

Исследование и разработка унифицированного средства генерации управляющих программ¹

Л.И. Мартинова,
к.т.н., доцент, liliya-martinova@yandex.ru,
Н.Н. Фокин,
асп., nikolfokin@gmail.com,
МГТУ «СТАНКИН», г. Москва

В работе проанализированы методы отладки управляющих программ и используемые САМ-системы, постпроцессоры, циклы и функции. Предложен подход к созданию унифицированного средства программирования для токарно-фрезерных станков с ЧПУ для формализации существующих методов программирования.

This report describes and analyzes methods of debugging control programs and using the CAM system, postprocessors, cycles and functions. The approach to creating a unified programming tool for CNC turning and milling machines for the formatting of complex programming methods is proposed.

На данный момент в России развитие производственных комплексов идет по двум путям. Первый путь – создание предприятия «с нуля», когда разрабатывается структура технологического комплекса и осуществляется подбор автоматизированного оборудования. Как правило, предпочтение отдается оборудованию от одних и тех же производителей и с системами ЧПУ от единого производителя и с одинаковыми системами ЧПУ [1]. Этот путь дает не только единовременные выгоды при закупке технологического оснащения, но и в последующей его эксплуатации, в частности, за счет сокращения расходов на инструменты, разработку постпроцессоров, сокращение времени и расходов.

Другой путь – это направление для заводов с уже сложившимся парком, как правило, качественного оборудования, но от разных производителей, с системами ЧПУ не только от разных производителей, но и разных моделей. Эти предприятия сталкиваются с необходимостью покупать новые постпроцессоры, 3D-модели новых станков и даже новые САМ-системы для программирования на вновь прибывшем оборудовании. Зачастую на таких предприятиях детали одного типоразмера изготавливаются на однотипных станках с разными системами ЧПУ. При внесении конструктором изменений в чертеж детали требуется переписывать управляющие программы для каждого станка (или каждой системы ЧПУ), на которых деталь изготавливается, что требует значительных затрат времени [2]. Учитывая то, что, во-первых, предприятий, которые идут по второму пути значительное количество, и, во-вторых, предприятия из первой группы сталкиваются с необходимостью создания или приобретения постпроцессоров и 3D-моделей станков, поэтому в настоящее время актуальна задача создания унифицированного средства программирования (кроссплатформенного редактора управляющих программ).

В связи с вышеизложенным в настоящее время актуальным является вопрос создания универсального инструментария для создания управляющих программ на разных системах ЧПУ. Разрабатываемый кроссплатформенный редактор управляющих программ (NCEditor) – это унифицированное средство генерации кода управляющих программ для станков с ЧПУ, ориентированное на большинство известных операционных систем (Windows, MacOS, Android, Linux). Создание управляющей программы выполняется в диалоговом режиме, при этом учитывающее особенности написания управляющих программ, как для оборудования с системами ЧПУ, работающими только с циклами ISO-7bit, так и для систем ЧПУ, работающих с собственными циклами [3, 4]. Исходные данные (геометрические параметры заготовки, параметры фазового пространства станка, параметры используемых при обработке циклов) сохраняются в форме шаблона и могут вызываться и редактироваться с последующей генерацией управляющей программы для каждой системы ЧПУ, что значительно сократит время написания управляющей программы.

Рассмотрим более детально как реализуется разработка и отладка управляющих программ при наличии различного оборудования и систем ЧПУ. Так, для системы ЧПУ Siemens можно создать УП как в коде ISO, так и на языке высокого уровня ShopTurn, а для системы ЧПУ Fanuc соответственно - в коде ISO и в виде четырехзначных G-кодов ManualGuide. И для Siemens и для Fanuc во втором случае управляющие программы будут написаны не отдельными движениями, которые можно получить генерацией кода постпроцессором, а циклами, при этом управляющая программа для системы ЧПУ Siemens не будет работать на стойке с системой ЧПУ Fanuc. Более того, даже системы ЧПУ одного производителя, но разных версий могут не поддерживать управляющие программы друг друга. Так, например, программы, написанные в более поздней версии ShopTurn для Siemens 840 не будут читаться на более ранних версиях (Siemens 810 и 828), но программы, написанные через Sinutrain будут работать на всех стойках. Примеры форматов управляющих программ для разных систем ЧПУ представлены в таблице 1.

Таблица 1

Системы ЧПУ и формат управляющей программы

Axioma	Fanuc	Siemens	Mazak	Hendenhein	Haas	NC-200
встроенные циклы Аxiома	встроенные циклы ManualGuide	встроенные циклы Sinumeric	встроенные циклы Mazatrol	встроенные циклы Hendenhein	ISO-7Bit	встроенные циклы NC
	ISO-7Bit	встроенные функции ShopTurn	ISO-7Bit			встроенные функции NC САПР

¹ Работа выполнена при поддержке Минобрнауки России в рамках выполнения государственного задания (№ 2.1237.2017/ПЧ)

Написание программ в формате циклов и функций на языке высокого программирования значительно сокращает как размер УП (производители ЧПУ закладывают совсем небольшой объем памяти), так и возможность редактирования УП в цехе на стойке (что практически невозможно сделать с УП, сгенерированной постпроцессором).

Вторая сторона вопроса разработки и отладки управляющих программ касается цехового редактирования, которое является неотъемлемой частью процесса внедрения продукции в производство. При внедрении окончательно решается задача назначения корректных режимов резания, так как руководствуясь данными производителя режущего инструмента, не всегда можно сразу получить реально корректный вариант: в САМ-системе программист может написать УП, используя при моделировании наиболее подходящий по геометрическим характеристикам режущий инструмент, тогда как по факту такого может не оказаться. Здесь возникает потеря времени на выбор режущего инструмента из существующей базы САМ-системы или проектирование и внесение нового инструмента в эту же базу. А если САМ-систем несколько? Далее эта проблема перетекает в сферу постпроцессоров, которые написаны по одному для каждой САМ-системы. Это еще более увеличивает временные затраты на написания УП. Даже в самом идеальном варианте, когда только одна САМ-система с наличием необходимого инструментария в базе и один постпроцессор, время на написание увеличивается, так как необходимо задавать дополнительные поверхности для выхода инструмента из зоны резания, выбирать линии траектории, производить анализ на столкновение.

САМ-системы, где присутствует и инструментальная и станочная базы, превентивный контроль столкновений ведется более тщательно, чем в универсальном редакторе. В редакторе сам программист задает области ограничений и, если подходы, отходы и отскоки выбраны неверно, это обнаружится только при симуляции на стойке, где все используемые в программе инструменты корректно описаны и привязаны, а также корректно задана нулевая точка детали. В таблице 2 представлены характеристики современных методов написания управляющих программ для станков с ЧПУ.

Таблица 2

Основные характеристики современных методов написания управляющих программ

	САМ-система цехового уровня	САМ-система	NCEditor
формат ввода исходных данных	диалоговый режим с графическими подсказками	диалоговый режим с использованием 3D-модели детали	диалоговый режим с графическими подсказками
задание траектории	проектирование во встроенном в систему ЧПУ диалоговом графическом редакторе	графический режим с использованием 3D-модели детали	проектирование во встроенном в NCEditor диалоговом графическом редакторе
симуляция обработки	непосредственно на стойке	непосредственно в САМ-системе с использованием 3D-модели станка и инструментального магазина	на стойке или на эмуляторе системы ЧПУ на ПК
использование конфигурации оборудования	автоматическое или полуавтоматическое использование геометрических характеристик и M-кодов станка	автоматически при наличии 3D-модели станка	ручной ввод станочных параметров для каждой системы ЧПУ с сохранением в файл конфигурации
унификация управляющей программы	управляющая программа только для системы ЧПУ, на которой она сгенерирована	дерево проекта и управляющая программа при наличии 3D-модели станка и постпроцессора	дерево проекта и управляющие программы в виде циклов обработки для всех известных систем ЧПУ

Ещё одним важным моментом является то, что даже программирование циклов и функций (при работе в диалоговом режиме) может содержать разный набор переменных, различающихся для разных систем ЧПУ. Например, при программировании нарезания резьбы одни системы запрашивают количество проходов, а другие - глубину съёма; одни позволяют сокращать глубину каждого врезания по нисходящей (дегрессивный метод), другие не поддерживают данную функция. При нарезании торцевых канавок одни стойки учитывают минимальный и максимальный диаметр для врезания, который устанавливается производителем режущего инструмента, другие производители игнорируют эту особенность. То же самое с шириной пластины. Как правило, поддержка таких параметров осуществляется в стойках со встроенной САМ-системой цехового уровня, и это исключает возможность применения на данном станке управляющей программы, созданной для станка с другой системой ЧПУ. Проверить правильность УП, сгенерированных в редакторе, можно на эмуляторе ЧПУ для ПК или непосредственно на стойке в процессе статического моделирования до начала обработки

В редакторе управляющих программ используется следующий принцип работы:

1. Введение основных параметров обработки, значения которых являются неотъемлемой частью каждого технологического перехода:
 - параметры заготовки: тип (с отверстием, без отверстия, многогранник) и геометрические параметры (диаметр, длина, торцевой припуск), система координат и единицы измерения.
 - фазовое пространство станка: точка смены инструмента, тип и величина отскоков, ограничения оборотов.

2. Введение дополнительных параметров обработки – M-кодов конкретного станка с возможностью сохранения файла конфигурации и его дальнейшего использования при написании управляющих программ для изготовления других деталей на этом же станке.
3. Выбор типа обработки (токарная, сверлильная, фрезерная) или выбор дополнительных установок (смена типа и величины отскока, уменьшение или увеличение максимальных оборотов, изменение системы координат заготовки, изменение точки смены инструмента).
4. Введение параметров обработки для каждого выбранного типа, проектирование новых токарных и фрезерных контуров или выбор позиций для операций сверления.
5. Каждый выбранный тип обработки записывается в форму проекта в соответствии с выбранным номером перехода.
6. Полученные данные можно сохранить в файл проекта, который можно будет загрузить в систему редактора для дальнейшего использования, например, написания управляющей программы на другую стойку.
7. Система генерирует код управляющей программы для выбранной системы ЧПУ, исходя из введенных переходов на экране проекта.
8. Полученную в результате генерации управляющую программу можно сохранить в формате, читаемом системой ЧПУ.

рис. Общие параметры обработки

Выводы

На основе исследования и анализа методов написания управляющих программ для токарно-фрезерных станков с ЧПУ обоснована возможность и описан подход к созданию кроссплатформенного редактора управляющих программ для систем ЧПУ.

Литература

1. Г. М. Мартинов, Л. И. Мартинова Перспективы развития систем числового программного управления в концепции «Индустрия 4.0» //Иновации, №8, 2016. с.17-20.
2. С.В. Евстафиева, Н.Ю. Червоннова, О.А. Кудинов Особенности применения параметрического программирования при создании управляющих программ для системы ЧПУ «АксиОМА Контрол». Труды конф. CAD/CAM/PDM - 2016. Под ред. А.В. Толока. М.: ООО «Аналитик». –2016. с. 57-60.
3. Принцип построения универсального интерпретатора языка программирования высокого уровня для систем ЧПУ. Мартинов Г.М., Обухов А.И., Пушков Р.Л. Мехатроника, автоматизация, управление. 2010. № 6. С. 42-50.
4. The Russian multi-functional CNC SYSTEM AXIOMA Control: practical aspects of application. Martinova L.I., Kozak N.V., Nezhmetdinov R.A., Pushkov R.L., Obukhov A.I. Automation and Remote Control. 2015. T. 76. № 1. С. 179-186.