Robotmaster V5 на платформе *Mastercam X5*

Новые возможности для устранения типичных коллизий при программировании роботов

Иво Липсте (COLLA, Рига) *ivo@colla.lv*

Преамбула

При выборе оборудования для задач механообработки, специалисты всё чаще сталкиваются с тем, что наиболее гибким, эффективным и экономически выгодным оказывается решение на базе индустриальных роботов (роботизированных ячеек). Не секрет, что роботам, так же как и станкам с ЧПУ, для выполнения обработки требуются управляющие программы. Сегодня любому мало-мальски грамотному специалисту понятно, что для эффективного использования станков с ЧПУ необходимо соответствующее программное обеспечение – мощная *CAM*-система. Аналогичным образом, говорить о ручной подготовке УП для роботов применительно к задачам механообработки – занятие в наше время несерьезное. К сожалению, тот софт для программирования роботов, который обычно предлагают их производители, по большей части применять вообще невозможно.

Поскольку *CAM*-системы уже десятилетиями используются для программирования обработки на станках с ЧПУ, то вполне логично предположить, что должна существовать и возможность подготовки УП для роботов непосредственно в *CAM*-среде. Такие решения известны, но практически все они способны только формально генерировать программный код для роботов, не учитывая особенностей их кинематики и самой специфики управления роботами. Можно сказать, что фактически они являются полурешениями (причем, как правило, отнюдь не дешевыми).

На сегодняшний день в мире известно только одно программное решение, которое способно качественно и эффективно обеспечить подготовку УП как для станков с ЧПУ, так и для индустриальных роботов манипуляторного типа. Речь идет о комбинации двух систем: лидера *CAM*-рынка – системы ***Mastercam*** и специализированного приложения ***Robotmaster***.

Основная идея их совместного использования заключается в следующем: с помощью богатого функционала *Mastercam* формируются эффективные траектории и последовательность обработки, а *Robotmaster* предоставляет обширный набор специальных инструментов для оптимизации и отладки УП с учетом особенностей поведения робота, что позволяет обеспечить четкое выполнение механической обработки.

В марте 2011 года канадская компания ***Jabez Technologies*** выпустила новую версию *Robotmaster,* предназначенную для работы в среде *Mastercam* версии *X5*. По мнению специалистов по программированию роботов, *Robotmaster,* предлагая столь впечатляющие новшества, очень далеко отрывается от своих и без того немногочисленных конкурентов. Маловероятно, что кто-либо из разработчиков сможет создать хотя бы отдаленно похожий по функциональности продукт в обозримом будущем.

Поскольку данная статья посвящена исключительно нововведениям, то для более подробного ознакомления со всем функционалом *Robotmaster* я рекомендую прочесть мои предыдущие публикации по этой тематике (см. *Observer #6/2009* и др.).

Как это обычно и бывает с новыми версиями, все новшества системы *Robotmaster V5* можно условно разбить на две большие группы:

- новые функциональные возможности;
- усовершенствования, повышающие производительность и быстродействие среды.

Начнем с первой группы. Сразу хочу предупредить, что для описания нового функционала, который появился в системе *Robotmaster* для *Mastercam X5,* обычного объема одной статьи будет недостаточно и потребуется продолжение.

Анализ рабочего пространства

Тем, кто занимается программированием роботов, известно, что найти такое местоположение детали, чтобы манипулятор мог дотянуться до неё со всех сторон, не так просто (это справедливо и для обратной задачи, когда деталь имеет фиксированное положение по отношению к роботу, и его нужно заставить вывернуться так, чтобы он всё же дотянулся до нужных поверхностей). Даже наличие какого-то софта, поставляемого в комплекте с роботом, не всегда помогает решить эту проблему. Сложность задачи обусловлена еще и тем, что зона досягаемости задается производителем относительно сочленения *J5* (так называемая точка ***P*** – пересечение осей вращения суставов *J4* и *J5*), а не относительно точки на конце условного инструмента, закрепленного на фланце 6-го сустава (рис. 1). Следовательно,

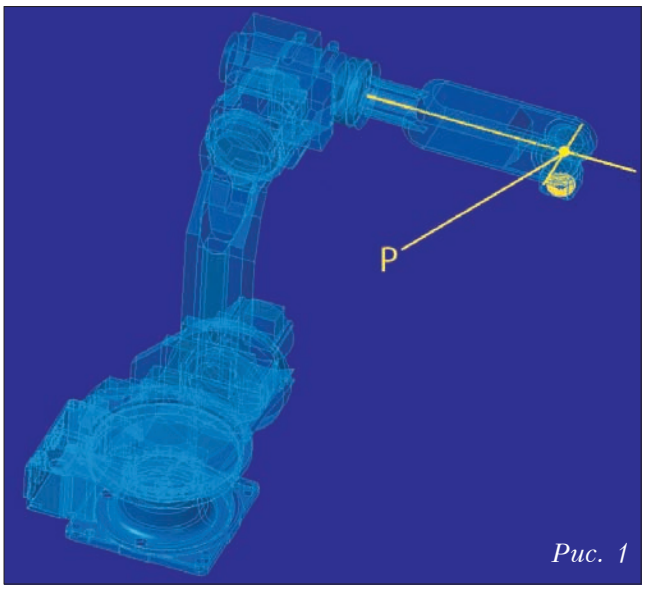

Рис. 1

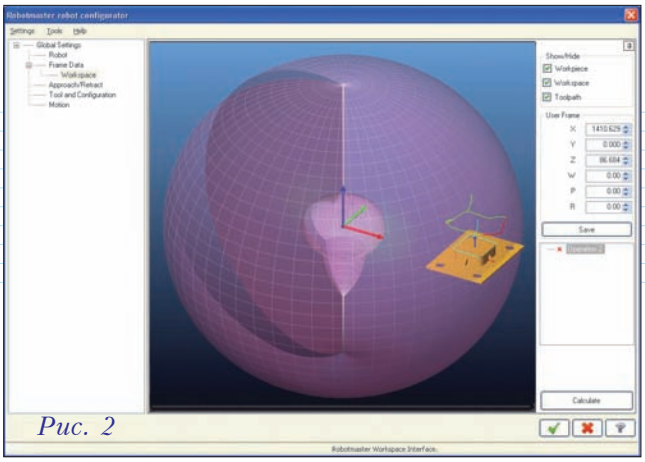

Рис. 2

глядя на конец инструмента, мы должны найти в пространстве такое положение детали (а значит и связанной с ней рассчитанной траектории обработки), чтобы траектория точки *P* всё время находилась внутри зоны досягаемости.

Для решения данной проблемы *Robotmaster* предлагает изящный инструментарий под названием “Анализ рабочего пространства”.

По заданным предельным значениям поворотов суставов робота система рассчитывает границу зоны досягаемости, которая имеет объемную округлую форму (рис. 2). Если пользователь по каким-то причинам ограничил предельные значения, введя их в систему, то при расчетах это, естественно, будет учтено.

В дереве основных параметров в разделе *Frame Dates* теперь появился подраздел *Workspace*, который открывает окно управления рабочим пространством выбранного робота, где оно отображается в виде полупрозрачного “воздушного шарика”.

Вертикальная белая область в правой части окна предлагает средства управления. Три поля в зоне *Show/Hide* позволяют показать или скрыть обрабатываемую деталь (*Workpiece*), рабочую область (*Workspace*) и траекторию (*Toolpath*). Ниже находится зона *User Frame* с шестью полями для ввода значений смещения и разворота детали по отношению к нулю робота, а также кнопка *Save* для сохранения данных. Под этой зоной выводится список выбранных для проверки операций (в нашем случае он невелик, ибо операция всего одна). Пометка в виде красного крестика рядом с операцией означает выход из зоны досягаемости.

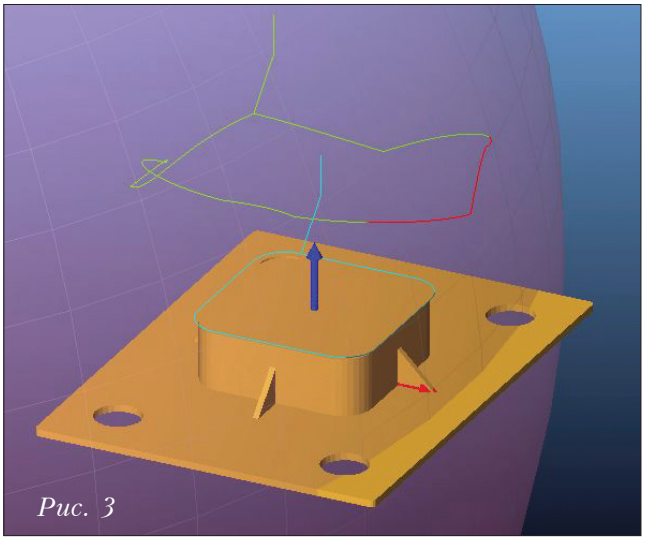

Рис. 3

Для того чтобы понять, в чём заключается проблема, увеличим картинку (рис. 3). Теперь мы четко видим две траектории. Нижняя (голубая), проходящая по кромке детали, – это траектория, по которой движется инструмент (в нашем случае фреза). Верхняя (зеленая, с красным фрагментом) – это траектория точки *P*, которую “выписывает” манипулятор для обеспечения движения инструмента по заданной (голубой) траектории. Красный фрагмент сигнализирует нам о выходе из зоны досягаемости. Чтобы разобраться, что именно и куда вышло, и как нам следует сместить деталь для устранения проблемы, можно переключиться на изображение графической области в четырех проекциях (рис. 4).

Итак, наводим курсор мышки на нужную проекцию детали, нажимаем левую клавишу и, удерживая её, перемещаем деталь до тех пор, пока не исчезнет красная зона на траектории

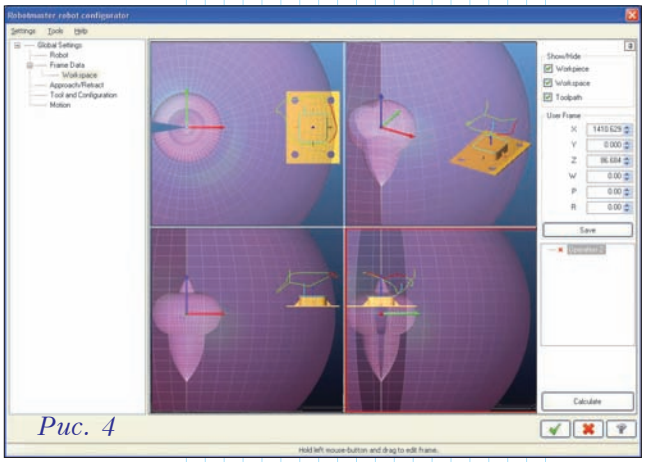

Рис. 4

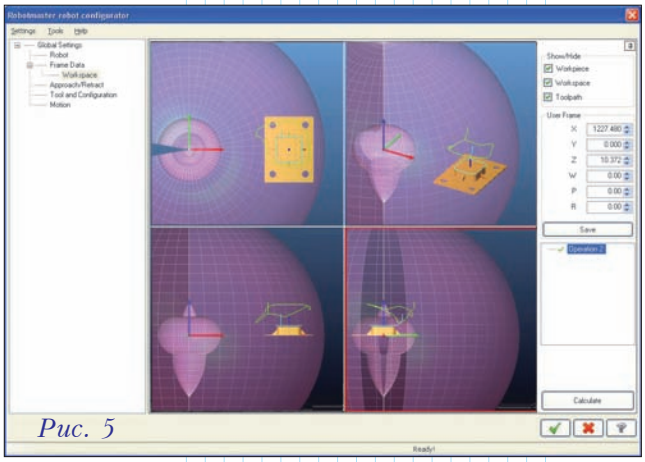

Рис. 5

точки *P*. После этого выполняем перерасчет для новой позиции (кнопка *Calculate*). Если красный фрагмент траектории поменял свой цвет на зеленый, а в списке операций вместо крестика появилась зеленая галочка, то мы достигли цели и манипулятор действует в зоне досягаемости (рис. 5).

В том случае, когда значения смещений по какой-то причине нас не устраивают, их можно изменить вручную в зоне *User Frame*. Но нельзя забывать о проверке – после ввода следует нажать кнопку пересчета.

Когда мы убедились, что всё в порядке, надо нажать *Save* и новые значения автоматически запишутся в установках *Frame Data*.

С помощью описанных выше достаточно простых манипуляций можно сэкономить уйму времени при подготовке УП для робота. Следует отметить, что работа над усилением данной функции продолжается. Уже в ближайших сервиспаках, как уверяют разработчики, мы увидим, что она будет не только управлять смещением исполнительных механизмов самого́ робота, но и учитывать форму рабочего пространства при использовании дополнительных устройств (таких, как поворотные агрегаты и линейные направляющие, которые расширяют зону действия и увеличивают количество управляемых осей).

Управление наклоном инструмента

Нередкими являются ситуации, когда при обрезке выпуклых деталей (особенно при обрезке кромки) нерабочие части шпинделя или другого навесного оборудования задевают деталь. Для того чтобы устранить касание, сохранив контур обрезки, можно немного наклонить инструмент (если появление некоторого углового отклонения на кромке является допустимым).

Как раз для таких случаев разработчики *Robotmaster* предлагают средство управления наклоном инструмента. По реализации оно чем-то напоминает оптимизацию разворота навесного агрегата вокруг оси инструмента, подробно описанную в прежних публикациях, и управление тоже осуществляется интерактивно-графическим способом. Разработчики отнесли эту новую функцию к средствам оптимизации, так как она позволяет интерактивно управлять наклоном инструмента и взаимодействует с другими доступными видами оптимизации.

Ниже мы на примере разберем в подробностях процесс отладки обработки – как с помощью нового средства управления наклоном инструмента можно устранить столкновение “нерабочих” частей робота и навесного оборудования с деталью (под рабочей частью мы в данном случае подразумеваем условное тело фрезы).

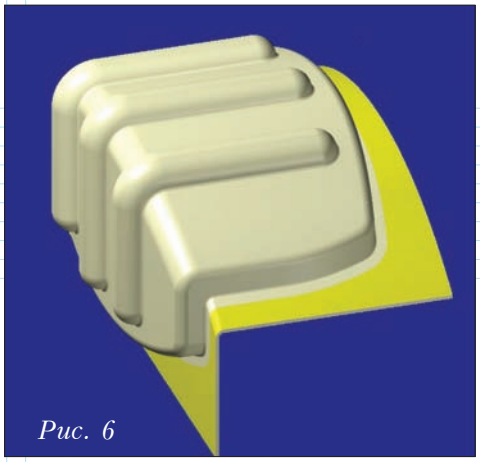
Рис. 6

На рис. 6 изображена деталь, которую требуется обрезать по контуру (отрезается область лимонно-желтого цвета).

Первым делом создаем в среде *Mastercam* 5-осевую траекторию обрезки детали по контуру. Затем, назначив основные параметры для робота, запускаем встроенный симулятор для проверки движений манипулятора. Система сразу же выдает сообщение о наличии проблемы – первая же точка траектории находится вне зоны досягаемости! Чтобы разобраться с этим, воспользуемся старым добрым методом оптимизации поворота инструмента вокруг оси (рис. 7), поскольку это даст нам возможность увидеть проблемные зоны по всей длине траектории.

После запуска оптимизатора мы отчетливо увидим на цветной диаграмме, что, если не разворачивать шпиндель вокруг оси, области в начале и в конце траектории будут недоступны для манипулятора (обратите внимание на правую часть окна с цветными квадратиками, которые информируют нас, каким цветом обозначается какой вид коллизии). Чтобы устранить эту проблему, переместим кривую управления поворотом инструмента в белую область и повторим расчет.

Возможно, после первого перемещения и перерасчета красная кривая еще не станет зеленой (зеленый цвет означает, что изменение положений суставов манипулятора при движенни по всей длине заданной траектории обработки не вызывает проблем, и обеспечивается правильное перемещение инструмента). Это связано с тем, что при развороте шпинделя вокруг оси инструмента изменятся углы поворота суставов – в этом случае, как правило, меняются и области коллизий.

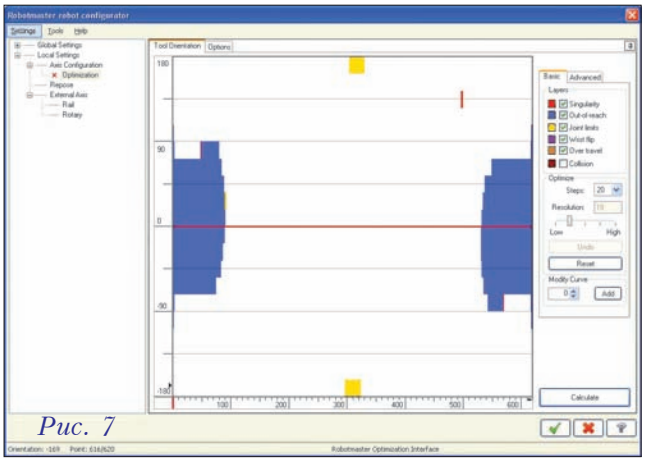
Рис. 7

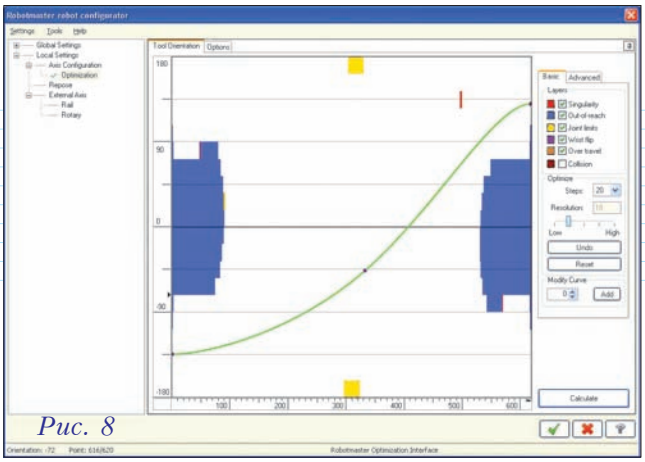
Рис. 8

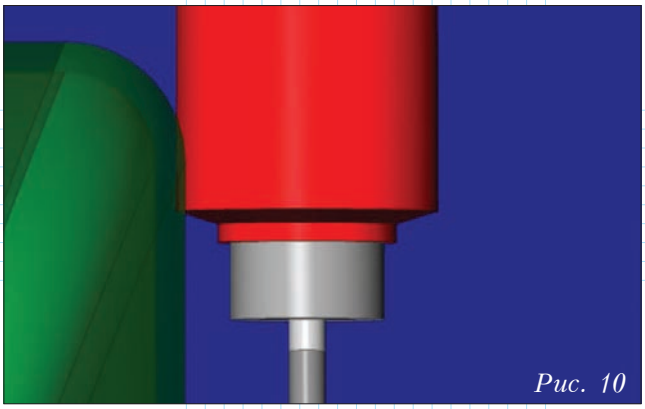
Рис. 10

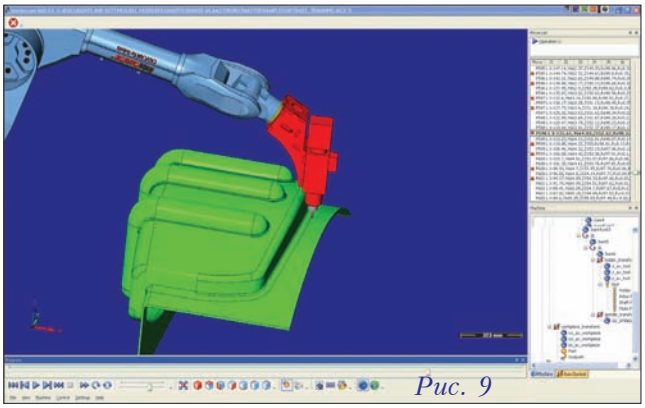
Рис. 9

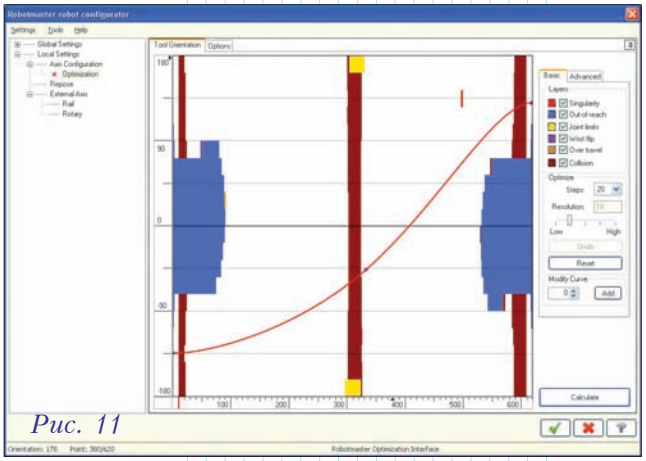
Рис. 11

Но мы всегда сможем, быстро и не углубляясь в исследования, найти такую форму кривой поворота (рис. 8), которая обеспечивает желаемую обрезку детали.

Далее нам необходимо повторить процесс симуляции обработки, чтобы убедиться, что все проблемы устранены. В нашем случае проверка показывает, что появилась новая проблема – столкновение корпуса шпинделя с деталью (рис. 9). Красный цвет шпинделя (рис. 10) информирует нас о том, что коллизии есть; в зоне, где выводятся координаты траектории, проблемные точки отмечаются красным крестиком.

Возникает вопрос – почему мы не обнаружили столкновение в оптимизаторе? Дело в том, что система предусматривает опции – как проверять траекторию при пересчете. Очевидно, что этот вид коллизий мы не указали (это дало некоторый небольшой выигрыш по времени), а симуляция по определению выявляет все проблемы. Поэтому нам придется вернуться в режим оптимизации, ужесточить требования проверки и запустить перерасчет.

Коричневые полосы (рис. 11) показывают отрезки траектории, на которых имеются столкновения. Главная проблема заключается в том, что эти зоны занимают всю высоту графической области окна. Это означает, что, как бы мы ни крутили инструмент вокруг своей оси, избежать коллизий таким способом не удастся.

Визуальная оценка положения шпинделя в точке, где происходит столкновение, показывает, что для беспроблемного прохождения по данному контуру нужно в некоторых местах наклонить инструмент в направлении от детали. При этом четко видно, что наклон нужен совсем небольшой, однако точное определение угла наклона подручными средствами может оказаться делом хлопотным и долгим. К тому же, таких участков несколько. Посмотрим, что предусмотрели разработчики для такого случая. Итак, входим в режим локальных установок для данной операции (где перед этим мы уже провели оптимизацию поворота шпинделя вокруг оси инструмента) и в верхней части окна выбираем закладку *Options* (рис. 12).

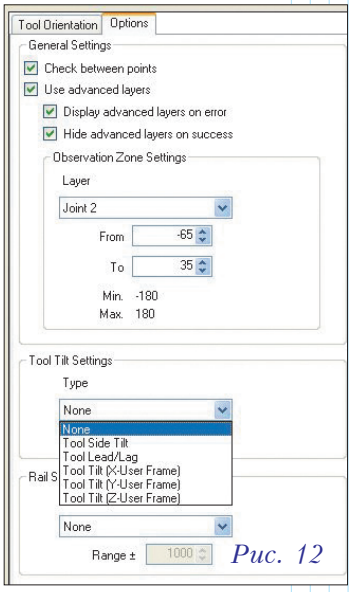

Рис. 12

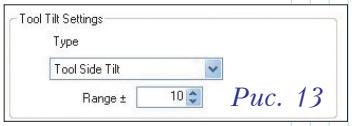

Рис. 13

В зоне установки наклона инструмента (*Tool Tilt Settings*) выбираем вариант наклона инструмента в

сторону (*Tool Side Tilt*); там же, в поле *Range*, вводим предполагаемый диапазон наклона ±5 градусов (рис. 13). После выбора, наверху экрана высвечивается закладка *Tool Side Tilt*. Щелкаем мышкой на закладке, чтобы активизировалось окно управления наклоном.

Теперь проведем расчет, предварительно включив соударения (*Collisions*) в список анализируемых проблем.

Коричневые "языки" на диаграмме (рис. 14) показывают, на каких участках траектории происходят столкновения. По левой границе графической зоны вверх и вниз от центральной линии располагается шкала, ограниченная заданным выше диапазоном, которая позволяет нам сделать вывод, что инструмент надо наклонить менее чем на один градус.

Итак, подводим курсор к линии, добавляем точки, а затем корректируем линию так, чтобы она огибала коричневые зоны (рис. 15), и повторяем расчет. Если цвет кривой после расчета становится зеленым, это означает, что проблему мы решили удачно.

Теперь мы можем перейти в режим оптимизации поворота инструмента и повторить расчет, чтобы удостовериться, что коричневые области исчезли. Ну а чтобы окончательно убедиться, что все проблемы устранены, проверим с помощью симуляции обработки (рис. 16).

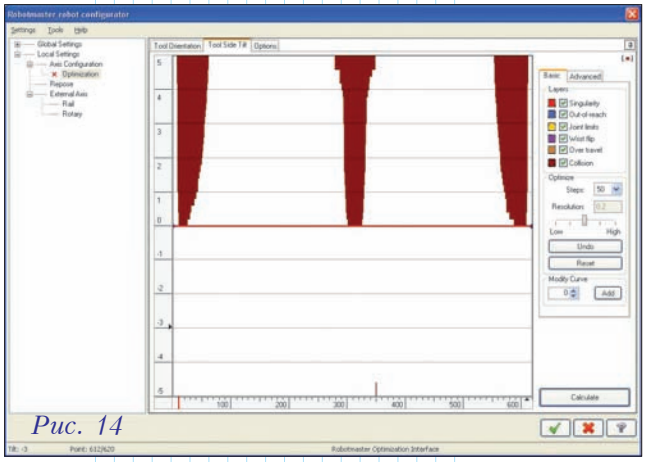

Рис. 14

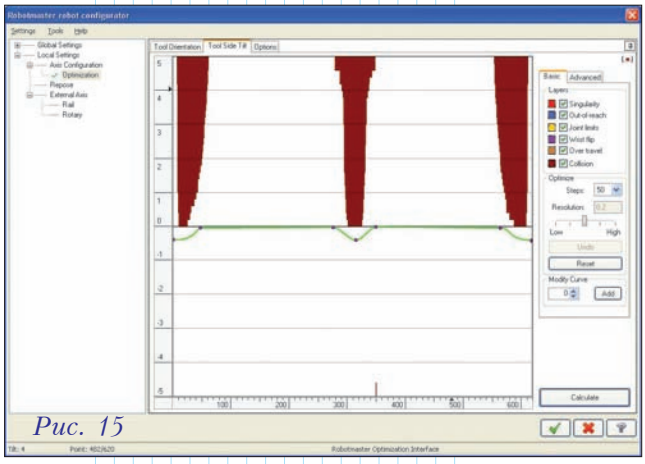

Рис. 15

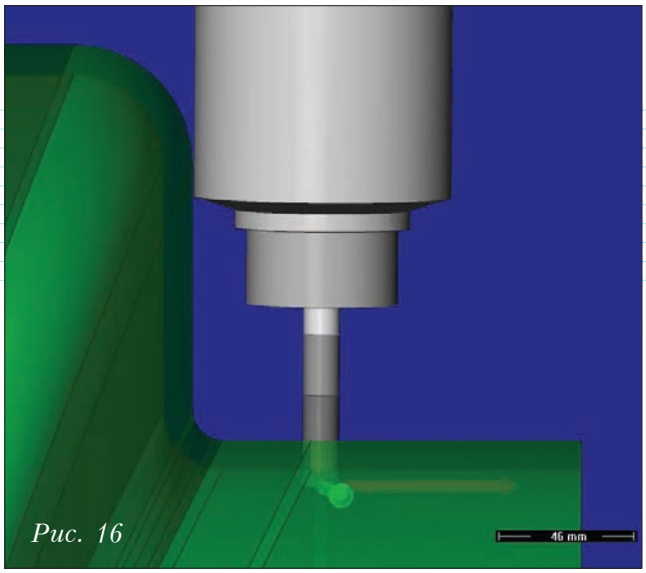

Рис. 16

Если бы было возможно продемонстрировать на странице журнала описанный выше пример вживую, то уважаемые читатели могли бы убедиться, что по времени всё это занимает несколько минут. Для получения программы для робота нам осталось всего лишь нажать кнопку генерации УП.

Хочу обратить ваше внимание на то, что хотя средства оптимизации имеют определенную специализацию, однако при этом они работают сообща. В нашем случае, если бы наклон инструмента повлек за собой, к примеру, выход из зоны досягаемости, то цвет кривой не поменялся бы на зеленый. Нам пришлось бы перейти в область оптимизации поворота и там проверить траекторию.

Оптимизация при движении по линейной направляющей

Если требуется изготавливать длинные детали, то манипулятор, как правило, устанавливается на линейной направляющей (рельсе), что дает возможность перемещать его вдоль детали, увеличивая, тем самым, зону обработки. Производственники, приобретая такие ячейки, обычно выбирают рельсы такой же длины, как у самой большой из обрабатываемых деталей. Однако, если вдуматься, рельс может быть и короче, поскольку нужно иметь в виду и те расстояния, на которые способен вытянуться манипулятор, находясь в крайних положениях на рельсе. Например, общая длина рельса составляет 5 метров, а зона досягаемости робота – 3 метра. В этом случае можно уверенно обрабатывать детали с максимальной длиной от 10 до 11 метров. Конечно, реальная зона досягаемости зависит от сложности формы детали и от её габаритов.

Существует два основных способа использования линейной направляющей. Первый – когда

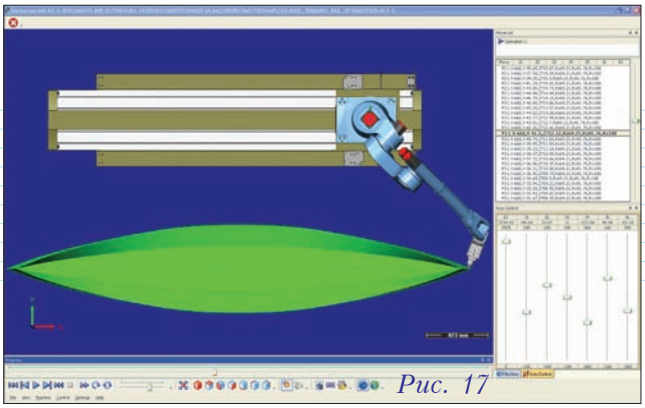

Рис. 17

позиция робота статична. В этом случае манипулятор сначала перемещается на позицию, с которой обеспечивается достаточная досягаемость, и затем обработка ведется “не сходя с места”. Это будет хорошим решением в том случае, когда роботизированная ячейка состоит из нескольких технологических зон или когда обработка длинной детали ведется по участкам.

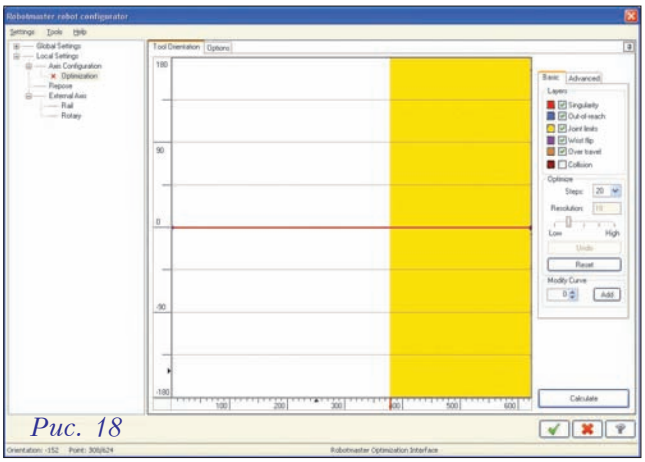

Рис. 18

При втором способе рельс служит седьмой осью (рис. 17), по которой робот постоянно перемещается в ходе обработки.

В новой версии *Robotmaster* разработчики ввели возможность оптимизации при движении робота по линейной направляющей. Рассмотрим это на примере обрезки по всей длине детали.

На рис. 17 изображена роботизированная ячейка, вдоль рельса которой расположено каноэ. Как мы видим, лодка длиннее, чем рельс. Наша задача – обрезать верхнюю боковую кромку по всей длине. Из рисунка видно, что, несмотря на использование линейной оси, постоянная ориентация манипулятора не позволяет выполнить полное движение обрезки по всей длине кромки.

Для более четкого понимания ситуации, воспользуемся оптимизатором поворота инструмента (рис. 18).

После завершения расчета с правой стороны появляется желтая непрерывная полоса, сигнализирующая о невозможности обработки во второй половине траектории.

Для лучшего понимания ситуации, отображенной в окне оптимизации поворота инструмента (рис. 18), снова прибегнем к симуляции обработки. Мы видим, что робот начинает обрезку с левого конца лодки и, не меняя своей ориентации, перемещается вправо, отрабатывая чуть больше половины заданной траектории (рис. 19).

Если мы воспользуемся способностью симулятора управлять суставами по отдельности (а также положением на рельсе) и развернем манипулятор вручную, то увидим, что траекторию вполне можно отработать по всей длине. Для этого нам только необходимо как-то убедить робота в необходимости развернуться.

В том же окне оптимизации, в закладке *Options* (рис. 20), в зоне параметров *Rail Settings*, выбираем оптимизацию при движении по линейной направляющей (*Rail 1*) и задаем

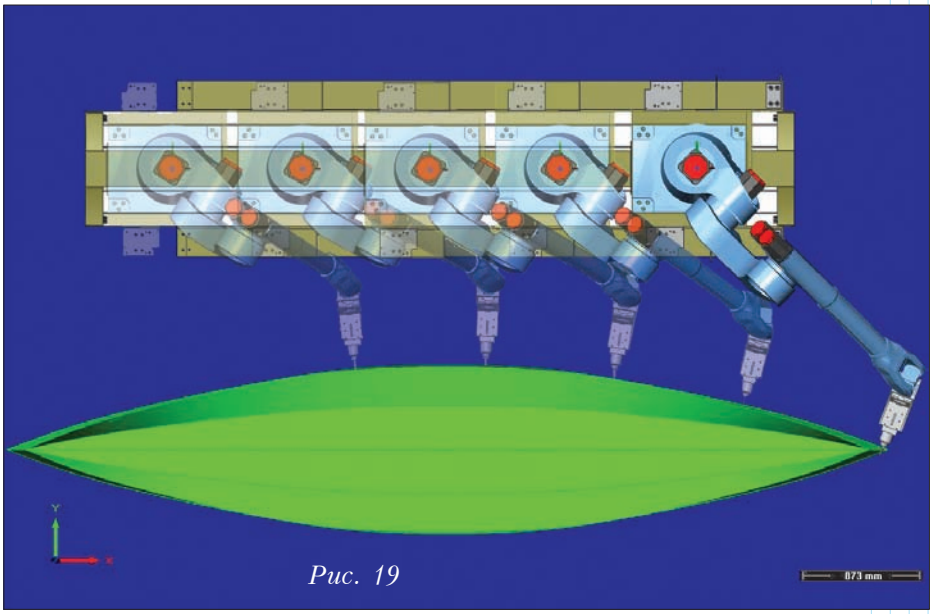

Рис. 19

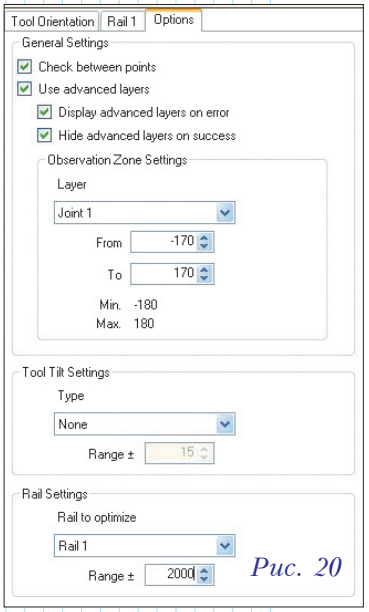

Рис. 20

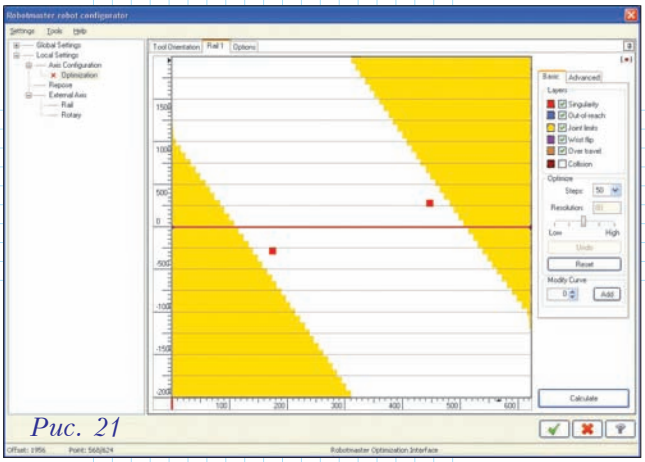

Рис. 21

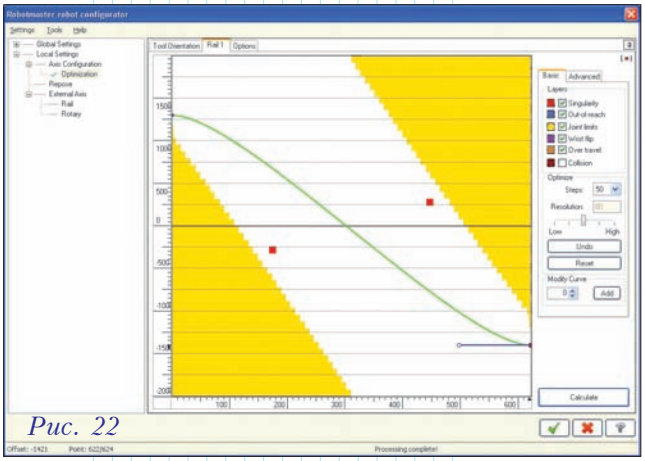

Рис. 22

значение *Range* – допустимое расстояние, на которое манипулятор может выдвинуть инструмент за пределы длины рельса.

После выбора в поле *Rail to optimize* значения *Rail 1*, наверху появляется закладка для входа в нужный нам режим, с которым мы сейчас и познакомимся поближе.

Запустив расчет, мы вскоре увидим уже знакомую нам по работе с другими видами оптимизации цветную картинку (рис. 21). На ней явно просматривается белая область, в которой, предположительно, можно обеспечить необходимое движение инструмента по траектории обработки. Как и в предыдущем случае, изменим с помощью мышки форму кривой так, чтобы она полностью попала в белую зону (рис. 22), и повторим расчет.

Итак, кривая позеленела, а это означает, что всё в порядке! Разберемся, что у нас получилось.

Желтые области информируют о выходе за пределы длины линейной направляющей. По шкале слева можно удостовериться, что величины этих выходов не превышают заданного нами в параметрах значения. Обратите внимание на желтый треугольник слева внизу. На шкале его вершине соответствует значение примерно 1250 мм – это значит, что траектория обработки выходит за пределы рельса именно на эту величину. Переместив левый конец кривой над этой вершиной в белую зону, мы развернули робот в положительном направлении относительно его основания. Соответственно, положение правого конца кривой означает разворот на другом конце траектории в отрицательном направлении. Если смотреть по ходу кривой, то становится понятно, что мы задали правило – как разворачивается робот при движении по направляющей.

Если повторить расчет в окне оптимизации поворота инструмента, мы убедимся, что желтая полоса, сигнализирующая о наличии проблемы, исчезла (здесь вам придется поверить мне на слово, без иллюстрации – она не поместилась).

На рис. 23 представлено условное отображение положений робота при обработке. Хорошо видно, что по мере перемещения вдоль рельса робот разворачивается, что позволяет ему безупречно выдерживать требуемую для обрезки кромки траекторию инструмента.

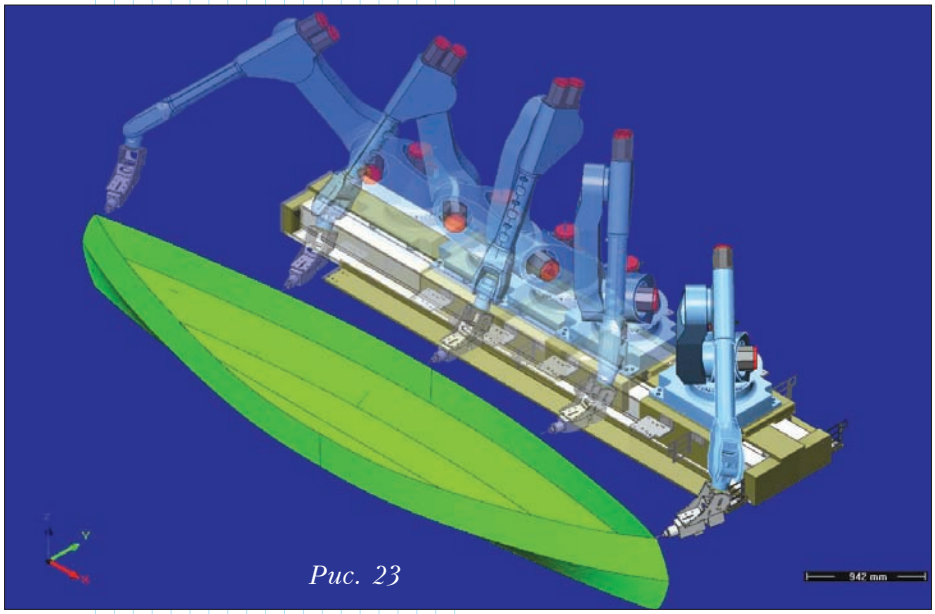

Рис. 23

Хочу еще раз подчеркнуть, что все виды оптимизации работают совместно, дополняя друг друга для более эффективного обнаружения и устранения коллизий с целью обеспечить движение инструмента по строго заданной траектории обработки.

В следующем номере журнала мы продолжим знакомство со средствами оптимизации и другими возможностями новой версии *Robotmaster*.

(Продолжение следует)