

Robotmaster 6.5: новый релиз ведущей системы программирования роботов на платформе Mastercam 2017

Иво Липсте (Группа компаний ЦОЛЛА и COLLA)

В июне этого года была выпущена новая версия популярной САМ-системы *Mastercam 2017*. Новая версия обозначила новую эру в развитии системы и новое поколение приложений. Как это обычно и бывает, множество нововведений, включая изменение облика *Mastercam 2017*, породило и множество обсуждений, в ходе которых была высказана масса мнений – как со стороны пользователей, так и со стороны обозревателей САМ-систем. Разработчик значительно изменил интерфейс, что существенно повысило эргономичность, а, следовательно, производительность пользователей при работе с системой, хотя и вызвало поначалу минутное недоумение в стане ветеранов, приверженных классике. В целом же новый *Mastercam* остался той же доброй “рабочей лошадкой”, но уже с новыми “вожжами”.

Впрочем, в данном случае речь пойдет не о самой системе, а о партнерском приложении *Robotmaster*, предназначенном для офлайн-программирования роботизированных ячеек. Естественно, множество партнеров, которые пишут приложения под *Mastercam*, очень активно занималось переводом своих систем на обновленную платформу, чтобы учесть и использовать все особенности версии 2017. Конечно, было бы хорошо, если обновления для всех приложений третьих разработчиков выпускались бы синхронно с выходом новых версий тех сред, в которых они призваны функционировать. Однако исторический опыт подсказывает, что это мечты, а в реальности, как правило, обновления выходят некоторое время спустя. Я бы назвал это прагматичным подходом, особенно когда изменения в “среде обитания” приложений столь существенны, как в случае *Mastercam 2017*. Ведь говорят же, что чем больше мастер, тем больше пауза... ☺

Новый релиз системы *Robotmaster* выйдет в свет в сентябре 2016 года. Естественно, главной характеристикой релиза 6.5 является способность работать в среде *Mastercam 2017*. Кроме того, есть ряд изменений, которые скрыты от глаз – это касается исправлений и улучшений алгоритмов вычислений, что в итоге проявляется как уменьшение продолжительности расчетов. Однако имеется и ряд значимых нововведений, заметных глазу, и о них сто́ит поговорить более подробно.

Управление правилами отслеживания соударений

В основном меню среды симуляции *Robotmaster* теперь доступна закладка, в которой можно

назначить правила отслеживания соударений для (рис. 1):

- составных элементов инструмента (*Tool*);
- компонентов навесного оборудования (*Tooling*);
- составных частей роботизированной ячейки (*Cell*).

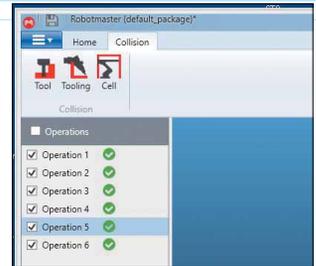


Рис. 1

Правила отслеживания соударений составных элементов инструмента

Данная функция предусматривает раздельное включение/отключение контроля соударений для патрона, нережущей части инструмента и его режущей части (рис. 2).

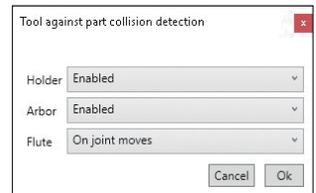


Рис. 2

Для режущей части можно включить контроль только соударений при тех движениях, которые выполняются путем вращения суставов (для простоты понимания – соударения с режущей частью инструмента при вспомогательных перемещениях на ускоренной подаче).

На первый взгляд, кажется – пусть уж система отслеживает всё, зачем что-то отключать?! Однако любой опытный технолог-программист понимает, что если включать все опции и всегда всё контролировать, то скорость расчетов существенно снизится. Нет смысла нагружать компьютер поиском соударений, которые на самом деле произойти не могут.

Возьмем такой пример: обрезка тонкостенной детали после вакуумной формовки, причем контур, по которому должен производиться рез, находится ниже модели детали. При симуляции фреза двигается как бы по телу детали, то есть в постоянном касании с деталью. Если в данном случае оставить опцию контроля столкновений с режущей частью фрезы включенной, то система будет извещать о соударениях по всей траектории реза.

Другой вариант: обрабатываем сложную скульптурную форму. В данном случае контроль за режущей частью фрезы должен быть включен, так как соударение с моделью будет, как правило, сигналом возможного зареза или подреза

обратной частью фрезы областей, не находящихся в зоне контакта фрезы с обрабатываемой моделью.

Наверное, кто-то в таких ситуациях скажет, что сама траектория инструмента была подготовлена некорректно. Возможно, иногда это так. Но чаще такого рода подрез является следствием специфической интерпретации предполагаемого движения управляющей системой робота, и отловить подобные ситуации можно только при симуляции.

Правила отслеживания соударений компонентов навесного оборудования

При создании модели навесного оборудования возможна ситуация, когда вся конструкция состоит из множества отдельных деталей, часть из которых (чисто вследствие конструктивного решения) ничего задеть не может – ни самого робота, ни компоненты ячейки, ни обрабатываемую деталь, ни элементы фиксирующей оснастки, ни заготовку. Естественно, нет смысла нагружать систему контролем соударений, которые произойти не могут – следовательно, его можно отключить.

Есть и другой пример, когда управление отслеживанием соударений существенно помогает в отладке траектории. На основе данного примера мы и разберем средства управления.

Итак, при выборе функции “правила отслеживания соударений компонентов навесного оборудования” мы видим пять деталей, входящих в состав навесного оборудования (рис. 3). В данном случае это дооборудованный краскопульт.

Для каждого из элементов мы можем указать, будет он контролироваться на соударение или нет. Также существует вариант назначения контроля элемента при выполнении вспомогательных движений. Если есть необходимость, мы можем поменять цвет, которым элемент будет отображаться во время симуляции.

Обратите внимание, что контроль детали ВЕЕМ выключен, а в разделе цвета мы назначили отображение в полупрозрачном виде. Деталь ВЕЕМ – это модель струи краски (рис. 4).

При прогоне симуляции мы четко видим пятно на поверхности, куда попадает краска, но соударение в данном случае не фиксируется. Если бы

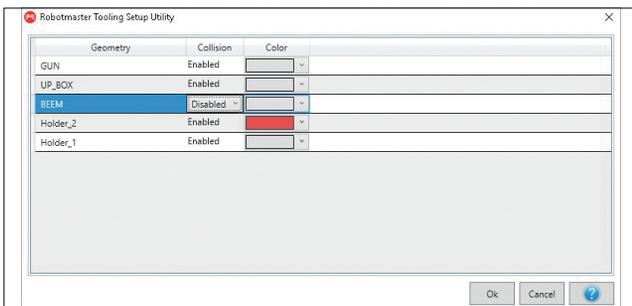


Рис. 3

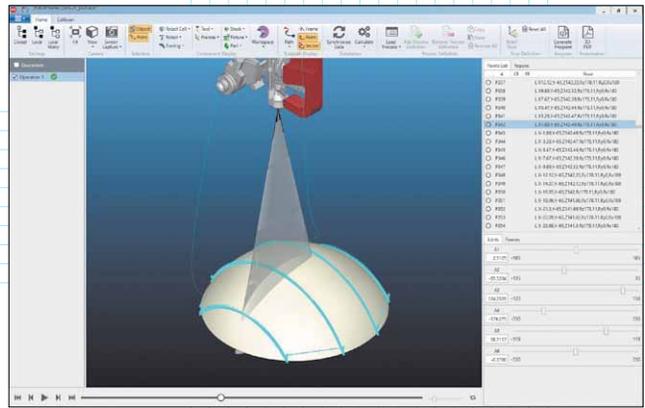


Рис. 4

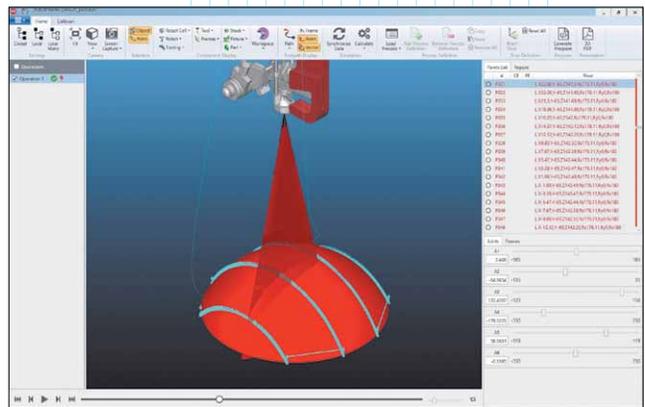


Рис. 5

контроль не был отключен, то система постоянно сигнализировала бы о проблемах (рис. 5), и мы в этом случае не смогли бы определить возможные соударения в других местах.

Правила отслеживания соударений составных частей роботизированной ячейки

Нередко бывают ситуации, когда возникает необходимость переназначить или добавить правила отслеживания соударений составных частей компонентов или объектов, находящихся в ячейке. Для этого предусмотрена соответствующая функция (рис. 6).

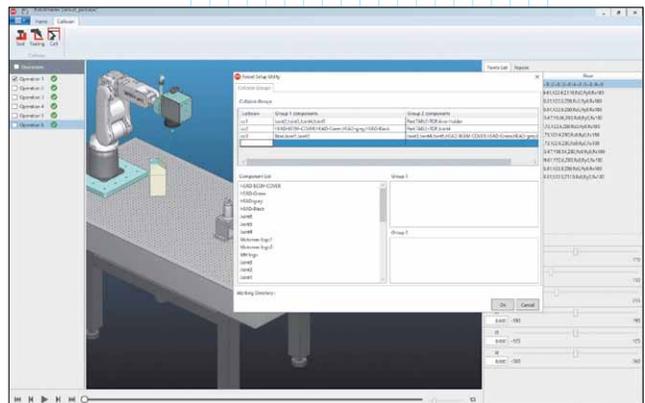


Рис. 6

На картинке мы видим небольшого робота с достаточно громоздким навесным оборудованием – лазерной головкой. Естественно, при движении возможны разные комбинации ситуаций, когда что-то с чем-то может соудариться. Если посмотреть на окно с меню, то в верхней части (*Collision groups*) можно увидеть уже назначенные группы контролируемых соударений. Подход тут прост: определяется один набор элементов ячейки (*Grup1*) и другой набор (*Grup 2*), которые будут контролироваться на взаимное соударение. Наличие нескольких групп обуславливается тем, что чисто математически одной группы часто недостаточно для подробного описания ситуации.

В нижней части окна слева мы видим список элементов, входящих в ячейку, а справа – два поля для внесения элементов, предназначенных для назначения новых наборов, подлежащих контролю. Заполнение полей происходит путем перетаскивания элементов в нужную группу.

К функциям контроля соударений нужно относиться очень внимательно, поскольку то, что здесь назначено, система *Robotmaster* будет затем использовать и учитывать в расчетах при различных оптимизациях, позволяющих уйти от множества проблем, связанных с особенностями робота, и от всяческих коллизий.

Автоматическая оптимизация

Начиная с текущей версии, производитель системы *Robotmaster* предлагает совершенно потрясающую новую функцию – автоматическую оптимизацию. Но на данный момент я прошу запастись терпением и не ждать больших чудес, поскольку это лишь начало воплощения в жизнь уникального подхода к упрощению отладки УП для роботов.

Сначала визуально рассмотрим один пример, который иллюстрирует данную функцию, а комментарии и более подробные пояснения последуют потом.

На **рис. 7** мы видим крупногабаритную деталь, которой требуется обработка верхней поверхности – по классическим канонам

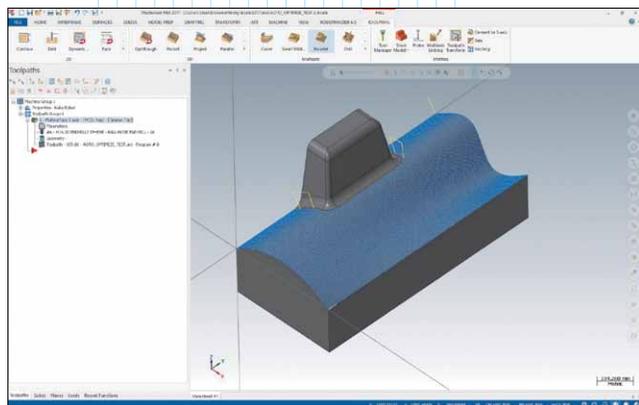


Рис. 7

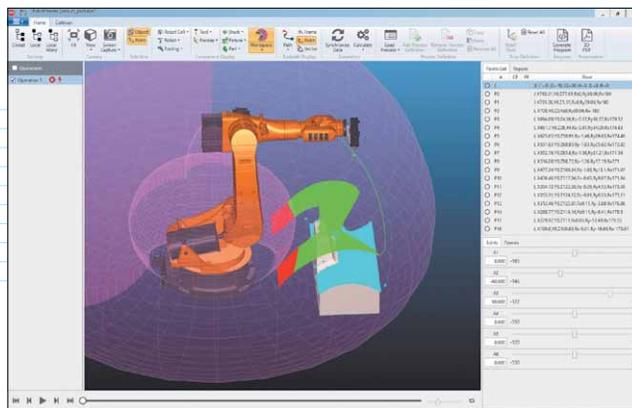


Рис. 8

5-осевой обработки, удерживая фрезу по нормали к обрабатываемой поверхности. Необходимая траектория инструмента создается с помощью функционала многоосевой обработки системы *Mastercam* (на иллюстрации траектория отображена голубым цветом).

Когда траектория сгенерирована, мы переходим в среду *Robotmaster* и активизируем меню глобальных свойств, в котором выбираем из списка предустановленных роботизированных ячеек нужную (в нашем случае это “голый” робот), а из другого списка – навесное оборудование (в нашем случае – шпиндель), а также указываем местонахождение модели детали перед роботом. В меню локальных свойств задаем установки для конкретной операции обработки.

После проверки в симуляторе мы видим (**рис. 8**), что обнаружилось множество проблем, и делаем простой вывод: без серьезной оптимизации провести инструмент по заданной траектории робот никак не сможет. При этом, несмотря на наличие проблем, мы можем детально просмотреть, как именно движется манипулятор во всех проблемных местах. Визуальный анализ показывает, что если менять положение суставов робота, то добиться нужного результата в принципе можно.

Для лучшего понимания самого процесса отладки траекторий я рекомендовал бы читателям ознакомиться с предыдущими статьями на эту тему (*Observer #1,3/2011* и др.). Ну а пользователи *Robotmaster* поймут сказанное с полуслова.

Чтобы понять, насколько сложна ситуация, подключаем к делу средства оптимизации. Сначала прибегнем к помощи стандартного оптимизатора, который позволяет использовать вращение вокруг оси инструмента для изменения положения суставов манипулятора, что зачастую обеспечивает выход из коллизий и решение других проблем, типичных для роботов.

Исходное условие в нашем случае таково: инструмент со шпинделем не вращается при прохождении по траектории. Итак, запускаем

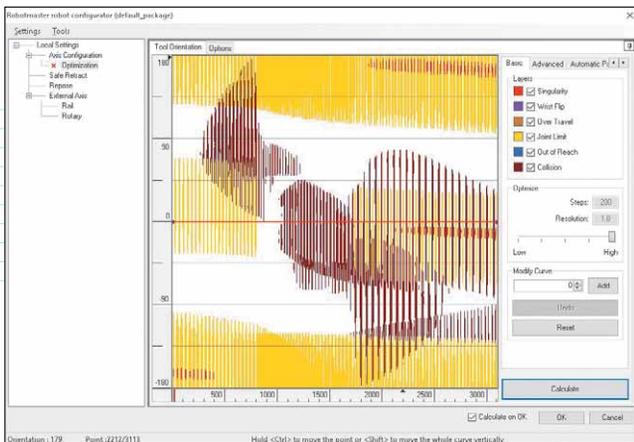


Рис. 9

расчет – и по его результатам видим достаточно страшную картину (рис. 9).

Напомним вкратце логику анализа полученной цветограммы.

Вертикальная ось показывает угол поворота шпинделя вокруг оси фрезы, горизонтальная ось отображает последовательность точек в траектории.

Как известно, рассчитанная в САМ-системе траектория состоит из отдельных точек, между которыми предполагается движение инструмента (линейно или по дуге), причем расстояние между этими точками может быть каким угодно. Оптимизатор *Robotmaster* анализирует весь диапазон поворота в каждой условной точке траектории. Важно отметить, что анализ движений робота производится по всей траектории, в том числе и в промежутках между точками, поскольку и там возможны коллизии.

На цветовой диаграмме закрашиваются соответствующим цветом те угловые значения, в которых обнаруживается проблема (то есть, при каком повороте сустава движение становится невозможным).

Белые области между цветными – зоны допустимых поворотов (то есть, там движение возможно).

Дешифровка цветовой диаграммы по типам проблем (легенда показана в правом верхнем углу экрана) дает нам следующую информацию. Имеются области, где движению мешают ограничения по вращению суставов манипулятора (желтый цвет). Есть области, где какие-то части робота и шпинделя соударяются с деталью (окрашены темно-красным), а где-то за желтым цветом скрываются еще и проблемы сингулярности (красный цвет).

При этом мы видим и обнадеживающую информацию – это непрерывная белая зона, по которой можно пройти по всей длине траектории.

Классический поход заключается в том, чтобы провести ручную сплайн, определяющий значения поворота шпинделя вокруг оси фрезы, по

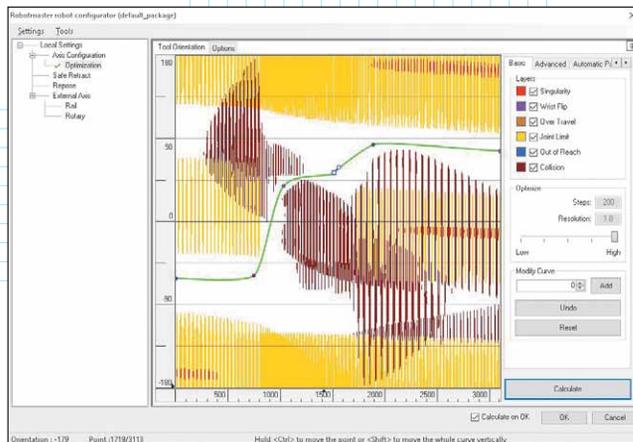


Рис. 10

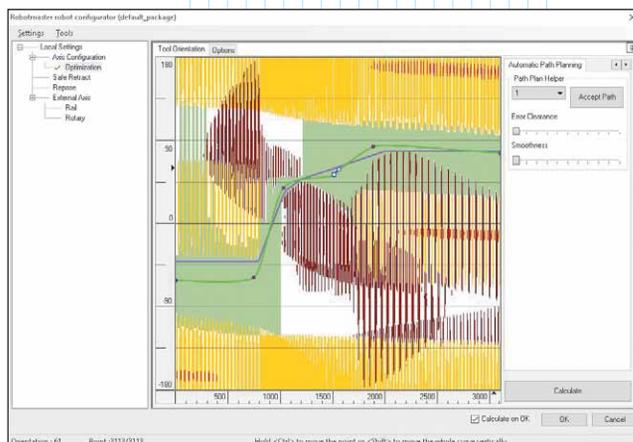


Рис. 11

всей белой зоне – с левого до правого края цветограммы. Мы сделаем небольшой эксперимент: сначала попробуем отладить траекторию в интерактивном режиме, а потом посмотрим, что нам предложит “автомат”.

О том, как работать со средствами оптимизации *Robotmaster*, я уже подробно рассказывал в предыдущих статьях, поэтому повторяться не буду. Итак, после неполной минуты, понадобившейся на проведение и изгибание и кривой, задающей правило поворота шпинделя вокруг оси фрезы, и после повторного расчета мы видим, что наша кривая стала зеленой (рис. 10). Это означает, что наш робот уже может провести инструмент по всей траектории обработки поверхности.

Теперь посмотрим, что предложит нам автоматическое средство оптимизации. В правом верхнем углу кликаем по закладке *Automatic Path Planning* (рис. 11).

Итак, теперь на цветограмме появились новые цвета. Тускло-зеленый отображает область уверенного прохождения траектории. Фиолетовая кривая – предлагаемое “автоматом” правило поворота шпинделя. Если сравнить её с зеленой кривой, которую мы определили в интерактивном режиме, то нетрудно заметить, что они весьма схожи.

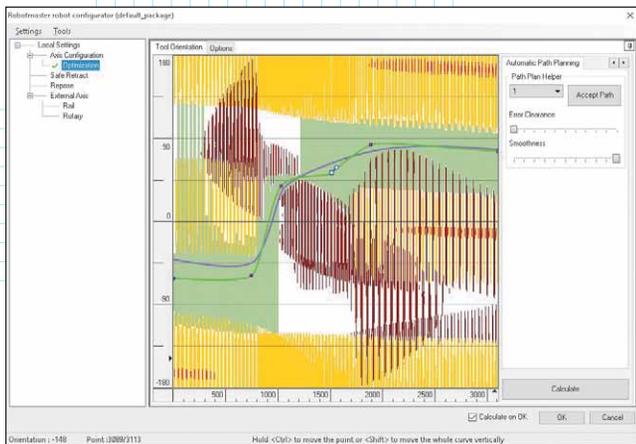


Рис. 12

В арсенале “автомата” имеется еще несколько возможностей.

Так, система может просчитать еще пару сотен комбинаций прохождения траектории. Выбирая другие номера в выпадающем списке *Path Plan Helper*, можно исследовать другие варианты.

Кнопка *Accept Path* для подтверждения выбора предлагаемой кривой – она понадобится, если пользователь согласится с каким-то из вариантов.

Два ползунка ниже позволяют управлять гладкостью предлагаемой кривой:

- *Error Clearance* – чем меньше значение (ползунок находится левее), тем меньше будет “дребезг” кривой, то есть система будет стараться по возможности рассчитать кривую с большими прямыми отрезками;
- *Smoothness* – управление сглаживанием кривой. В левом положении ползунок кривая получается угловатой – это означает, что в “острых” местах кривой робот будет делать довольно резкий поворот. Как правило, это не самый лучший вариант движения. Если же переместить ползунок правее, то кривая станет более полой и будет приближаться к форме нашей кривой (рис. 12).

Теперь у пользователя есть два варианта действий: подрихтовать свою кривую, подстраиваясь под кривую “автомата” и зеленую зону уверенного прохождения, или же просто согласиться с предложением “автомата”. Впрочем, существует и третий вариант: согласиться, и затем уже подправить кривую на свое усмотрение.

Естественно, математика – это только математика, так что советую в обязательном порядке внимательно просмотреть траекторию в режиме симуляции и оценить поведение робота. При этом особое внимание надо обратить на близость частей манипулятора и шпинделя к обрабатываемой детали и к другим объектам в зоне движения. Если дистанция подозрительно мала, то лучше вернуться к оптимизации и попробовать как-то отвернуть манипулятор подальше. Дело в том, что реальные обстоятельства в реальной

производственной ячейке наверняка будут немного отличаться от идеализированной ситуации в виртуальной ячейке, которую мы видим при симуляции. Таким образом, зорочек побольше не помешает.

Послесловие

Хотелось бы адресовать несколько слов читателям, которых заинтересовало вышесказанное, но которые почувствовали, что оценить возможности набора *Mastercam+Robotmaster* в деле подготовки УП для промышленных роботов только по одной этой статье невозможно. Так оно и есть, и для получения полной картины вам по сути дела придется прочитать всё множество предыдущих публикаций – в первую очередь, естественно, посвященных системе *Robotmaster (Observer #4,6/2013, 1/2014, 2/2016* и др.). При этом следует помнить, что без подготовленных средствами *Mastercam* траекторий инструмента, отлаживать и оптимизировать будет нечего...

Могу достаточно уверенно сказать (точнее, в очередной раз повторить), что функциональный задел, положенный создателями в основу системы *Robotmaster*, настолько велик, что вряд ли в ближайшее время конкуренты смогут хотя бы приблизиться к ней, не говоря уже о выравнивании возможностей. Не только мы, реселлеры системы *Robotmaster*, но и независимые обозреватели САМ-решений, предназначенных для подготовки УП для роботов, четко указывают, что *Robotmaster* является бесспорным лидером на этом рынке. Не стоит ловиться на уверенное применение конкурентами *Robotmaster* основных понятий, таких как офлайн-программирование, на рекламу каких-то алгоритмов, которые создают чуть ли не безупречные траектории, и прочее красноречие. Обещания обязательно нужно проверять тестированием на реальных задачах, которое и покажет их подлинную цену.

Описываемый набор средств проверки и интерактивной оптимизации УП дает возможность не только быстро выявить причины, по которым робот не может выполнить движение по заданной траектории, но и быстро отладить её, что является несомненным конкурентным преимуществом. Но каким бы совершенным ни было ПО, есть и другой аспект, который необходимо учитывать для успешного применения роботизированных ячеек: понимание технологом “строптивного нрава” подопечного робота. При формировании траекторий обработки в среде *Mastercam* следует иметь в виду, что не всегда разумно создавать их для робота точно так же, как для обычного станка с ЧПУ. В идеале надо научиться учитывать нюансы кинематики робота уже на стадии создания траекторий.

Если четкое понимание наличествует и правильные программные инструменты имеются, то подготовка обработки даже сложных деталей станет значительно проще и быстрее. 😊