

## АРХИТЕКТУРА И РЕАЛИЗАЦИЯ СИСТЕМЫ ТВЕРДОТЕЛЬНОГО МОДЕЛИРОВАНИЯ ОБРАБОТКИ ИЗДЕЛИЙ В РЕАЛЬНОМ ВРЕМЕНИ

А.И. Обухов, С.В. Евстафиева, С.В. Рыбников (ФГБОУ ВО МГТУ "СТАНКИН")

Визуализация отработки управляющей программы позволяет избежать множества ошибок, связанных с позиционированием, переориентацией инструмента, проверить правильность подхода инструмента к заготовке и т.д. Особенно актуальным это становится при многокоординатной обработке [1]. При этом часто функции моделирования обработки выполняются сторонними программными продуктами, а не системами ЧПУ ввиду высокой стоимости разработки данного приложения для систем ЧПУ, а также ограничений производительности аппаратного обеспечения последних. Представлен новый подход к построению системы твердотельного моделирования обработки изделий в реальном времени, которая может быть реализована в составе системы ЧПУ<sup>1</sup>.

Ключевые слова: моделирование обработки изделий, воксельная модель, разреженное октодереве, цифровой двойник детали.

### Введение

Современная система ЧПУ не может считаться полноценным продуктом без средств моделирования обработки изделий в виртуальной среде. Возможность просмотра процесса и результата обработки в 3D формате до фактического выполнения управляющей программы на станке позволяет:

- убедиться в отсутствии явных ошибок при генерации или ручном написании программы, коллизий и случайных подрезов детали инструментом при переориентации и позиционировании;
- выбрать оптимальную геометрию инструмента для достижения требуемого качества поверхности (при необходимости использования режущего инструмента, отличного от заданного при генерации программы в САМ-системе);
- изменять стратегию резания или подходов инструмента к заготовке по результатам анализа последовательности движений инструмента.

Функции моделирования обработки часто возлагаются на отдельные программные продукты ввиду сложности и высокой стоимости их разработки, а также ограничений аппаратного обеспечения систем ЧПУ (относительно низкая производительность вычислительной платформы, отсутствие высокопроизводительной графической подсистемы). Тем не менее, наличие модуля виртуального моделирования непосредственно в составе программного обеспечения системы ЧПУ имеет преимущество высокой интеграции с ядром и всей инфраструктурой математического обеспечения. Это позволяет учесть при моделировании все особенности реализации геометрической задачи системы ЧПУ и просматривать процесс виртуальной обработки в режиме реального времени (как при эмуляции, так и в процессе реальной обработки детали на станке).

Рассмотрим новый подход к разработке модуля виртуального моделирования обработки.

### Представление модели изделия. Метод визуализации

Существует несколько применяемых на практике способов представления модели трехмерного объекта как структуры данных. Из компромиссных вариантов, основанных на использовании неполной информации о топологии твердого тела, можно указать карты высот, каркасные и поверхностные модели. Эти варианты имеют ограниченное применение, так как пригодны для описания объекта обработки лишь при некоторых простых сценариях. В общем случае необходимо использовать полноценное описание объемного тела, основанное на структуре конструктивной твердотельной геометрии, граничном представлении или воксельной модели (экзотические варианты не рассматриваем) [2]. Конструктивная твердотельная геометрия и граничное представление широко используются для описания моделей в CAD-системах, но плохо подходят для моделирования обработки, особенно в масштабах реального времени. Потенциально неограниченная последовательность перемещений инструмента, обрабатываемых с частотой в десятки Гц, приводит к лавинообразному росту числа элементов модели и необходимости регулярного проведения процедуры упрощения поверхностей. Это требует немалых вычислительных ресурсов и сводит на нет преимущество точности представления методов, основанных на математическом описании формирующих объект поверхностей. По этой причине для программного представления деформируемых в реальном времени тел наиболее перспективной представляется воксельная<sup>2</sup> модель, которая «естественным» образом подходит для интенсивных процедур удаления объема, характерных при моделировании резания.

Воксельная модель в чистом виде (как обобщение двумерного растрового представления на случай трех измерений, то есть трехмерный массив значений)

<sup>1</sup> Работа выполнена при поддержке Министерства образования и науки РФ в рамках выполнения Госзадания (№ 2.1237.2017/4.6) и проводилось с использованием оборудования, предоставленного центром коллективного пользования ФГБОУ ВО МГТУ "СТАНКИН".

<sup>2</sup> Воксел — элемент объемного изображения, содержащий значение элемента раstra в трехмерном пространстве. Воксели являются аналогами двумерных пикселей для трехмерного пространства.

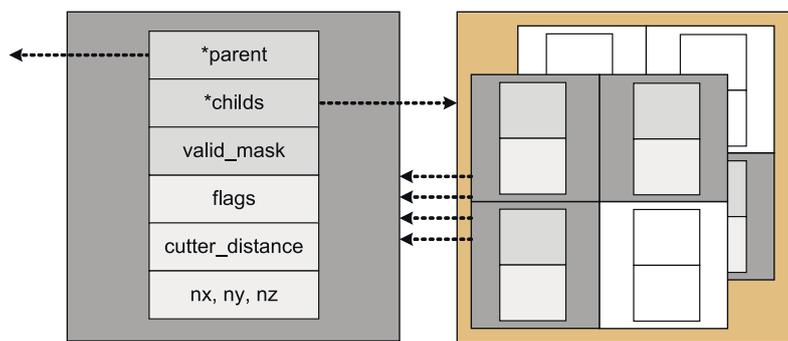


Рис. 1. Структура вокселя и связь между двумя уровнями октодерева

требует чрезвычайно больших объемов памяти и ограничивает разработчика в возможностях построения быстрых алгоритмов визуализации и изменения структуры объекта. Поэтому были предложены разнообразные структуры для оптимизированного представления массива вокселей. Основного внимания заслуживают иерархические структуры, в особенности разреженное октодерево<sup>3</sup> [3]. В ряде публикаций показана эффективность такого представления для минимизации занимаемой моделью памяти и ускорения процесса визуализации. Мы же покажем, каким образом воксельное октодерево позволяет реализовать эффективный алгоритм виртуальной обработки изделия в реальном времени.

Разреженное октодерево представляет собой иерархию уровней двоичного разбиения пространства. Корень дерева — куб, сторона которого равна наибольшему измерению объекта (из трех измерений по осям декартовых координат). Все узлы, кроме узлов последнего уровня дробления (то есть «листьев» дерева, представляющих наименьшие значимые элементы

Таблица 1. Оценка занимаемой памяти для моделей с разным разрешением

Разрешение модели	Размер, Кб
256 x 256 x 256	2760
512 x 512 x 512	10684
1024 x 1024 x 1024	43920
2048 x 2048 x 2048	174198

объема), имеют массив из восьми дочерних узлов. Дочерние узлы соответствуют кубическим объемам, полученным рассечением родительского куба пополам по каждой декартовой координате. Каждый узел должен содержать признак активности, то есть показывать, пустой ли это объем пространства или содержащий часть материала объекта.

В нашем варианте дерева структура вокселей и связей между уровнями выглядит следующим образом (рис. 1). Здесь parent — это указатель на родительский блок. У корневого узла он равен нулю. Узел имеет указатель на массив восьми дочерних узлов (childs). Если указатель childs равен нулю, то блок представляет собой сплошную область объекта или лист дерева, его дробление не требуется. Таким образом, память выделяется только тогда, когда часть родительского объема удаляется из сплошного объекта. Наличие одного указателя на массив всех дочерних узлов вместо восьми отдельных указателей позволяет существенно сократить число операций с динамической памятью при изменении объекта. Для определения активности дочерних узлов имеется маска valid\_mask. 8 бит ее значения соответствуют восьми узлам. Для сплошного блока маска равна 255 (то есть все узлы активны, ни один не удален). При редактировании объекта производятся операции с масками узлов, а освобождение выделенной под каждый указатель childs памяти происходит только тогда, когда маска соответствующего узла обнуляется (это говорит о том, что объем узла полностью пуст).

Остальные элементы структуры определяют свойства вокселя, не связанные с топологией дерева: flags — индикатор типа узла (сырой объем заготовки, поверхность заготовки с предварительно рассчитанной нормалью, обработанная поверхность); cutter\_distance — текущее расстояние от центра вокселя до поверхности инструмента, параметр определяет приблизительную величину врезания инструмента в объем вокселя; nx, ny, nz — компоненты вектора нормали для расчета освещения.

В табл. 1 приведено сравнение размеров файлов обработанных и сохраненных на диске моделей разного разрешения для одной и той же детали. При увеличении разрешения в 2 раза объем занимаемой моделью памяти увеличивается приблизительно в 4 раза. Такая закономерность прослеживается для любых деталей. Здесь проявляется ключевое преимущество разреженного октодерева перед простым воксельным представлением: объем памяти модели пропорционален площади обработанной поверхности, а не объему (это наше эмпирическое

Таблица 2. Среднее время отрисовки кадра для моделей разного разрешения

Разрешение модели	Время отрисовки кадра для процессора, мс	
	Core i3-540	Core i7-3770
256 x 256 x 256	11	5
512 x 512 x 512	13	6
1024 x 1024 x 1024	17	8
2048 x 2048 x 2048	22	10



Рис. 2. Два способа визуализации цифрового двойника детали

<sup>3</sup> Октодерево — тип древовидной структуры данных, в которой у каждого внутреннего узла ровно восемь «потомков». Восьмеричные деревья чаще всего используются для разделения трёхмерного пространства, рекурсивно разделяя его на восемь ячеек.

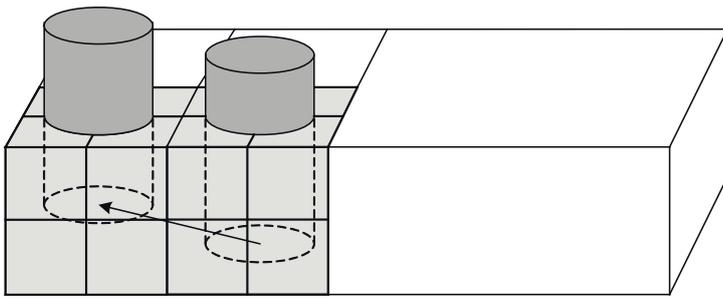


Рис. 3. Подмножество блоков, затронутых инструментом

наблюдение. Формальных доказательств этого тезиса в литературе пока не нет).

Известный алгоритм визуализации объекта на основе принципа бросания лучей (ray casting) [3] адаптирован для исполнения кода на центральном процессоре (в оригинальной работе был представлен код для CUDA). Решение использовать оптимизированный и отлаженный алгоритм с убедительными опубликованными результатами себя оправдало — визуализация объекта обработки в реальном времени оказалась возможной даже для терминальных компьютеров системы ЧПУ средней производительности (процессор Core i3–540, табл. 2).

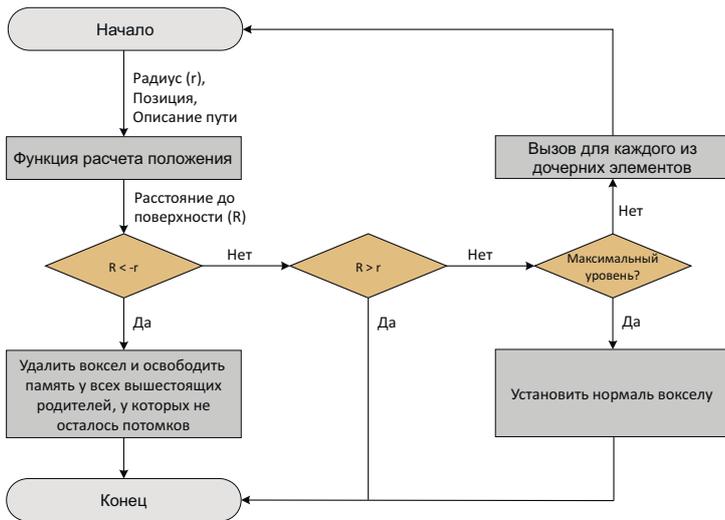


Рис. 4. Процедура виртуальной обработки (удаления материала модели)

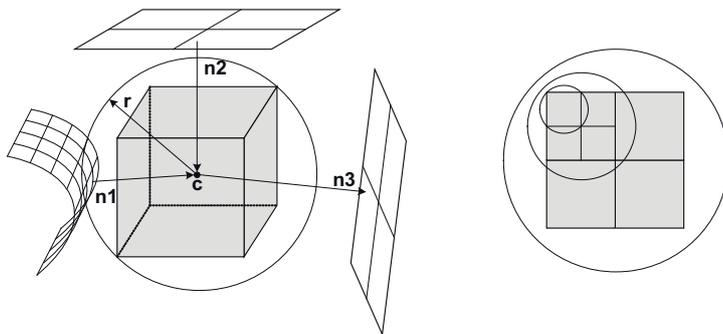


Рис. 5. Представление блока как сферы при расчетах положения

Качественный расчет освещения в трехмерной графике не мыслим без информации об ориентации освещаемых поверхностей. Для воксельных моделей чаще всего используются процедуры аппроксимации нормалей на основе градиента поля [4]. Такой метод в предложенной системе тоже реализован. Но качественно иного уровня реализма удалось достичь за счет реализации идеи присвоения вокселу значения нормали ближайшей точки поверхности инструмента (информация о ближайшей точке является косвенным результатом процедуры удаления объема при виртуальной обработке). На рис. 2 показано сравнение визуализации цифрового двойника детали с помощью аппроксимации нормали градиентом (справа) и метода копирования нормали поверхности инструмента (слева). Для расчета освещенности в обоих случаях используется классический метод Фонга [5]. Градиент рассчитан по 26 ближайшим соседям [6].

Метод копирования нормали ближайшей точки поверхности инструмента позволяет реалистично передать характер поверхности обработанного изделия, исключив характерную для градиентного метода дискретность уровней затенения и паразитную сглаженность острых граней детали. Фактически данный результат близок к отображению точного граничного представления реальной поверхности изделия, полученного на металлорежущем станке.

**Алгоритм виртуальной обработки**

Использование структуры разреженного октодерева позволяет не только уменьшить объем занимаемой памяти и эффективно визуализировать модель, но и реализовать быструю процедуру виртуальной обработки изделия. При простом растровом представлении воксельных данных вычислительная сложность задачи удаления объема при движении инструмента в общем случае пропорциональна кубу удаляемого объема (и разрешению). Двоичное разбиение пространства октодеревом позволяет разработать алгоритм, сложность которого пропорциональна поверхности, заметаемой инструментом. Это дает возможность выйти на качественно новый уровень виртуальной обработки, обеспечив возможность реализовывать процедуру удаления объема десятки и даже сотни раз в секунду при существующих ресурсах современных процессоров общего назначения (без использования массово-параллельных вычислений на графических картах). Такая частота достаточна для эмуляции реальной обработки в виртуальной среде с высокой точностью.

В предварительной фазе (рис. 3) процедуры удаления материала определяется множество блоков, которые затронуты движением инструмента.

Процедура виртуальной обработки рекурсивна и реализуется по алгоритму, представленному на рис. 4.

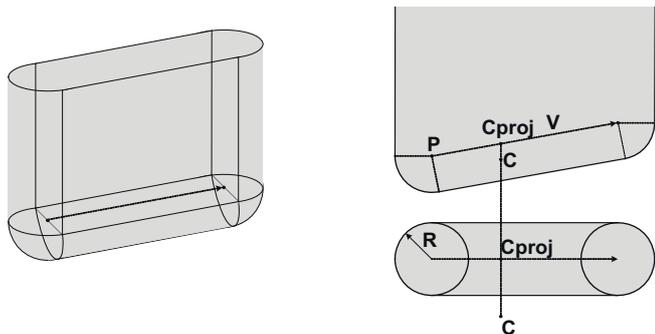


Рис. 6. Трехкоординатное движение инструмента со сферическим концом

Функция расчета положения реализуется отдельно для каждого типа инструмента и вида перемещения (с постоянной или переменной ориентацией инструмента).

Рассмотрим самый простой пример реализации функции расчета положения для случая линейного движения фрезы со сферическим концом без изменения ориентации (рис. 6).

На схеме  $V$  — вектор смещения инструмента;  $P$  — точка начального положения центра сферы инструмента;  $C$  — точка центра блока, для которого определяется расстояние (жирным шрифтом здесь и далее выделены векторные величины). Положение центра сферы конца инструмента в параметрическом виде можно представить выражением:

$$CP(t) = P + Vt, 0 \leq t \leq 1. \quad (1)$$

Отсюда найдем положение центра сферы  $C_{proj}$ , которое соответствует пересечению перпендикуляра, проведенного из точки  $C$  к вектору  $V$  в плоскости  $XY$ ,  $D$  — координаты центра блока относительно точки начального положения центра сферы инструмента. Для этого вычислим соответствующее значение параметра  $t_{proj}$ :

$$t_{proj} = \frac{D_x V_x + D_y V_y}{V_x^2 + V_y^2}, 0 \leq t_{proj} \leq 1, D = C - P. \quad (2)$$

Теперь найдем искомое расстояние  $d$  от точки  $C$  до поверхности, генерируемой движением инструмента. Имеем одну из двух конфигураций.

1. Если  $C_{proj}$  находится выше точки  $C$  или на одном с ней уровне ( $Z_c \leq Z_{cproj}$ ), расстояние  $d$  определяется относительно боковой цилиндрической поверхности:

$$d = \sqrt{d_{cx}^2 + d_{cy}^2} - R, d_c = C - C_{proj}, \quad (3)$$

где  $R$  — радиус скругления фрезы со сферическим концом.

2. В противном случае кратчайшему пути до поверхности будет соответствовать расстояние от точки  $C$  до ближайшей сферы из параметрического семейства  $CP$  (1). Параметр центра ближайшей сферы  $t_c$  получим из выражения (4):

$$t_c = \frac{D_x V_x + D_y V_y + D_z V_z}{V_x^2 + V_y^2 + V_z^2}, 0 \leq t_c \leq 1, D = C - P. \quad (4)$$

Отсюда:

$$d = \sqrt{d_{cx}^2 + d_{cy}^2 + d_{cz}^2} - R, d_c = C - CP(t_c). \quad (5)$$

Для инструментов с плоским и тороидальным концом расчеты значительно усложняются ввиду присутствия поверхностей с образующими 4-го порядка (кроме движений в плоскости  $XY$  и вдоль оси  $Z$  системы координат инструмента). Предложены аналитические методы расчетов для таких случаев [7, 8], однако мы применили подход на основе оптимизации нулевого порядка, позволяющий быстро вычислить приближенную величину расстояния от точки до криволинейной поверхности, избежав построения весьма громоздких и не всегда надежных алгоритмов точного решения уравнений высших порядков.

### Пользовательский интерфейс режима моделирования обработки

Экран режима моделирования обработки реализован по стандартной схеме интерфейса оператора ЧПУ с нижней и боковой панелями клавиш [9–11]. Предусмотрено два основных вида: полноэкранный с раскрытием панели визуализации на всю область рабочего экрана и стандартный с отображением текста программы и технологической информации (рис. 7).

$S$ -клавишам назначены операции по настройке ракурса отображения. Диалог параметров позволяет изменить цвет фона, материала заготовки и координатных осей. Настраивается период перерисовки вида при активном процессе обработки (на «слабых» машинах можно сделать период больше, снизив тем самым нагрузку на процессор). Устанавливается видимость осей координат и тип освещения (в простом режиме приме-

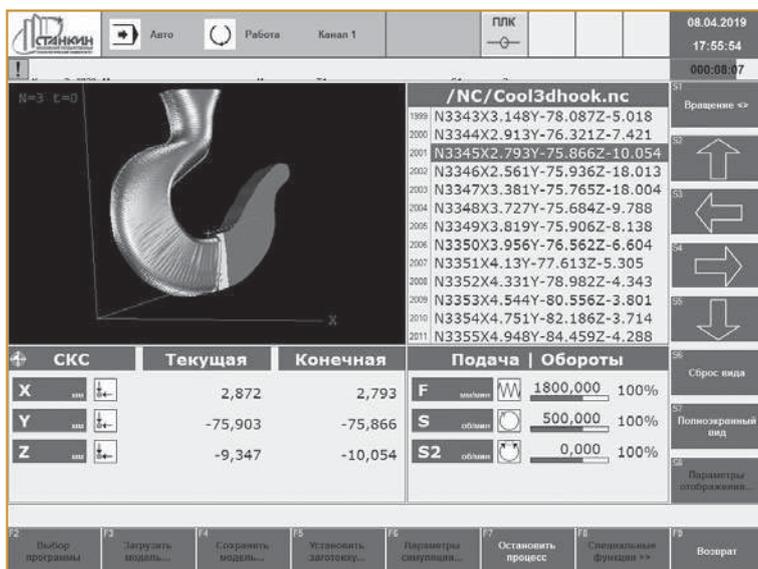


Рис. 7. Стандартный вид режима моделирования

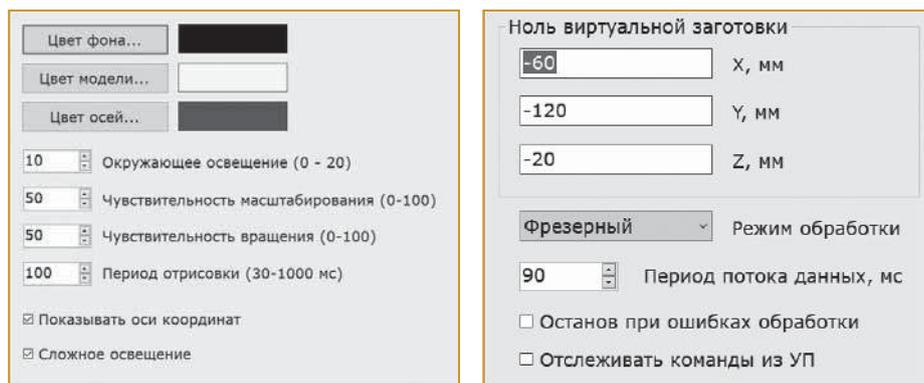


Рис. 8. Диалог параметров отображения (слева), обработки (справа)

няется один источник света и не реализуется вклад компоненты отражения, а в режиме сложного освещения устанавливается несколько источников и отражения активизируются).

В диалоге настройки параметров обработки задаются: смещение нуля виртуальной заготовки относительно текущей системы координат детали, вид обработки, период потока данных (определяет частоту обработки координат положения инструмента и вызова процедуры удаления объема), поведение при ошибках обработки и получении команд автозапуска.

#### Автозапуск моделирования обработки

Для запуска моделирования обработки из управляющей программы предусмотрена директива языка высокого уровня workpiece, которой передаются основные параметры заготовки и режима обработки. Директива должна быть указана в начале текста программы.

Набор параметров директивы зависит от формы заготовки (форма определяется первым параметром, например, 0 для параллелепипеда и 1 для цилиндра). Для любой заготовки задается разрешение, способ обработки (токарный, фрезерный), размеры и смещение.

#### Заключение

Предложенный способ представления изделия в виде разреженной воксельной модели позволил реализовать эффективный алгоритм виртуальной обработки изделия в реальном времени, при котором объем памяти, занимаемой моделью, и вычислительная сложность процедуры пропорциональны площади обработанной поверхности, а не объему.

Реализованный алгоритм виртуальной обработки дает возможность осуществлять процедуру удаления объема сотни раз в секунду при существующих ресурсах современных процессоров. Такая частота достаточна для эмуляции реального процесса резания

в виртуальной среде с высокой точностью.

Расчет освещения для воксельной модели реализован на основе идеи присвоения вокселу значения нормали ближайшей точки поверхности инструмента, что позволило получить результат, близкий к отображению точного граничного представления реальной поверхности изделия.

Для работы с функцией визуализации обработки реализованы два экрана оператора системы ЧПУ: полноэкранный вид с раскрытием панели визуализации на всю область рабочего экрана и стандартный вид с отображением текста программы и технологической информации.

#### Список литературы

1. *Мартинов Г.М., Обухов А.И., Пушков Р.Л., Евстафиева С.В.* Особенности реализации и специфика применения функций многокоординатной обработки в системе ЧПУ «АксиОМА Контроль» // Автоматизация в промышленности. №5. 2017. с.17-22.
2. *Никулин Е.А.* Компьютерная геометрия и алгоритмы машинной графики. Санкт-Петербург: БХВ-Петербург. 2003.
3. *Laine S., Karras T.* Efficient Sparse Voxel Octrees - Analysis, Extensions, and Implementation. NVIDIA Technical Report NVR-2010-001, Feb. 2010.
4. *Yagel R., Cohen D. and Kaufman A.* Normal Estimation In 3d Discrete Space. The Visual Computer, 8(5-6):278{291, June 1992.
5. *Phong B.T.* Illumination for Computer Generated Pictures. Communications of the ACM, 18(6):311{317, June 1975.
6. *Thürmer G., Wuthrich C.* Normal Computation for Discrete Surfaces in 3D Space. Eurographics. 1997. 16(3):C15-C26.
7. *Wang W.P., Wang K.K.* Geometric modeling for swept volume of moving solids[J] // IEEE Computer Graphics and Applications. 1986. 6(12): 8-17.
8. *Chung Y.C., Park J.W., Shin H., Choi B.K.* Modeling the Surface Swept by a Generalized Cutter for NC Verification. Computer-Aided Design (30) 8587-594. 1998.
9. *Martinov G.M., Kozak N.V., Nezhmetdinov R.A., Grigoriev A.S., Obukhov A.I., Martinova L.I.* Method of decomposition and synthesis of the custom CNC systems // Automation and Remote Control. 2017. Vol. 78. Issue 3. pp. 525-536.
10. *Georgi M. Martinov, Aleksandr I. Obuhov, Lilija I. Martinova, Anton S. Grigoriev* An Approach to Building a Specialized CNC System for Laser Engraving Machining // Procedia CIRP. 2016. Vol. 41. pp. 998-1003.
11. *Martinova L.I., Kozak N.V., Nezhmetdinov R.A., Pushkov R.L., Obukhov A.I.* The Russian multi-functional CNC system AxiOMA control: Practical aspects of application // Automation and Remote Control. 2015. Vol. 76, Issue 1, pp. 179-186.

*Обухов Александр Игоревич — канд. техн. наук, доцент, Евстафиева Светлана Владимировна — старший преподаватель кафедры компьютерных систем управления, Рыбников Сергей Валентинович — инженер ФГБОУ ВО МГТУ "СТАНКИН". Контактный телефон +7 (499) 972-94-40. E-mail: obukhov@ncsystems.ru*

Поступила в редакцию 10.04.2019.

Принята к публикации 26.04.2019