса, п	одкласса,								
447 🔁	P0187	Резцы_токарные_проходные,_с_механи	4	группы и т.д. ————— Наименование класса, группы и т.д.									
448 🔄	P01B7C	Резцы_токарные_проходные,_с_механи	4	Для СТП огра	ничения:								
449 🔁	P0186	Резцы токарные проходные, с механи	4	№ ГОСТа —		~> СТП ограничения							

Рис. 49. Перечень инструмента, оснастки в PDM-системе SMARTEAM

Проектирование технологической документации

Формирование маршрутной карты осуществляется активизацией соответствующей кнопки (МК), расположенной на титульном листе СквТП, ТП по методам обработки и титульном листе изделия. В приложении 1 приведен пример маршрутной карты, полученной в SMARTEAM.

Текстовая технологическая документация по требуемой форме и карты эскизов формируются автоматически. По запросу технолога на вывод требуемого технологического документа в базе данных автоматически создается паспорт этого документа, заполняются все поля паспорта и формируется файл информации в виде этого документа. Полученный таким образом документ может быть проверен, утвержден и сдан в электронный архив. Для вывода карт эскизов в базу данных должны быть введены файлы эскизов, выполненные в одной из САD-систем. При проектировании эскизов могут быть заимствованы модели или чертежи изделий, разработанные конструктором. Вывод на печать возможен по одному документу или комплектами. Система поддерживает стандарты ЕСТД, однако в системе имеются все средства для разработки программ вывода документации по любой форме силами пользователей.

Сформированные таким образом документы могут быть проверены, утверждены и сданы в электронный архив.

Создание операционных карт. Технологическую документацию вводят в состав объектов «№ операции». На титульном листе объектов «№ операции» необходимо выбрать «Ввести объект «Документ техно-логический», при активизации данной опции на экран выводится список, из которого можно выбрать: ОК; Оки; ОКк; КЭ; КН; КС; КК; ТО. Для формирования операционной карты в предложенном списке необходимо указать ОК.

Для формирования ОК с эскизом необходимо выполнить следующие операции:

- 1. Выбрать опцию «Ввести объект эскиз» в состав требуемого документа.
- 2. Создать эскиз (изображение изделия) в САD-системе или на кульмане с последующим сканированием.
- 3. Привязать файл эскиза в формате *.gif к объекту «Эскиз технологический»;
- 4. Выбрать опцию «Сформировать ТК», после чего файл эскиза будет автоматически объединен с файлом бланка ТК и общий файл привязан к объекту ТК.

При проектировании эскиза в CAD-системе файл эскиза можно сохранить в SMARTEAM и в формате CAD-системы для последующего редактирования, и в формате *.gif для создания ТК с эскизом.

- Путь к файлу в формате CAD-системы вводят стр. «Автор+Файл» паспорта объекта «Эскиз технологический» и просматривают на стр. «Просмотр».
- Путь к файлу в формате *.gif вводят на ТЛ паспорта и просматривают на стр. «Эскиз в формате *.gif».

В SMARTEAM существует возможность формировать документы **без эскиза** для этого необходимо выбрать кнопку **«Сформировать ТК»**. При этом система задает вопрос: «Вы хотите сформировать ТК без эскиза?». При ответе «Да» выбранная КТ будет автоматически сформирована на соответствующем бланке.

В приложении 2 приведен пример операционной карты, полученной в SMARTEAM.

Отображение дерева ТП со всеми логическими связями

На последнем этапе выполнения работы рекомендуется отобразить в дереве ТП логические связи со всеми созданными объектами. Для этого из контекстного меню ТП следует выбрать команду Привязанные объекты>Логические связи ...



Рис. 49 Отображение логических связей проектов с ТП

Приложение 1

																	гост з	1118-82 4	орма 1 С	АПР			
Тубл									_						F				+	<u> </u>			
зам.																			+				
Тодп.																			-				
																кт	д 1-Р	01715	2	1			
Разрабо	азработал Велюров 22.10.2007 000.000.							.000.00	01		MK1-P 01715												
Тровери	л						_					BTVD	va										
/тверд	ил											Diy	κα										
M01		Kod		ED	MO	EU	Hazey	VIANA	Koda	2200				Dnoch		24004				M			
M02		700		LD	мд	ЕП	п.расл.	NPIIVI	1003	acum.				προφ	аль а ре	змеры			<u> </u>	1015			
-	Llev	Vu	PM (Drien	-	(од нэши		onenauuu	,,					0	ก็กวมวนค	ייטע פוווי	VMAUM	,	<u> </u>				
5	1400	71	1111	Код.	наименова	чие обор	удования	операци	4	СМ	Пре	od. P	YT.	KP	коил	EH		7 Кит.	Тп.з.	Tum			
ЩУ	C6r	n xna	нение и	и тран	нопортира	вку пом	а и отхол	ов черн	ных и шве	тных	мета	аппов и	сплаво	впрои	изволи	гь по и	нструк	пии 632	25001.1				
	1000	<u>, vba</u>		1 1001		Dity sion			пыхиць						Броди		ine i pyr						
Δ	01	02		05	01П Зач	истка				иот	->1-7	2/80											
<u>Б</u>	Vct	3H08K3	9 0 0 3	ачист	ки Т6358.	547																	
	1201	anobid	а для эт	анист		0-11				_			_	-									
A	01	02	0	10	P1A4 T0KA	РНО-ВИНТ	ОРЕЗНАЯ о	трезка (п	рорезка)	иот	->41	02-104	2001	_									
Б	Авт	омат п	родоль	ного	точения о	: ЧПУ АГ	TT-901																
	1.01		POMON							_			_	-									
A	01	02	0	15	01H2 Пр	омывк	а раство	рителя	ми	иот	->по	ттп											
Б	Пин	ия ста	пионар	тных	ванн		•	•		_													
										_			_	-									
Α	01	02	0	20	P1A1 TOK	АРНО-В	инторез	НАЯ. пр	одольная	иот	->62	02-130	2000:0	20-> C	квТП (01715	xx						
Б	Ток	арно-ч	асовой	i T-65	5					_													
										_			_	-									
Α	01	02	0	25	01H2 Пр	омывк	а раство	рителя	ми	иот	->по	ттп		_									
Б	Пин	ия ста	пионар	тных	ванн		•	•		-	1												
										_			_	-									
Α	01	02	0	30	К1 Контр	оль				иот	->1-5	4/85			-				L				
Б	Pa6	бочее м	лесто к	атно	олера					_													
	1									-	-		-	-									
										_			_					_		-			
													_	-				_					
			T T							<u> </u>			1		<u> </u>				1				
						1	I	I									- 1						
МК	(Мари	upyn	пная	я карта	a																	
																	ГС	OCT 3.111	8-82 Øop	ма 16 СА			

Приложение 2

1																	гос	T 3.14	04 - 86 Фор	ма 3 САПР	
1														E							
Дубл.																					
Взам.																					
Подп.																					
																			1	1	
Разраб.										0	00 000	0.01						011.01	1715		
-																				1/13	
Н.конт	роль										Βту	/лка						01	02	010	
	Ha	именова	ание опе	ерации			Материал			Твердос	ть	EB	ΜД		Прос	филь	и размерь	1	M3	коид	
P1A4 TO	КАРНС	-винто	РЕЗНАЯ	1 отрезі	ка (прорезі	ка															
(Обору	дование,	устрой	іство	чпу	06	означение прог	раммы		То	1	6	Тп.з.	Тш	m.			CO	ж		
Автом	ат про,	дольного	о точени:	ясч⊓у	∕ A∏T-901																
Р							Pp - pe	кимы рез	зания (ПИ, D ш	иB, L	, t, i,	S, n, v)								
MT					сож													_			
	"TT" 0	собо отв	етственн	ная опе	ерация!																
01	Donn	DODTH TOP		nyuno	anoouonu																
П>тси	1 182	53416 10k	лец выде	эржива	и разшеры																
	Pesel	,2130-40	001T15 K	8																	
Π	ΠПр	оверить	на соотв	зетстви	е шерохова	атости пове	рхностей														
L																					
<u> </u>												-						_			
<u> </u>																					
<u> </u>																		-			
L																		_			
	<u> </u>																				
									1												
OK	ОК Операционная карта												22.10.2007 11:29								

Список литературы

- 1. Алиев Р., Давыдов К. Использование систем симуляции для повышения точности и производительности механической обработки // журнал «САПР и графика» №10 2001
- 2. Волков И. Vericut производственный комплекс на рабочем столе // журнал «САПР и графика» №4 2003
- 3. Зильбербург Л.И., Молочник В.И., Яблочников Е.И. Реинжиниринг и автоматизация технологической подготовки производства в машиностроении. – СПб: Политехника, 2004. – 152 с.
- 4. Молочник В. И. IMSpost универсальный генератор постпроцессоров для оборудования с ЧПУ // журнал «САПР и графика» №8 2002
- 5. Сопротивление материалов. Под ред. А.Ф.Смирнова. Учебник для вузов. Изд. 3, М., Высшая школа, 1975.
- 6. Шатров Б.В. Теоретические основы анализа конструкций с применением метода конечных элементов – М: МАИ, 1998.
- 7. Яблочников Е.И, Маслов Ю.В. Автоматизация ТПП в приборостроении / Учебное пособие. – СПб: СПбГИТМО (ТУ), 2003. – 104 с.
- 8. Яблочников Е.И, Фомина Ю.Н., Тремба В.Ю. Использование РLМ-технологий в проектировании и подготовке промышленного производства. /Региональная информатика-2006 «РИ-2006».
- 9. Язык программирования АРТ // журнал «Вычислительная техника и ее применение» №11 1991.
- 10. Сайт компании "Би Питрон" www.bee-pitron.com.
- 11. Справочная документация по системе IMSpost
- 12. Справочная документация по системе VeriCut
- 13. MSC.Patran 2001. User's Guide for Windows.
- 14. MSC.Patran 2000 (r 2), Release Guide.





В 2007 году СПбГУ ИТМО стал победителем конкурса инновационных образовательных программ вузов России на 2007–2008 годы. Реализация инновационной образовательной программы «Инновационная система подготовки специалистов нового поколения в области информационных и оптических технологий» позволит выйти на качественно новый уровень подготовки выпускников и удовлетворить возрастающий спрос на специалистов в информационной, оптической и других высокотехнологичных отраслях экономики.

КАФЕДРА ТЕХНОЛОГИИ ПРИБОРОСТРОЕНИЯ

Кафедра технологии приборостроения относится к числу ведущих кафедр института со дня его основания в 1931 году. Тогда она называлась кафедрой механической технологии и возглавлялась известным ученым в области разработки инструмента профессором А.П. Знаменским. Позже она была переименована в кафедру технологии приборостроения.

За время своего существования кафедра выпустила из стен института более тысячи квалифицированных инженеров, более сотни кандидатов и докторов наук. В разные годы ее возглавляли известные ученые и педагоги профессора Николай Павлович Соболев и Сергей Петрович Митрофанов.

Кафедра имеет выдающиеся научные достижения. Заслуженным деятелем науки и техники РСФСР, профессором С.П. Митрофановым были разработаны научные основы группового производства, за что он Ленинской удостоен премии CCCP. Методы был группового производства с успехом применяются в промышленности и постоянно развиваются его учениками. Заслуженным изобретателем Российской Федерации Юрием Григорьевичем Шнейдером разрабатаны метод и инструментарий нанесения регулярного микрорельефа на функциональной поверхности.

В настоящее время кафедра осуществляет выпуск специалистов по специальностям "Технология приборостроения" (инженер-технолог, инженер-технолог-менеджер, инженер-технолог искусственному ПО интеллекту в приборостроении) и "Системы автоматизированного проектирования" (инженер-системотехник). Ha кафедре ведется подготовка бакалавров, магистров, инженеров И аспирантов ПО специализациям силами семи профессоров и девяти названным доцентов.

Владимир Сергеевич Гусельников Анна Леонидовна Комисаренко Михаил Михайлович Шальнов Под ред. к.т.н., доцента Евгения Ивановича Яблочникова

Моделирование приборов, систем и производственных процессов

Приложение II Методические рекомендации по выполнению СРС

В авторской редакции		
Зав. редакционно-издательским отд	Н.Ф. Гусарова	
Лицензия ИД № 00408 от 05.11.99		
Подписано к печати 08.10.2008		
Отпечатано на ризографе	Тираж 100	Заказ № 1255

Редакционно-издательский отдел

Санкт-Петербургского государственного университета информационных технологий, механики и оптики 197101, Санкт-Петербург, Кронверкский пр., 49

