

05, май 2016

УДК 621.865.8

Генерирование траектории движения манипулятора мобильного робота при обработке объектов с параметрически задаваемой формой

Коротковс В., студент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Робототехнические системы и мехатроника»*

*Научный руководитель: Воротников С.А., к.т.н., доцент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Робототехнические системы и мехатроника»*

vorotn@bmstu.ru

1. Введение

На производствах различного плана всё чаще используются промышленные манипуляторы (ПМ). Они применяются для дуговой и точечной сварки, резки, покраски, обслуживания станков, механической обработки и т.д.

ПМ в большинстве случаев жёстко прикрепляются к полу (стене, потолку), либо устанавливаются на однокоординатный линейный позиционер, позволяющий перемещать его с точностью, соизмеримой с точностью самого ПМ.

Также, практически всегда ПМ работают по жёсткой, заранее написанной оператором программе. К примеру, при механической обработке, в случае изменения типа обрабатываемой заготовки или её положения относительно ПМ, необходимо переписывать программу движения.

Несмотря на привычную детерминированность среды работы ПМ, их использование возможно и для выполнения нестандартных задач. К примеру – манипулирование инструментом в среде с заранее неизвестными препятствиями.

Учитывая темпы развития промышленной робототехники, число нестандартных решений с использованием ПМ будет расти. Именно с задачей такого класса столкнулись авторы статьи.

2. Постановка задачи

Некоторое время назад в НУЦ «Робототехника» МГТУ им. Н.Э. Баумана поступил заказ на разработку комплекса, предназначенного для работы в заранее неопределённой среде, где траектория движения ПМ не задается априори, а должна генерироваться автоматически, в зависимости от места, где находится ПМ.

Комплекс должен включать в себя мобильный робот (МР) с колёсным шасси и установленным на нем 6-осевым ПМ. ПМ использует инструмент типа «нож» для обработки некоторых объектов. Оператор МР должен подводить ПМ к объектам обработки (ОО, в данном проекте – это растения), стационарно расположенным на участке. Заранее известны их координаты в глобальной системе координат (СК). После остановки МР вблизи ОО, определяется матрица пересчета координат основания ПМ из собственной СК в глобальную СК. После этого необходимо запустить обработку всех ОО, находящихся в зоне досяжимости ПМ.

В соответствии с заданием, оператор МР перемещает ПМ в заранее рассчитанные точки (рабочие места или РМ), количество и расположение которых рассчитываются так, чтобы время обработки участка было минимальным. Однако, в реальных условиях, МР не может быть точно спозиционирован в РМ, поэтому оператору необходимо переместить МР в некоторую допустимую область в окрестности РМ. Примерный вид участка с ОО изображён на рис. 1.

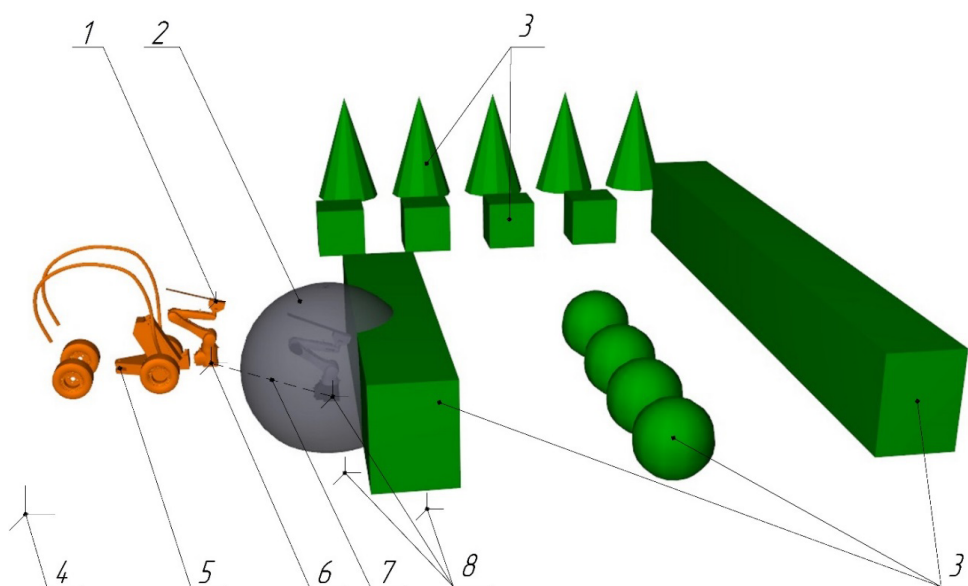


Рис. 1. Вид участка с ОО: 1 – инструмент, 2 – зона досяжимости, 3 – ОО, 4 – глобальная СК, 5 – МР, 6 – ПМ, 7 – траектория МР, 8 – РМ

Таким образом, оператор МР позиционирует ПМ в этой области, после чего запускается обработка ОО, во время которой ПМ должен обработать всё, что находится в зоне его достижимости.

На рис. 2 приведена структура МР. С пульта управления МР в систему логики приходят высокоуровневые команды («Выбрать участок», «Определить местоположение МР», «Начать обработку» и т.д.). Система логики обменивается информацией с базой данных и отправляет необходимую информацию в систему управления МР. Система управления МР в свою очередь управляет инструментом, ПМ и колёсным шасси. Обмен между большей частью блоков осуществляется через ROS [1].

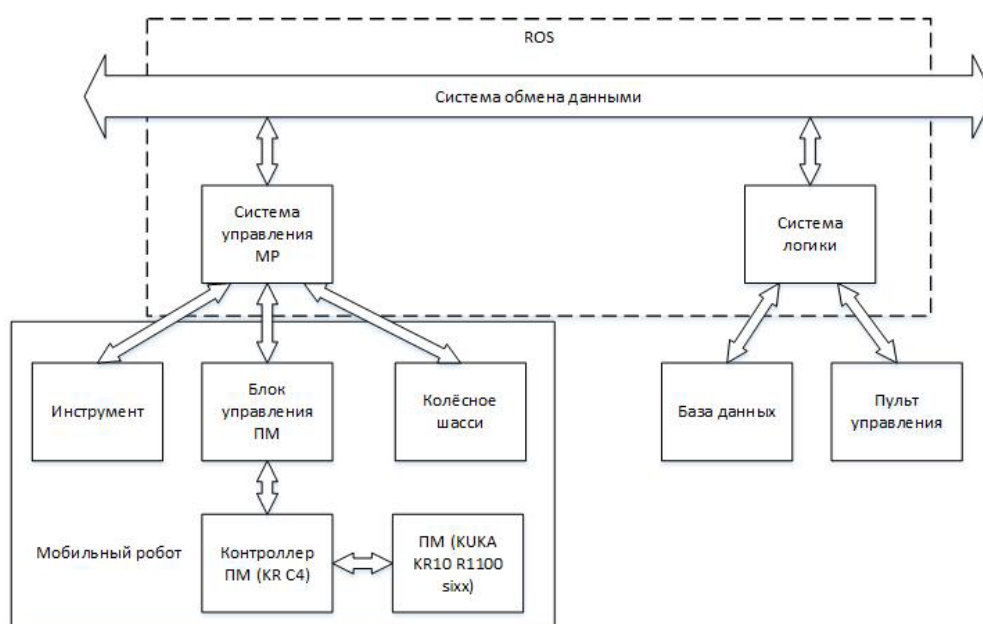


Рис. 2. Структура системы управления МР

Таким образом, после позиционирования ПМ в непосредственной близости от ОО, оператор, с помощью пульта управления отдаёт команду на обработку, которая поступает в систему управления ПМ, где генерируется траектория обработки.

3. Анализ САМ-систем

Существует несколько вариантов задания траектории движения ПМ. В простых случаях это делается вручную. Оператор ПМ последовательно задаёт точки траектории ПМ. Однако, при механической обработке деталей сложной формы ручное задание точек траектории потребует больших временных затрат и, при этом, не будет в должной мере соблюдена технология обработки.

В задачах механической обработки также используются САМ-системы, позволяющие генерировать программы движения для ПМ в режиме офлайн. Выделяют следующие САМ-системы: Robotmaster [2], SprutCAM Robot [3] и Delcam PowerMILL Robot [4].

Использование САМ-системы значительно облегчает работу технолога по сравнению с ручным методом задания траектории обработки, но не может полностью исключить его из процесса генерирования движения ПМ.

Однако, в ходе выбора САМ-системы авторами было обнаружено, что все они не предназначены для использования инструмента типа «нож». Кроме того, на формирование технологического процесса обработки даже несложных деталей уходит достаточно много времени. Так, по оценке авторов, для простых деталей типа параллелепипед (рис. 3 *а*), усечённая пирамида (рис. 3 *б*) и треугольная пирамида (рис. 3 *в*) технологу с небольшим опытом работы необходимо 15 минут. Для формирования технологического процесса обработки конуса (рис. 3 *г*), усечённого конуса (рис. 3 *д*) или сферы (рис. 3 *е*) уйдёт не менее получаса, для сложной фигуры (рис. 3 *ж*) - целый рабочий день.

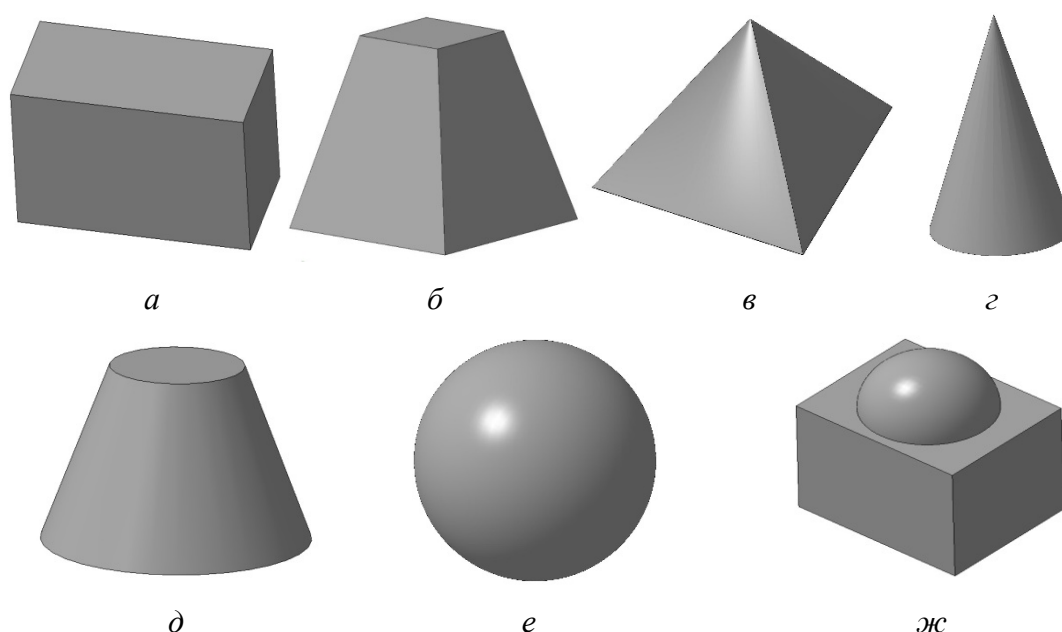


Рис. 3. Объекты обработки: *а* – параллелепипед, *б* – усечённая пирамида, *в* – треугольная пирамида, *г* – конус, *д* – усечённый конус, *е* – сфера, *ж* – сложная составная фигура

Исходя из анализа возможностей САМ-систем, можно заключить следующее:

1. САМ-системы позволяют написать программу обработки офлайн, однако это делается для определённого ПМ (рис. 1). В реальных условиях оператор МР не

сможет точно вывести ПМ в РМ, что приведёт к возникновению ошибки обработки равной ошибке вывода ПМ в РМ, а также к столкновению ПМ с препятствиями.

2. САМ-системы не работают без технолога, а привлечение технолога требует дополнительных финансовых и временных затрат. Написание скриптов для автоматизации работы САМ-системы очень трудоёмко и скорее всего не приведёт к нужному результату.

Поэтому, для генерирования траектории движения ПМ по обработке ОО предлагается разработать алгоритм, позволяющий в автоматическом режиме генерировать траекторию обработки произвольно задаваемых в пространстве ОО с проверкой возможности выхода в каждую точку и пересылкой данных в контроллер ПМ. При этом, оператор должен одной командой задавать начало обработки всех ОО, находящихся в зоне досягаемости ПМ.

4. Алгоритм генерирования траектории движения ПМ

При разработке алгоритма обработки ОО, из-за большого количества возможных форм ОО, было решено стандартизировать набор обрабатываемых поверхностей. При этом любой ОО разбивается на примитивы, принцип обработки которых известен.

В рамках алгоритма предлагается поверхность сложного объекта аппроксимировать следующими примитивами: 3-х (рис. 4 а) и 4-х угольными (рис. 4 б) плоскими гранями, а также поверхностями конической (рис. 4 в, г) и сферической (рис. 4 д) формы.

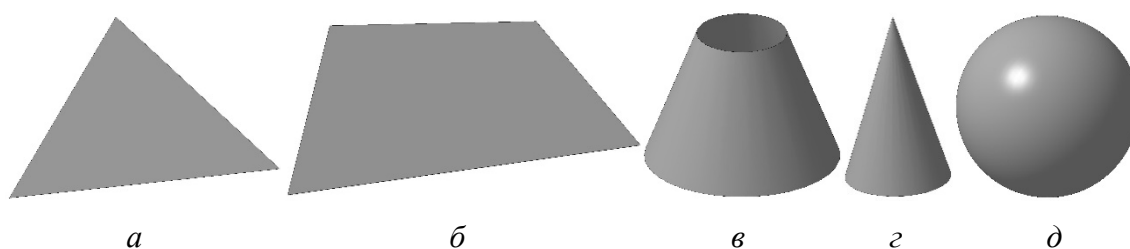


Рис. 4. Примитивы: а – 3-угольная грань, б – 4-угольная грань, в, г – коническая поверхность, д – сферическая поверхность

Таким образом, алгоритм обработки ОО можно представить в виде совокупности следующих этапов (рис. 5):

- а) подготовительные операции;
- б) разбиение ОО на примитивы;
- в) определение достижимых ОО;

- г) генерирование линий и точек обработки;
- д) определение обобщённых координат ПМ;
- е) выбор оптимальной траектории обработки;
- ж) построение траекторий перехода между траекториями обработки;
- з) отправка траектории в контроллер ПМ.

Рассмотрим реализацию основных этапов алгоритма подробнее.

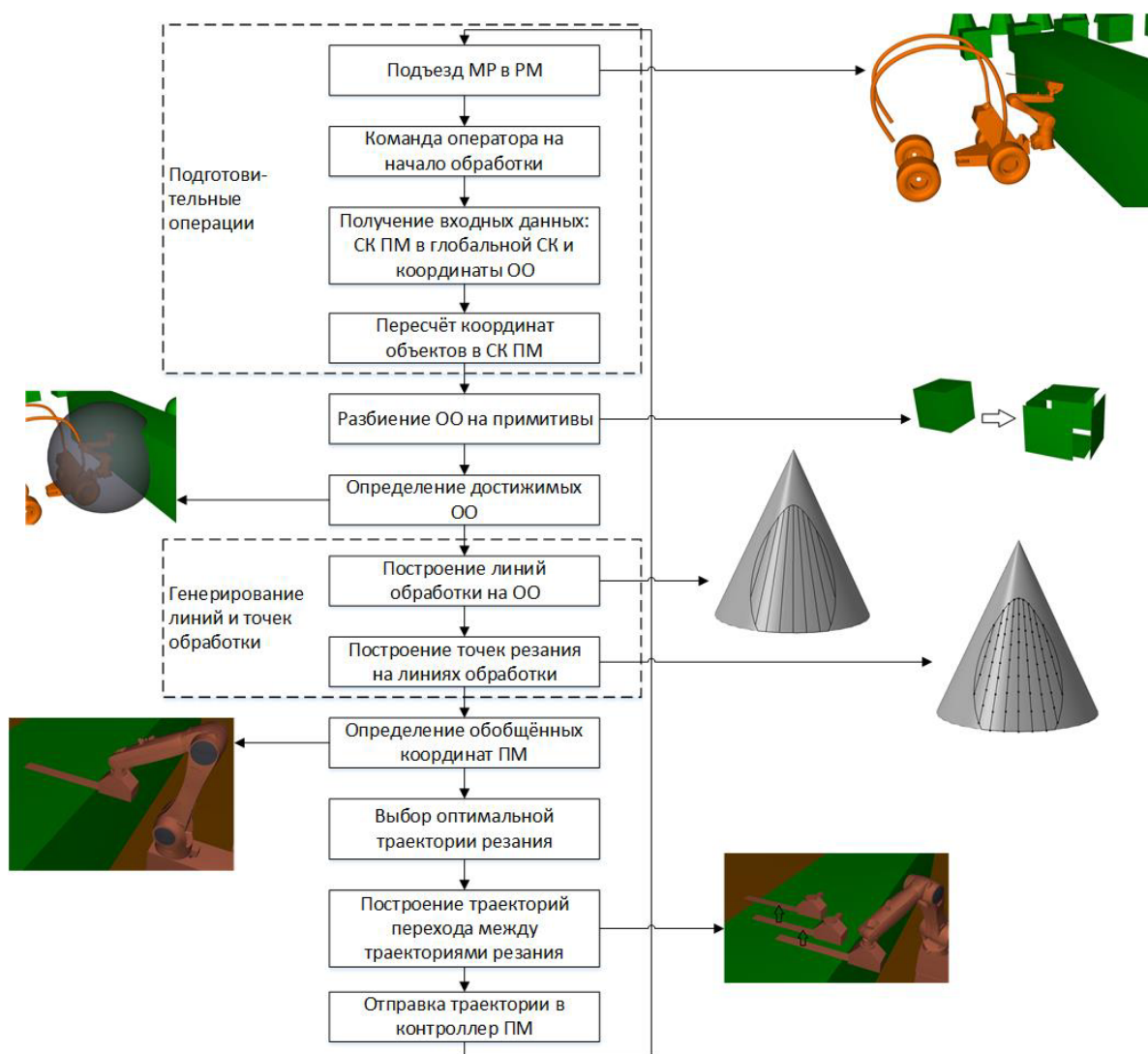


Рис. 5. Алгоритм генерирования траектории ПМ для одного РМ

а) Подготовительные операции

В рамках данного этапа выполняется группа операций, предваряющих непосредственно обработку ОО. Сначала МР перемещается в РМ, затем по команде оператора происходит определение положения ПМ в глобальной СК. Далее происходит пересчёт координат ОО в СК ПМ.

б) Разбиение ОО на примитивы

На данном этапе происходит разбиение составных ОО на примитивы (рис. 4). Например, усечённый конус (рис. 3 д) разбивается на коническую поверхность (рис. 4 в), а также нижнее и верхнее основания, для обработки задаваемые 4-угольными гранями (рис. 4 б). Шестигранник (рис. 3 а, б) разбивается на шесть 4-угольных граней.

в) Определение достижимых ОО

На этой стадии расчёта исключаются ОО, находящиеся вне зоны достижимости ПМ (вне области 2 рис. 1). Для этого на поверхностях ОО строится сетка с большим числом точек. Каждая точка проверяется на вхождение в зону достижимости ПМ. Если таких точек на ОО нет – ОО исключается из дальнейших операций с текущим РМ.

г) Генерирование линий и точек обработки

Цель данного пункта – сгенерировать на поверхностях примитивов линии, по которым пройдёт обрабатывающий инструмент.

В рамках алгоритма, для каждого вида поверхности реализована своя процедура обработки, определяемая спецификой формы ОО. Сначала, на поверхностях ОО с заданным шагом строятся линии обработки, геометрия которых зависит от типа ОО. Далее, на линиях обработки также с определённым шагом вычисляются точки резания, по которым затем пройдёт обрабатывающий инструмент.

Рассмотрим примеры обработки поверхностей различной формы.

- При обработке сферической поверхности, задаваемой центром и радиусом, реализован алгоритм расчёта полюса. В соответствии с ним, программа обработки рассматривает все достижимые точки на поверхности сферы и находит центр этого массива (полюс). Далее, из полюса генерируется множество дуг (меридианов), начинающихся в нём и охватывающих достижимую часть сферы. Принцип обработки дуг с направлениями обработки показан на рис. 6 а.

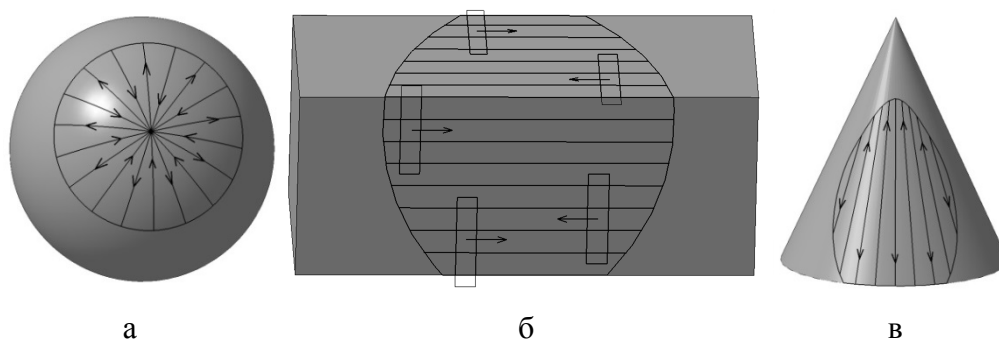


Рис. 6. Принцип обработки различных фигур: а - сфера, б – параллелепипед, прямоугольниками показан инструмент, в – конус

- Для обработки 4-угольной грани между противоположными сторонами четырёхугольника строится множество линий с заданным шагом (рис. 6 б). При этом 4-угольная грань задаётся 4 точками вершин. Следовательно, для описания 4-угольной грани всего необходимо 12 параметров, учитывая, что каждая вершина задается тремя параметрами – координатами x , y , z .

- Принцип обработки 3-угольной грани подобен принципу обработки 4-угольной. Она задаётся 3 точками (9 параметров). Построение линий обработки происходит между одной из вершин и противоположной ей стороной.

- Коническая поверхность обрабатывается по направляющим (рис. 6 в). Предполагается, что цилиндрическая поверхность является частным случаем конической поверхности.

Таким образом, в рамках алгоритма генерируются линии обработки, по которым пройдет инструмент. Однако, при этом остается открытым вопрос о достижимости точек резания инструментом с учетом его необходимой ориентации. Поэтому, необходимо проверить возможность обработки каждой такой точки.

д) Определение обобщённых координат ПМ

В рамках данного этапа рассматривается возможность обработки точки резания инструментом ПМ. При этом возникают следующие ситуации:

1. точка не находится в зоне достижимости ПМ;
2. в точке происходит столкновение инструмента и звеньев ПМ между собой или с другими объектами;
3. при переходе в точку резания имеет место смена конфигурации ПМ, во время которой изменение обобщённых координат некоторых звеньев в десятки раз превышает их обычные изменения. Это может привести к столкновению и поэтому смен конфигурации ПМ лучше избегать;

4. точка резания достижима и в ней не происходит указанных выше коллизий.

Для предотвращения этих коллизий необходимо решить обратную задачу по положению ПМ, а именно - по известному положению СК фланца или СК точек реза определить обобщенные координаты ПМ. Это позволяет найти положения звеньев ПМ, и соответственно – определить, отсутствует ли при этом соответствующие коллизии.

При реализации алгоритма проверки столкновений была использована библиотека ColDet [5]. Она предназначена для решения задачи определения пересечения объектов, аппроксимированных треугольными гранями. Поверхности задаются в виде координат вершин треугольников. Таким образом, для ПМ генерируются только те точки, в которые он может выйти.

е) Выбор траектории резания

На этой стадии расчёта происходит генерирование траектории движения ПМ для обработки ОО. Назовём траекторией резания – массив точек, заданных обобщёнными координатами, в которые ПМ должен последовательно выйти для выполнения реза вдоль соответствующей линии обработки. При этом существенно, что для каждой линии обработки может существовать несколько различных траекторий резания. Это зависит от:

- положения точки реза инструмента;
- ориентации инструмента;
- конфигурации ПМ (зависит от варианта решения обратной задачи по положению).

После рассмотрения всех вариантов траекторий резания, происходит отбор, в результате которого выбирается траектория, которая позволяет захватить наибольшую часть линии обработки. При этом для плоских поверхностей учитывается, что при резании вдоль одной линии также обрабатываются и другие (см. рис. 6 б).

Рассмотрим варианты резания несколько подробнее.

В конкретной реализации лезвие инструмента имеет длину 60 см. На нём заранее через равные промежутки заданы 5 точек реза (рис. 7), которыми ПМ может вести обработку ОО. Резание может вестись как одной, так и другой стороной инструмента. Также и сам инструмент может быть развёрнут на 180°.

Используемый в проекте ПМ KUKA KR10 R1100 sixx имеет 6 степеней подвижности и соответственно 6 обобщённых координат. Это означает, что обратная задача по положению имеет ограниченное число решений [6, 7]. В приведенных ниже вариантах не допускается перекручивание и излом кабеля:

- полный поворот 6 оси. Программно допустимый диапазон вращения 6 оси ПМ составляет $-358...358^\circ$, соответственно в большей части ситуаций оно может быть повернуто на 360° в одну, либо в другую сторону. После поворота, СК фланца в СК ПМ не изменится.
- перекручивание 4, 5, 6 осей. Диапазон вращения 4 оси ПМ составляет $-185...185^\circ$. При повороте 4 оси на $\pm 180^\circ$, смене знака угла поворота 5 оси и вращении 6 оси 180° , СК фланца в СК ПМ также останется прежним.

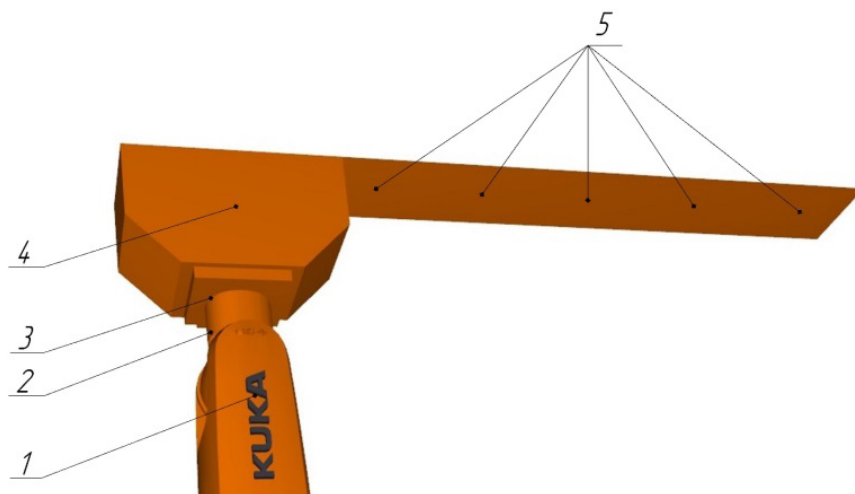


Рис. 7. Промышленный робот KUKA KR10 R1100 sixx с инструментом: 1-3 – 4,5,6 звенья ПМ, 4 – инструмент, 5 – положения точки реза

Таким образом, в результате работы программы генерируется набор траекторий резания, охватывающий всю достижимую часть ОО. Однако, стоит отметить, что при перемещении между траекториями резания возможны столкновения инструмента и звеньев ПМ между собой и с другими объектами. Планирование траектории ПМ для выполнения этой задачи не является тривиальной задачей и также требует проработки. Она решается на следующем этапе алгоритма.

ж) Построение переходов между траекториями резания

Цель данного пункта – предложить решение задачи перемещения ПМ между траекториями резания с учётом препятствий. Для этого предлагается использовать библиотеку планирования движения OMPL [8], используемую в задачах мобильной, сервисной и промышленной робототехники. Она позволяет задать:

- n -размерное пространство (в данном случае – шестимерное пространство обобщенных координат ПМ) с ограничениями на каждое измерение;

- условие проверки набора из 6 обобщённых координат на отсутствие столкновений, (для данной проверки используется ColDet);
- начальное и конечное положения ПМ;
- алгоритм планирования, в данном случае используется RRT-Connect [9, 10] (быстро исследующие случайные деревья). Он позволяет спланировать траекторию перехода между начальным и конечным положениями ПМ.

Использование данной библиотеки позволило перемещаться между траекториями обработки без коллизий. Отметим, что данный алгоритм позволяет планировать траекторию движения между любыми достижимыми точками.

3) Отправка траектории в контроллер ПМ

На завершающем этапе алгоритма, после расчета траектории движения ПМ в режиме резания и построении переходов между линиями обработки, происходит отправка траектории движения ПМ из блока управления ПМ в контроллер ПМ. Для передачи данных была применена программа EthernetKRL [11], установленная в контроллер ПМ. Она позволяет обмениваться данными по протоколу TCP/IP (Ethernet). Со стороны блока управления ПМ, отправка и приём данных происходит с помощью библиотеки libevent [12].

5. Выводы

В данной статье предложен новый алгоритм для автоматического генерирования траектории движения ПМ для обработки произвольно задаваемых сферической, конической и плоской поверхностей режущим инструментом типа «нож». При генерировании траектории учитываются зона достижимости ПМ, смена конфигураций и столкновения. Среди сгенерированных траекторий происходит отбор, в результате которого выбираются траектории, позволяющие захватить наибольшую площадь ОО. Также на основе OMPL авторами был разработан механизм планирования движения между любыми достижимыми точками с огибанием препятствий.

Список литературы

- [1] Фреймворк для программирования роботов ROS. Режим доступа: <http://www.ros.org/> (дата обращения 23.12.2015).
- [2] САМ-система для промышленных роботов Robotmaster. Режим доступа: <http://www.robotmaster.com/> (дата обращения 23.12.2015).

- [3] САМ-система для промышленных роботов SprutCAM Robot. Режим доступа: <http://www.sprutcam.com/home/sprutcam-robot> (дата обращения 23.12.2015).
- [4] САМ-система для промышленных роботов Delcam PowerMILL Robot. Режим доступа: <http://www.delcam.com/software/robotics/> (дата обращения 23.12.2015).
- [5] Библиотека для контроля пересечений 3D объектов ColDet. Режим доступа: <http://sourceforge.net/projects/coldet> (дата обращения 23.12.2015).
- [6] Фу К., Гонсалес Р., Ли К. Робототехника: пер. с англ. М.: Мир, 1989. С. 69-94. [Fu, K.-S.; Gonzales, C.S.; Lee, G.: Robotics, Mc Graw-Hill, 1989.].
- [7] Зенкевич С. Л., Ющенко А. С. Управление роботами. М.: Изд-во МГТУ им. Н. Э. Баумана, 2000. 91 с.
- [8] Библиотека для планирования движения OMPL. Режим доступа: <http://ompl.kavrakilab.org/> (дата обращения 23.12.2015).
- [9] Rity T., Anupam S., Rahul K. Intelligent Planning for Mobile Robotics: Algorithmic Approaches, 2012. 64 p.
- [10] Описание алгоритма RRT-Connect. Режим доступа: <https://personalrobotics.ri.cmu.edu/files/courses/papers/Kuffner00-rrtconnect.pdf> (дата обращения 23.12.2015)
- [11] Программный модуль для обмена данными по протоколу Ethernet TCP/IP с промышленным роботом KUKA. Режим доступа: http://www.kuka-robotics.com/russia/ru/products/software/hub_technologies/print/start.htm (дата обращения 23.12.2015).
- [12] Библиотека для оповещения о событиях и организации асинхронного ввода/вывода libevent. Режим доступа: <http://libevent.org/> (дата обращения 23.12.2015).