DOI: 10.18137/RNU.V9187.24.01.P.48 УДК 681.51

Я.Ю. Пикалов, М.В. Сарамуд, Н.В. Штабель, С.Б. Ткачев

АЛГОРИТМЫ И ПРОГРАММЫ АВТОМАТИЧЕСКОГО БАЗИРОВАНИЯ НА РОБОТИЗИРОВАННОМ ТЕХНОЛОГИЧЕСКОМ КОМПЛЕКСЕ ФРЕЗЕРНОЙ ОБРАБОТКИ

Аннотация. Представлено решение проблемы быстрой настройки рабочей системы координат роботизированного комплекса механической обработки в условиях динамично изменяющейся номенклатуры обрабатываемых изделий. Описывается система уравнений, алгоритм вычисления положения и ориентации рабочей системы координат детали относительно исходной системы координат робота, а также их реализация в виде программы автоматического базирования для двух технологических оснасток. Программа написана на языке Kuka Robot Language для робота KUKA KR90 R3100 с датчиком касания Renishaw OMP-40 и включает этапы диалогового выбора технологической оснастки и номера измеряемой базы, а также соответствующее им автоматическое измерение точек базовых поверхностей, вычисления значений линейных и угловых параметров положения рабочей системы координат и запись полученных значений в соответствующие переменные системы управления роботом. Кроме того, описываются подходы по настройке постпроцессоров для подготовки управляющих программ базирования с использованием CAM-сред. Предложенные решения в виде управляющих программ и настроенного постпроцессора успешно используются на производстве при обработке углепластиковых изделий.

Ключевые слова: промышленная робототехника, фрезерование, система управления роботом, автоматическое базирование, управляющая программа.

Ya.Yu. Pikalov, M.V. Saramud, N.V. Shtabel, S.B. Tkachev

ALGORITHMS AND PROGRAMS FOR AUTOMATIC BASING ON A ROBOTIC MILLING PROCESSING COMPLEX

Abstract. The article presents a solution to the problem of quickly setting up the working coordinate system of a robotic machining complex in a dynamically changing range of machined products. The system of equations, the algorithm for calculating the position and orientation of the working coordinate system of the part relative to the initial coordinate system of the robot, as well as their implementation in the form of an automatic locating program for two technological equipment are described. The program is written in the Kuka Robot Language for the KUKA KR90 R3100 robot with a Renishaw OMP-40 touch sensor and includes the steps for the dialog selection of technological equipment and the number of the measured base, as well as the corresponding automatic measurement of points of base surfaces, calculation of the values of linear and angular position parameters working coordinate system and recording the obtained values into the corresponding variables of the robot control system. In addition, approaches to setting up postprocessors for preparing control programs based on the use of CAM environments are described. The proposed solutions in the form of control programs and a customized postprocessor are successfully used in production when processing carbon fiber products.

Keywords: industrial robotics, milling, robot control system, automatic basing, NC program.

Работа выполнена при поддержке Министерства науки и высшего образования Российской Федерации (госконтракт № FEFE-2020-0017).

Пикалов Яков Юрьевич
кандидат технических наук, ведущий научный сотрудник научно-исследовательской лаборатории «Робототехнические системы», Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева, город Красноярск. Сфера научных интересов: программирование механической обработки, автоматизация технологических процессов машиностроения, робототехника. Автор 53 опубликованных научных работ. SPIN-код: 1728-3027, AuthorID: 606492.
Электронный адрес: yapibest@mail.ru

Сарамуд Михаил Владимирович
кандидат технических наук, старший научный сотрудник научно-исследовательской лаборатории «Робототехнические системы», Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева, город Красноярск. Сфера научных интересов: надежность программного обеспечения, робототехника, системный анализ, оптимизация, имитационное моделирование. Автор 88 опубликованных научных работ. SPIN-код: 3747-8689, AuthorID: 744907.
Электронный адрес: msaramud@gmail.com

Штабель Николай Владимирович
научный сотрудник научно-исследовательской лаборатории «Робототехнические системы», Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева, город Красноярск. Сфера научных интересов: автоматизация, робототехника, телекоммуникации, программирование. Автор 42 опубликованных научных работ. SPIN-код: 3067-5803, AuthorID: 860724.
Электронный адрес: shtabnik@gmail.com

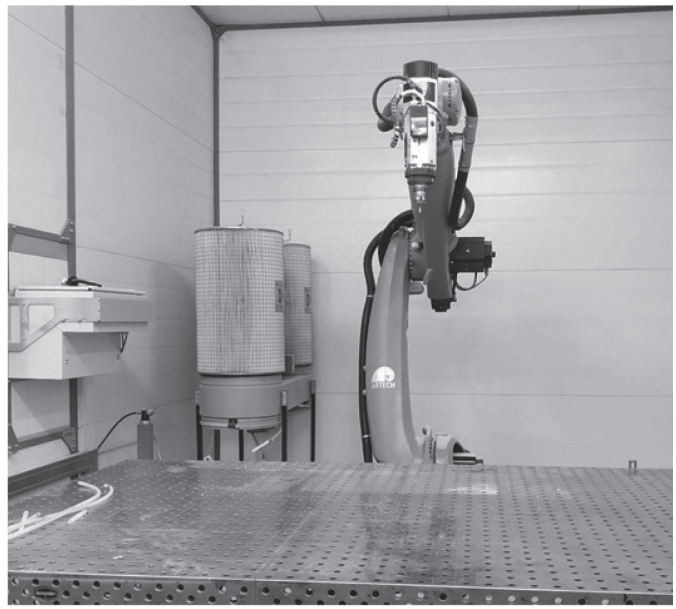
Ткачев Степан Борисович
научный сотрудник научно-исследовательской лаборатории «Робототехнические системы», Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева, город Красноярск. Сфера научных интересов: автоматизация, робототехника, силовая электроника. Автор 19 опубликованных научных работ. AuthorID: 597441.
Электронный адрес: steep_st@mail.ru

Введение

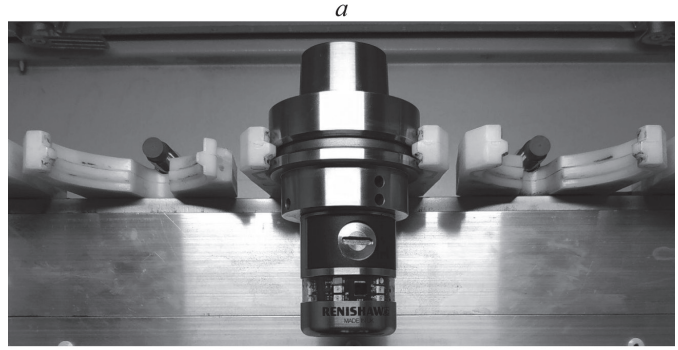
За последнее десятилетие область автоматизации различных производственных процессов существенно расширилась. Если сначала автоматизации подвергались простые этапы, например подготовка конструкторской документации, то на данный момент автоматизируется практически весь жизненный цикл изделия. При этом машины становятся «исполнителями» на производстве, а человек меняет свою роль на «руководителя» производственными процессами.

На машиностроительных предприятиях широкое распространение получили роботизированные технологические комплексы, реализующие механическую обработку деталей. В качестве исполнительного механизма может быть использован фрезерный шпиндель, установленный на манипуляторе, заготовка закреплена на станке (см. Рисунок 1).

Достоинства таких решений заключаются в возможности выполнения пятиосевой обработки при относительно небольшой стоимости в сравнении с традиционными пятиосевыми станками с численно-программным управлением (далее – ЧПУ).



а



б

Рисунок 1. Робот KUKA KR90 R3100 с установленным шпинделем (*а*) и измерительный щуп (*б*)
Источник: фото авторов.

Как и для станков с ЧПУ, обработка деталей производится по управляющей программе, разрабатываемой в CAM-среде технологом-программистом. Задача оператора заключается в сборке необходимого инструмента, подготовке технологической оснастки, закрепления, снятия и базирования обрабатываемой заготовки и выполнении управляющей программы обработки.

Актуальность использования

Аналогично со станками в качестве основного измерительного элемента в таких роботизированных комплексах используется контактный щуп с датчиком касания (см. Рисунок 1, *б*). Использование различных сканирующих систем или систем на основе технического зрения с алгоритмами машинного обучения, которые применяются в сварочных комплексах для позиционирования заготовок [1], является неактуальным ввиду более низкой точности и сложной технической реализации.

Существует ряд исследований, направленных на поиск оптимального положения заготовки относительно базовой системы координат робота, обеспечивающих повышение точности обработки с учетом структурно-динамической [2; 3] либо эластичной [4] моделей, либо на основании экспериментальных данных [5], а также с конфигурациями, требующими минимально возможных крутящих моментов в соединениях при максимально возможной манипулятивности [6].

Авторы статьи [7] предлагают метод одновременной оптимизации и положения робота, и установки заготовки.

Однако в описанных ранее исследованиях существуют две проблемы. Во-первых, на практике не всегда можно разместить заготовку в оптимальной зоне; во-вторых, в процессе обработки криволинейных поверхностей может происходить существенное изменение позы робота, в результате чего также изменяются его жесткость, точность, скорости и ускорения его сочленений, что, в свою очередь, уменьшает ценность ранее представленных исследований [8; 9]. В ряде случаев можно сохранить оптимальную по жесткости позу робота, если использовать поворотный стол, который будет ориентировать заготовку нужным образом [10]. Однако данный элемент является менее жестким в сравнении с неподвижной монолитной конструкцией стола, а также не решает полностью проблему вынужденного изменения позы робота в процессе фрезерной обработки.

Другими направлениями повышения качества обработки уже имеющейся позиции заготовки являются: поиск оптимальной позы робота, обеспечивающей высокие показатели жесткости для сгенерированной в CAM-среде траектории фрезерной обработки (за счет избыточной степени свободы робота) [11], а также изменения производительности в зависимости от вычисленных характеристик жесткости [12], функционально зависящей от позы робота [13; 14]. Однако это не решает проблем, связанных с процессом базирования изделий на столе.

Таким образом, актуальным для условий динамично изменяющейся номенклатуры деталей является поиск эффективных методов переналадки оборудования под новую технологическую оснастку или под новое ее положение и ориентацию.

В станках с ЧПУ точность формы и относительного положения поверхностей стола и направляющих достаточно высока и обеспечивается монолитностью несущей конструкции, соответственно, при базировании не требуется вычисления углового положения базовых поверхностей заготовки (технологической оснастки), то есть достаточно измерения по одной точке на базовой поверхности с использованием встроенных макросов на станке ЧПУ.

В роботизированных фрезерных комплексах сложнее обеспечить точность положения и формы стола относительно базовой системы координат робота, так как рабочий стол – это отдельное изделие, качество изготовления и монтажа которого во многом зависит от производственных возможностей организации-интегратора. Это приводит к необходимости (кроме вычисления положения базовых поверхностей заготовки) вычислять и угловую ориентацию данных поверхностей.

Обрабатываемые детали могут иметь сложную форму без характерных плоскостей и цилиндрических поверхностей, по которым удобно выполнять базирование с помощью щупа. В этом случае для каждой из деталей необходимо изготовление специальной технологической оснастки с необходимыми поверхностями для базирования. При монтаже приспособления на рабочем столе робота оператор роботизированного комплекса на

первом этапе в ручном режиме выставляет ориентацию базовых поверхностей, обеспечивая минимальные перекосы рабочей системы координат (далее – WCS) и исходной системы координат робота (далее – ICS). На втором этапе после ручной настройки ориентации WCS наладчик с помощью программ базирования по элементарным поверхностям выполняет настройку ее положения. На данную операцию может уходить от трех до пяти часов, при этом первый этап составляет около 90 % времени.

В данной статье предлагается решение этой проблемы с помощью специального математического аппарата и алгоритмов, реализованных в управляющей программе для выполнения автоматической настройки WCS, и настроенного постпроцессора, позволяющего технологу-программисту в кратчайшие сроки готовить программы базирования в CAM-среде. Программа базирования выполняет полный цикл измерения базовых поверхностей приспособления и рассчитывает положение и ориентацию WCS относительно ICS.

Математическая модель

В общем случае для однозначного определения положения тела в пространстве необходимо приложить (или измерить) шесть опорных точек, образующих комплект баз. Для призматических деталей (или деталей с призматическими базовыми поверхностями) комплект баз представлен на Рисунке 2. Также такой комплект баз называется «плоскость – прямая – точка» (далее – ППТ).

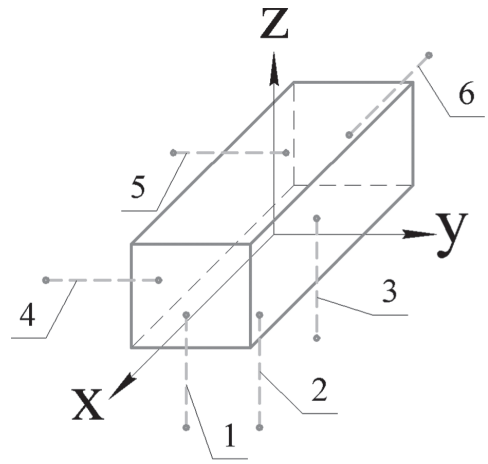


Рисунок 2. Комплект баз ППТ: опорные точки *1, 2, 3* – установочная плоскость; точки *4, 5* – направляющая/прямая; точка *6* – опорная точка
Источник: здесь и далее рисунки выполнены авторами.

Если выполнить измерения щупом по опорным точкам в соответствии со схемой на Рисунке 2, то рабочая система координат детали окажется на пересечении плоскостей с опорными точками.

Положение рабочей системы координат в системе управления роботом задается с использованием трех линейных координат Δx, Δy, Δz и трех последовательных поворотов вокруг осей: θ_A – вращение вокруг оси Z; θ_B – вокруг оси Y; θ_C – вокруг оси X.

Программа базирования была разработана под два приспособления. На Рисунках 2, 3 изображены вакуумные столы с обозначенными на них базовыми плоскостями (Пл.).

Для первого приспособления (см. Рисунок 3):

Пл. 1, 2, 3, 4 образуют установочную базу приспособления, на основании их вычисляется положение плоскости *Oxy*;

Пл. 5, 6 определяют направляющую базу и являются плоскостью *Oyz* WCS;

Пл. 7, 8 определяют опорную базу приспособления, а плоскость симметрии для этих граней является плоскостью *Oxz* WCS.

На каждой из указанных плоскостей измеряется по одной точке. По этим восьми измеренным точкам производятся все последующие расчеты, связанные с определением положения и ориентации рабочей системы координат. Последовательность измерения представлена на схеме Рисунка 5. Точки, в которых происходит измерение, обозначены в круглых сносках, при этом направление измерения обозначено на линии сноски: для установочной базы измеряются точки *3*, *9*, *13*, *20*, для направляющей базы – *10* и *14*, для опорной базы – *6* и *17*.

Для второго приспособления (см. Рисунок 4):

Пл. 1, 2 образуют установочную базу приспособления, на основании их вычисляется положение плоскости *Oxy*;

Пл. 3 определяет направляющую базу и является плоскостью *Oyz* WCS;

Пл. 4, 5 определяют опорную базу приспособления, а плоскость симметрии для этих граней является плоскостью *Oxz* WCS.

Последовательность измерения показана на Рисунке 6. На установочной базе измеряются точки *3*, *9*, *17*, *24*, для направляющей базы – *6* и *21*, для опорной базы – *10* и *18*.

Точки на Рисунках 5, 6, позиции которых взяты в круг, будут получены при измерении и использоваться для вычисления координат и ориентации плоскостей WCS (на линии выноски обозначены направления измерения).

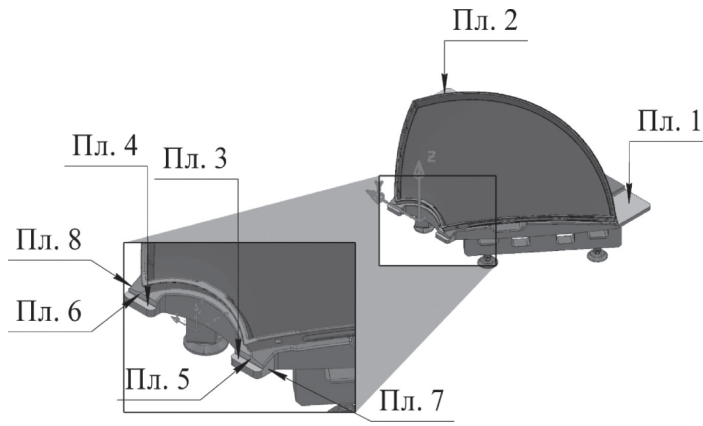


Рисунок 3. Базовые поверхности первого приспособления

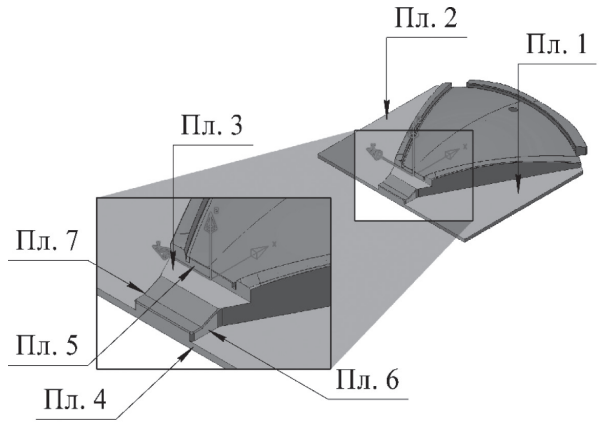


Рисунок 4. Базовые поверхности второго приспособления

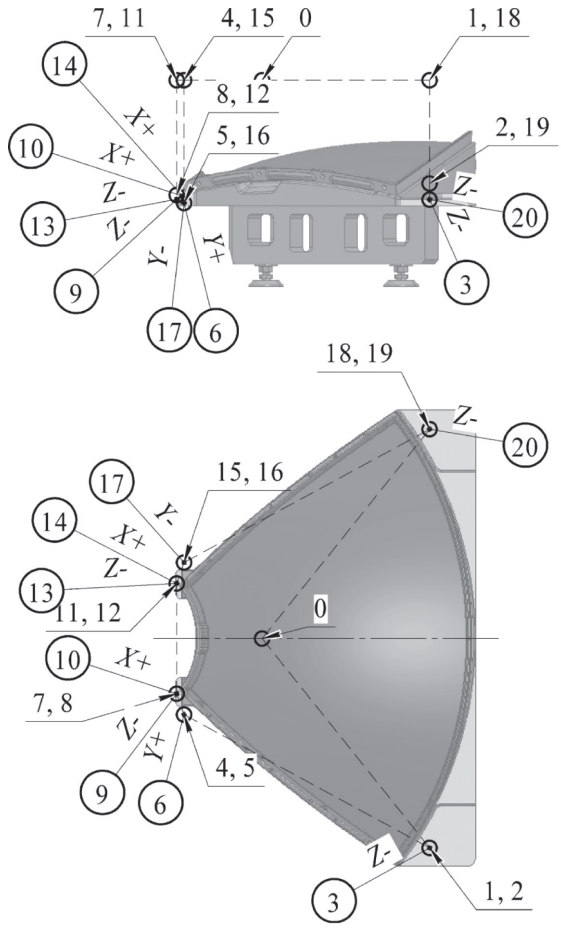


Рисунок 5. Точки, используемые при измерении первого приспособления

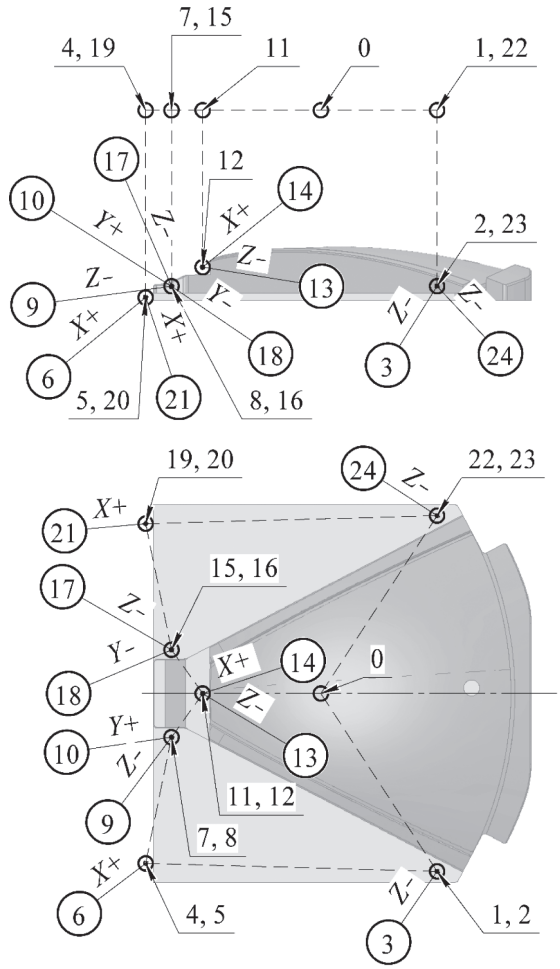


Рисунок 6. Точки, используемые при измерении второго приспособления

Полный алгоритм работы программы представлен в Таблице 1.

Таблица 1

Алгоритм работы программы базирования

№	Действие	
	Для приспособления 1 (см. Рисунок 3)	**Для приспособления 2 (см. Рисунок 4)**
1	Выполнение измерений точек на базовых поверхностях в следующей последовательности:	
	по схеме на Рисунке 5: 0 → 1 → 2 → 3 (измерение по Z–) → 1 → 4 → 5 → 6 (измерение по Y+) → 4 → 7 → 8 → 9 (измерение по Z–) → 10 (измерение по X+) → 7 → 11 → 12 → 13 (измерение по Z–) → 14 (измерение по X+) → 11 → 15 → 16 → 17 (измерение по Y–) → 15 → 18 → 19 → 20 (измерение по Z–) → 18 → 0	по схеме на Рисунке 6: 0 → 1 → 2 → 3 (измерение по Z–) → 1 → 4 → 5 → 6 (измерение по X+) → 4 → 7 → 8 → 9 (измерение по Z–) → 10 (измерение по Y+) → 7 → 11 → 12 → 13 (измерение по Z–) → 14 (измерение по X+) → 11 → 15 → 16 → 17 (измерение по Z–) → 18 (измерение по Y–) → 15 → 19 → 20 → 21 (измерение по X+) → 19 → 22 → 23 → 24 (измерение по Z–) → 22 → 0

Окончание таблицы 1

2	Вычисление смещений Δx, Δy, Δz приспособления WCS:	
	2.1. Вычисление коэффициентов уравнения для базовой плоскости *Oxy* WCS,	
	проходящей через измеренные точки 3, 9, 13, 20	проходящей через точку 13 и параллельную плоскости, проходящей через точки 3, 6, 16 и 20
	2.2. Вычисление коэффициентов уравнения плоскости *Oyz* WCS, перпендикулярной *Oxy* WCS:	
	проходящей через измеренные точки 10 и 14	проходящей через точку 14 и параллельную плоскости, которая также перпендикулярна *Oxy* WCS и проходит через точки 10 и 13
	2.3. Вычисление коэффициентов уравнения плоскости *Oxz* WCS, перпендикулярной линии пересечения плоскостей *Oxy* WCS и *Oyz* WCS	
	и проходящей через середину отрезка между измеренными точками 6 и 17	и проходящей через середину отрезка между измеренными точками 10 и 18
	2.4. Определение точки пересечения плоскостей *Oxy*, *Oyz*, *Oxz* WCS в пространстве, то есть значений смещения WCS Δx, Δy, Δz	
	2.5. Составление уравнений координатных плоскостей для ICS	
3	Вычисление углов θ_A, θ_B, θ_C ориентации WCS в пространстве:	
	3.1. Вычисление угла θ_A ориентации WCS как угла между осью *Ox* ICS и проекцией оси *Ox* WCS на плоскость *Oxy* ICS	
	3.2. Вычисление угла θ_B ориентации WCS как угла между осью *Ox* WCS и ее проекцией на плоскость *Oxy* ICS	
	3.3. Вычисление угла θ_C ориентации WCS как угла между осью *Oy* WCS и осью *Oy* ICS (*A*)	
4	Присвоение вычисленных значений смещений и углов ориентации соответствующим координатам выбранной базы робота	

Источник: таблица составлена авторами.

Для реализации математического аппарата вычисления координатных плоскостей применялись следующие математические выкладки:

• общее уравнение плоскости с вектором нормали $\vec{n}=(A,B,C)$:

$$Ax+By+Cz+D=0; \quad (1)$$

• нормальное уравнение плоскости с единичным вектором нормали $\vec{n}=(\cos\alpha,\cos\beta,\cos\gamma)$:

$$\cos\alpha\cdot x+\cos\beta\cdot y+\cos\gamma\cdot z-p=0; \quad (2)$$

• уравнение плоскости через три точки $M_1(x_1,y_1,z_1)$, $M_2(x_2,y_2,z_2)$ и $M_3(x_3,y_3,z_3)$, не лежащие на одной прямой:

$$\begin{vmatrix} x-x_1 & y-y_1 & z-z_1 \\ x_2-x_1 & y_2-y_1 & z_2-z_1 \\ x_3-x_1 & y_3-y_1 & z_3-z_1 \end{vmatrix}=0; \quad (3)$$

• каноническое уравнение прямой в пространстве через точку $M_0(x_0,y_0,z_0)$ с направляющим вектором $\vec{a}=(a_x,a_y,a_z)$.

$$\frac{x-x_0}{a_x}=\frac{y-y_0}{a_y}=\frac{z-z_0}{a_z}=\lambda; \tag{4}$$

• параметрический вид уравнения прямой:

$$\begin{cases} x=x_0+a_x\cdot\lambda, \\ y=y_0+a_y\cdot\lambda, \\ z=z_0+a_z\cdot\lambda; \end{cases} \tag{5}$$

• векторное произведение двух векторов:

$$\vec{c}=\vec{a}\times\vec{b}=\begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ A_a & B_a & C_a \\ A_b & B_b & C_b \end{vmatrix}; \tag{6}$$

• точка пересечения трех плоскостей:

$$\begin{cases} A_1\cdot x+B_1\cdot y+C_1\cdot z+D_1=0, \\ A_2\cdot x+B_2\cdot y+C_2\cdot z+D_2=0, \\ A_3\cdot x+B_3\cdot y+C_3\cdot z+D_3=0; \end{cases} \tag{7}$$

• скалярное произведение двух векторов:

$$\vec{a}\cdot\vec{b}=|\vec{a}|\cdot|\vec{b}|\cdot\cos\theta_{ab}; \tag{8}$$

• скалярное произведение двух векторов в координатной форме:

$$\vec{a}\cdot\vec{b}=a_x\cdot b_x+a_y\cdot b_y+a_z\cdot b_z. \tag{9}$$

Ниже подробно описаны этапы, представленные в Таблице 1.

В соответствии с п. 2.1 для первого приспособления коэффициенты $A_{xyR}, B_{xyR}, C_{xyR}, D_{xyR}$ плоскости *Oxy* WCS вычисляются на основании совместного решения уравнений (1) и (3). Так как для вычисления коэффициентов уравнения плоскости требуется только три точки, то используются измеренные точки *3*, *20* и середина отрезка между точками *9* и *13* $M_{913}\left(x_{913}, y_{913}, z_{913}\right)$:

$$x_{913}=\frac{x_9+x_{13}}{2},\ y_{913}=\frac{y_9+y_{13}}{2},\ z_{913}=\frac{z_9+z_{13}}{2}.$$

В соответствии с п. 2.1 для второго приспособления коэффициенты плоскости, проходящей через измеренные точки, как и для первого приспособления, вычисляются для плоскости, проходящей через точки *3*, *24* и середину отрезка между точками *9* и *17* $M_{917}\left(x_{917}, y_{917}, z_{917}\right)$:

$$x_{917}=\frac{x_9+x_{17}}{2},\ y_{917}=\frac{y_9+y_{17}}{2},\ z_{917}=\frac{z_9+z_{17}}{2}.$$

Так как у параллельных плоскостей координаты векторов нормалей будут одинаковыми, то задача сводится к определению коэффициента D_{xyR}, который находится из уравнения (1) при известных координатах точки, через которую проходит плоскость (то есть через измеренную точку *13*):

$$D_{xyR}=-A_{xyR}\cdot x_{13}-B_{xyR}\cdot y_{13}-C_{xyR}\cdot z_{13}.$$

По п. 2.2 для первого приспособления вычисление коэффициентов A_{yzR}, B_{yzR}, C_{yzR}, D_{yzR} плоскости *Oyz* проводится по измеренным точкам *10*, *14* и точке, которая лежит на прямой, перпендикулярной плоскости *Oxy* WCS, проходящей через середину отрезка между точками *10* и *14*.

Середина отрезка между точками *10* и *14* – точка $M_{1014}\left(x_{1014}, y_{1014}, z_{1014}\right)$:

$$x_{1014}=\frac{x_{10}+x_{14}}{2},\ y_{1014}=\frac{y_{10}+y_{14}}{2},\ z_{1014}=\frac{z_{10}+z_{14}}{2}.$$

Точка, лежащая на прямой, перпендикулярной плоскости *Oxy* WCS, может быть найдена с помощью координат нормального вектора данной плоскости:

$$x_{1014nxyR}=x_{1014}-A_{xyR},\ y_{1014nxyR}=y_{1014}-B_{xyR},\ z_{1014nxyR}=z_{1014}-C_{xyR}.$$

После чего производится вычисление коэффициентов плоскости аналогично ранее описанным способом.

По п. 2.2 для второго приспособления вычисляются коэффициенты A_{yzR}, B_{yzR}, C_{yzR} плоскости *Oyz* по измеренным точкам *6*, *21* и точке, которая лежит на прямой, перпендикулярной плоскости *Oxy* WCS, проходящей через середину отрезка между точками *6* и *21* (аналогично п. 2.1).

Вычисляется коэффициент D_{yzR} плоскости *Oyz*, проходящей через измеренную точку *14*:

$$D_{yzR}=-A_{yzR}\cdot x_{14}-B_{yzR}\cdot y_{14}-C_{yzR}\cdot z_{14}.$$

В соответствии с п 2.3 сначала необходимо определить координаты середины отрезка $M_{617}\left(x_{617}, y_{617}, z_{617}\right)$ между измеренными точками *6* и *17* для первого приспособления и точками *10, 18* $M_{1018}\left(x_{1018}, y_{1018}, z_{1018}\right)$ – для второго приспособления:

$$x_{617}=\frac{x_6+x_{17}}{2},\ y_{617}=\frac{y_6+y_{17}}{2},\ z_{617}=\frac{z_6+z_{17}}{2};$$

$$x_{1018}=\frac{x_{10}+x_{18}}{2},\ y_{1018}=\frac{y_{10}+y_{18}}{2},\ z_{1018}=\frac{z_{10}+z_{18}}{2}.$$

Далее вычисляются координаты направляющего вектора прямой пересечения плоскостей *Oxy* и *Oyz* по формуле (6). Данный вектор будет являться вектором нормали $\vec{n}_{xzR}$ для плоскости *Oxz*, а полученные множители перед ортами $\vec{i}$, $\vec{j}$, $\vec{k}$ – его координатами (они же коэффициенты A_{xzR}, B_{xzR}, C_{xzR}) плоскости *Oxz*.

Для определения коэффициента D_{xzR} необходимо решить уравнение (1) относительно D_{xzR} с известными координатами точки M_{617} и M_{1018} для первого и второго приспособлений соответственно:

$$D_{xzR}=-A_{xzR}\cdot x_{617}-B_{xzR}\cdot y_{617}-C_{xzR}\cdot z_{617};$$

$$D_{xzR}=-A_{xzR}\cdot x_{1018}-B_{xzR}\cdot y_{1018}-C_{xzR}\cdot z_{1018}.$$

По п. 2.4 для вычисления положения начала WCS координат (точки $O\left(x_{0R}, y_{0R}, z_{0R}\right)$) необходимо решить систему из трех уравнений (7) пересекающихся плоскостей *Oxy*, *Oxz*, *Oyz* методом Крамера.

По п. 2.5 уравнения плоскостей *Oxy*, *Oxz*, *Oyz* для ICS являются частным случаем уравнения (1), так как две из трех координат вектора нормали каждой из плоскостей равны нулю, а оставшуюся можно принять равной единице. Получим:

для *Ox*

$$z-z_{0R}=0;$$

для *Oxz*

$$y-y_{0R}=0;$$

для *Oyz*

$$x - x_{0R} = 0.$$

По п. 3.1 вычисление угла между проекцией вектора $\vec{Ox}_R\left(A_{yzR}, B_{yzR}, C_{yzR}\right)$ на плоскость *Oxy* ICS и ее осью $\vec{Ox}$ производится совместным решением уравнений (8) и (9). При этом проекцию вектора на *Oxy* ICS можно записать как $\vec{Ox}_{Rxy}\left(A_{yzR}, B_{yzR}, 0\right)$, а координаты единичного вектора – как $\vec{Ox}(1,0,0)$. Тогда угол θ_A можно принять

$$\cos\theta_A = \frac{A_{yzR}}{\sqrt{{A_{yzR}}^2 + {B_{yzR}}^2}}.$$

По п. 3.2 для вычисления угла поворота θ_B СК вокруг оси *Oy* необходимо, как и в п. 3.1, решить систему уравнений (8) и (9) для векторов $\vec{Ox}_R\left(A_{yzR}, B_{yzR}, C_{yzR}\right)$ и $\vec{Ox}_{Rxy}\left(A_{yzR}, B_{yzR}, 0\right)$:

$$\cos\theta_B = \frac{{A_{yzR}}^2 + {B_{yzR}}^2}{\sqrt{{A_{yzR}}^2 + {B_{yzR}}^2 + {C_{yzR}}^2} \cdot \sqrt{{A_{yzR}}^2 + {B_{yzR}}^2}}.$$

По п. 3.3 координаты единичного вектора оси *Oy* ICS (A) можно записать как $\vec{Oy}_A\left(-\sin\theta_A, \cos\theta_A, 0\right)$, соответственно, угол определяется по формуле

$$\cos\theta_C = \frac{-A_{xzR} \cdot \sin\theta_A + B_{xzR} \cdot \cos\theta_A}{\sqrt{{A_{xzR}}^2 + {B_{xzR}}^2 + {C_{xzR}}^2}}.$$

Описанный алгоритм в Таблице 1 и математические выкладки были реализованы для робота KUKA KR30 R3100 в коде программы на языке Kuka Robot Language (KRL) под систему KUKA System Software 8.2.

Программная реализация

Программа измерения разбита на несколько частей.

Головная часть программы состоит из двух файлов (Smart_Base.SRC и Smart_Base.DAT). Файл *Smart_Base.SRC* содержит текст программы, подпрограмм и функций, в которых запрограммированы выбор измеряемой технологической оснастки и номера базы для сохранения вычисляемых координат и углов ориентации, последовательность перемещений по точкам измерения базовых точек, определение координат точек и вычисления искомых величин. Перемещения от точки к точке на безопасной высоте выполняются с использованием команды PTP. При вертикальном движении используется команда линейных перемещений LIN, замеры точек выполняются с помощью подпрограммы *Measure Point()*.

Файл *Smart_Base.DAT* содержит объявление используемых переменных, значения координат точек перемещения и измеренных точек.

Вспомогательная часть также состоит из двух файлов (Geom_Proc_Func.SRC и Geom_Proc_Func.DAT) и содержит необходимые процедуры и функции для вычисления коэффициентов плоскостей, прямых, векторов и др. То есть в этой части реализован общий математический аппарат из области геометрии плоскостей и векторов, который должен быть доступен и для других программ.

Одна из основных сущностей, используемых для вычисления положения и ориентации рабочей системы координат, – плоскость, которая может быть описана четырьмя коэффициентами *A*, *B*, *C*, *D* (см. уравнение (1)). Для этого был создан глобальный тип переменной PLABCD_TYPE:

GLOBAL STRUC PLABCD_TYPE REAL A, B, C, D

Для промежуточных вычислений используются координатные плоскости WCS:

DECL PLABCD_TYPE PLN_XY, PLN_XZ, PLN_YZ

В главной программе *Smart_base.src* можно выделить следующие характерные разделы программы.

1. Инициализация начальных параметров. Выбор инструмента (щупа) и базы, присвоение значений переменным, описывающим параметры движения, расстояние отвода щупа после касания поверхности.

2. Вызов диалоговых окон для выбора технологической оснастки, подлежащей базированию:

DetalNum = ShowDialogDetal()

и номер базы, в которую требуется сохранить вычисленные значения:

baseNum = ShowDialogBase()

3. Количество и размещение измеряемых точек для двух деталей различно, соответственно, управляющие программы для их измерения также различаются, поэтому подпрограмма измерения вызывается с входным параметром DetalNum. При выполнении функции значения измеренных точек сохраняются в массиве MP[], при этом индексы массива соответствуют номерам точек на Рисунках 5, 6:

PART_MEAS_POINTS(DetalNum)

4. В зависимости от номера технологической оснастки выполняется функция Find_Planes() для определения трех координатных плоскостей PLN_XY, PLN_XZ, PLN_YZ, а для оснастки Рисунка 6 смещение координатных плоскостей XY и YZ в точки 13 и 14 соответственно происходит с помощью функции OFFSETPL1PNT(). По завершении вычисляются углы ориентации рабочей системы координат с помощью подпрограммы FIND_ANGLES().

5. Точка пересечения трех координатных плоскостей определяет начальную точку рабочей системы координат и определяется с помощью подпрограммы Find_StartPNT(). Найденное значение положения и ориентации рабочей системы координат сохраняется в переменную resultPos, а затем соответствующие поля ранее выбранной базы.

В качестве входных параметров в подпрограмме Find_Planes() используются восемь точек:

P1PLN, P2PLN, P3PLN, P4PLN – четыре точки на одной плоскости, которые в дальнейшем приводятся к трем точкам установочной плоскости (между точками 2 и 3 вычисляется средняя точка P23PLN);

P1LIN, P2LIN – две точки образующие прямую (направляющая база);

P1PNT, P2PNT – две точки, середина между которыми является опорной базой.

Для базирования и настройки WCS оператору достаточно выполнить следующую последовательность действий:

1) установить приспособление на стол РТК в его штатную позицию и закрепить. Погрешность установки приспособления допускается в районе ±5 мм;

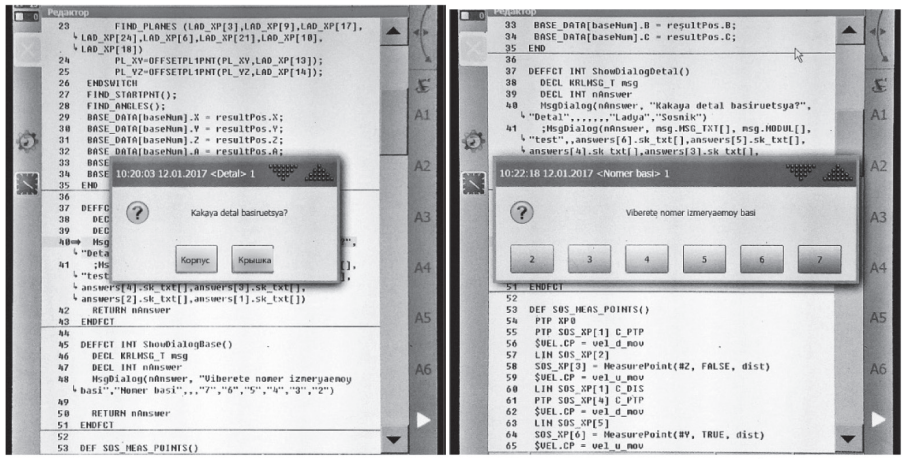
2) установить в шпиндель робота измерительный модуль *Renishaw OMP-40*;

3) с помощью графического интерфейса системы управления роботом выбрать программу автоматического базирования *Smart_Base.SRC* и выполнить:

a. в первом появившемся диалоговом окне (см. Рисунок 7, *а*) выбрать измеряемую деталь;

b. во втором диалоговом окне (см. Рисунок 7, *б*) выбрать номер базы, в которую будут сохранены вычисленные значения положения WCS и ее ориентации в пространстве;

4) проверить вычисленные и присвоенные значения в соответствующих переменных системы управления роботом.



а *б*

Рисунок 7. Диалоговые окна при выполнении программы автоматической настройки WCS

Если измерения выполнены успешно, автоматически происходит вычисление необходимых значений и их запись в выбранную ранее базу.

Представленная управляющая программа была разработана только для двух изделий и подразумевает установку приспособлений в одно и то же место относительно базы робота (погрешность ± 5 мм). Для повышения удобства использования предложенного алгоритма более целесообразным является, во-первых, перемещение всех точек отсчитывать от начального положения щупа (оператор до запуска программы базирования подводит щуп к характерному конструктивному элементу на детали или приспособлении и затем запускает программу базирования), во-вторых, для подготовки программы базирования использовать CAM-среду, в которой технологу-программисту будет комфортно с использованием графического интерфейса CAM-среды подготовить траекторию перемещения щупа по всем точкам измерения и с помощью постпроцессора выполнить вывод программы. Чтобы реализовать данные возможности, необходимо разработать постпроцессор, который будет выводить в нужном формате управляющую программу для базирования.

Рекомендации по разработке постпроцессора

Современные CAM-среды имеют широкие возможности по разработке управляющих программ механической обработки, однако они не всегда имеют необходимые инструменты для программирования измерений с помощью щупа. В данной статье предлагается способ для обхода этих ограничений.

Как правило, все программируемые в CAM-среде перемещения могут быть классифицированы следующим образом (см. Рисунок 8):

1 – движение с ускоренной подачи (от точки *1* к точке *2*) – выполняется на максимальных скоростях, на высотах, в которых исключены столкновения с заготовкой и крепежными элементами;

2 – движение врезания (от точки *2* до точки *3*) – инструмент врезается в материал и подходит к началу рабочего сегмента траектории; подачу для данного движения можно задать отличную от рабочей подачи;

3 – движение рабочей подачи, в процессе которого подразумевается съем материала с заготовки (от точки *3* к точке *4*);

4 – ускоренный отвод инструмента до безопасной высоты (от точки *4* к точке *5*).

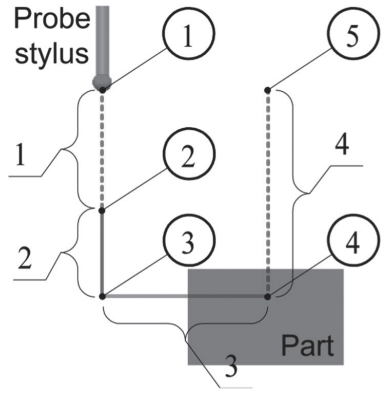


Рисунок 8. Движения инструмента по траектории

При программировании процесса измерения движения 1 и 4 можно оставить без изменений, то есть использовать простые перемещения (LIN или PTP) без контроля столкновений. Перемещение 2 следует выполнять на пониженной скорости и включить обработку прерываний, так как если произойдет срабатывание датчика касания на данном участке, то робот выполнит немедленную остановку, выполнение программы будет приостановлено, и данная ситуация будет считаться внештатной. Перемещение 3 – измерение точки на поверхности. Перед началом перемещения включается обработка прерываний по датчику касаний, и перемещение осуществляется либо до касания щупом измеряемой поверхности, либо до конечной запрограммированной точки; однако если прерывание не сработает, то поверхность не будет измерена, выполнение программы будет приостановлено.

Точки, которые будут измерены на роботе, в CAM-среде задаются как лежащие внутри детали (точка *4* на Рисунке 8), то есть робот не будет достигать координат данной точки. При этом координаты точки отвода (точка *5* на Рисунке 8) необходимо скорректировать, чтобы избежать внештатных столкновений щупа с деталью. Такое изменение координат запрограммированных в CAM-среде точек приводит к необходимости организовать сохранность первоначальных значений в управляющей программе.

Для возможности коррекции координат точек в управляющей программе измеряемые точки и точки траекторий необходимо выводить в виде массива типа FRAME, а для обеспечения сохранности создавать дублирующий массив точек, в котором будут происхо-

дить изменения значений координат и в начале каждого запуска программы восстанавливаться первоначальные значения, сгенерированные в CAM-среде.

Общие рекомендации по настройке отдельных блоков в постпроцессоре выглядят следующим образом.

1. Блок Program Start или Program Begin. Обычно располагается в начале программы, и в нем необходимо отключить опережающий просмотр управляющей программы и ввести номер базы baseNum, которая будет измеряться. При этом можно использовать переменную, значение которой присваивается в CAM-среде либо вводится вручную уже в готовой программе:

```
$ADVANCE = 0
baseNum = 10
```

Также вначале необходимо использовать временную базу, которую следует расположить в точке с текущим положением щупа:

```
BASE_DATA[31].X = $POS_ACT.X
BASE_DATA[31].Y = $POS_ACT.Y
BASE_DATA[31].Z = $POS_ACT.Z
BASE_DATA[31] = {A 0, B 0, C 0}
BAS(#BASE,31)
```

Для обеспечения сохранности первоначальных значений в «рабочий массив точек» записываются значения точек, полученных из CAM-среды:

```
FOR I = 1 TO 48
PNT[I] = CAM_PNT[I]
ENDFOR
```

Так как фактическая ориентация щупа может отличаться от запрограммированной, робот, выполняя переориентацию при перемещении в первую точку, может аварийно столкнуться с деталью или оснасткой. Чтобы этого избежать, следует отвести щуп на безопасное расстояние от детали, выполнить разворот и затем вернуться обратно к исходной точке. В качестве безопасного расстояния принимается высота 100 мм от текущей позиции вдоль оси щупа. При этом сначала осуществляется проверка – действительно ли текущая ориентация по трем составляющим отличается от заданной в первой точке:

```
IF ($POS_ACT.A<>PNT[1].A)OR($POS_ACT.B<>PNT[1].B) OR($POS_ACT.C<>PNT[1].C)) THEN
```

Тогда будет выполнен разворот щупа в соответствии с запрограммированными значениями с помощью четырех точек.

Первая точка – текущая позиция и ориентация щупа:

```
TMP_P1 = {X 0, Y 0, Z 0}
TMP_P1.A = $POS_ACT.A
TMP_P1.B = $POS_ACT.B
TMP_P1.C = $POS_ACT.C
```

Координаты второй точки получены с помощью геометрического оператора, который добавляет смещение относительно точки TMP_P1 (векторная сумма):

```
TMP_P2 = TMP_P1:{Z 100}
```

В третьей точке ориентация щупа приводится к запрограммированной:

```
TMP_P3 = TMP_P2
TMP_P3.A = PNT[1].A
```

```
TMP_P3.B = PNT[1].B
TMP_P3.C = PNT[1].C
```

Четвертая точка возвращает щуп в исходную позицию, но с новой ориентацией:

```
TMP_P4 = TMP_P1
TMP_P4.A = PNT[1].A
TMP_P4.B = PNT[1].B
TMP_P4.C = PNT[1].C
```

Далее осуществляется последовательность перемещений по ранее заданным точкам:

```
PTP TMP_P2
PTP TMP_P3
PTP TMP_P4
ENDIF
```

2. Блок Toolpath start или Toolpath Begin. Каждая траектория, созданная в CAM-среде и добавленная в управляющую программу для постпроцессирования, начинается с данного блока. В качестве начальной точки используется текущее положение щупа, следующая точка соответствует первой точке траектории. Так как щуп уже находится в начальной точке траектории, реального перемещения робота не происходит:

```
PTP $AXIS_ACT
PTP PNT[1] C_PTP
```

3. Блок Rapid move или Joint move. Перемещения, которые соответствуют участкам *1* и *4* Рисунка 8. При переходе от точки к точке возникает необходимость изменять индекс точки. Возможны два варианта изменения индекса:

• если постпроцессор содержит такой инструмент, как счетчик (<C1>), фрагмент данного блока будет выглядеть следующим образом:

```
LIN PNT[<C1>]
```

• в текст управляющей программы вместо <C1> при каждом обращении к блоку будет выводиться последовательно увеличивающееся значение (1, 2, 3 и т. д.);

• если постпроцессор не содержит счетчиков, его можно организовать внутри управляющей программы:

```
C1 = C1 + 1
LIN PNT[C1]
```

4. Блок Plunge move. Перемещения, которые соответствуют участку *2* Рисунка 8. В начале перемещения необходимо включить обработку прерываний, по завершении – выключить. Данная операция выполняется с помощью подпрограммы SafetyMoving():

```
SafetyMoving(true)
LIN PNT[<C1>]
SafetyMoving(false)
```

5. Блок Cutting move или Feedrate move. Перемещения, которые соответствуют участку *3* Рисунка 8 (из точки *3* в точку *4*). Внутри функции MeasurePNT() реализуется перемещение в точку *4* с включенным обработчиком прерываний по датчику касания. В момент срабатывания датчика робот останавливается, а достигнутые координаты с учетом радиуса сферы наконечника принимаются за точку измеряемой поверхности и возвращаются в качестве результата работы функции MeasurePNT(). Данное значение записывается в массив MP[]. В подпрограмме Offset_PNT() происходит изменение координат точки 5 (см. Рисунок 8) на величину разницы между текущим положением щупа (то есть

после измерения и отвода на безопасное расстояние) и положением точки *4* (недостижимой точки внутри детали).

Для изменения индексов в массиве MP[] используется счетчик <C2>в постпроцессоре:

```
MP[<C2>]=MeasurePNT(PNT[<C1>])
OFFSET_PNT($POS_ACT, PNT[<C1>],PNT[<C1>+1])
```

либо счетчик в виде переменной C2 в управляющей программе:

```
C2=C2+1
C1=C1+1
MP[C2]=MeasurePNT(PNT[C1])
OFFSET_PNT($POS_ACT, PNT[C1],PNT[C1+1])
```

После измерения будет блок ускоренных перемещений, который описан в п. 3. Для каждой последующей измеряемой точки блоки, описанные в пп. 3–5, будут повторяться.

6. Блок Program End или Program Finish. Содержит завершающие строки управляющей программы. В данный блок необходимо добавить подпрограммы для вычисления координатных плоскостей *XY*, *YZ*, *XZ*, а затем вычисления точки их пересечения и углов ориентации.

Так как данный блок является индивидуальным для каждой детали, его целесообразно вводить вручную уже в готовой управляющей программе.

Также в этот блок нужно добавить описанные ранее функции для вычисления углов ориентации и точки начала рабочей системы координат:

```
FIND_ANGLES()
FIND_STARTPNT()
```

Так как все вычисления производились относительно временной системы координат, то значения начальной точки измеряемой системы координат необходимо скорректировать:

```
BASE_DATA[baseNum] = resultPos
BASE_DATA[baseNum].X = BASE_DATA[31].X + resultPos.X;
BASE_DATA[baseNum].Y = BASE_DATA[31].Y + resultPos.Y;
BASE_DATA[baseNum].Z = BASE_DATA[31].Z + resultPos.Z;
```

Также в данный блок можно добавить подпрограммы и функции, используемые в управляющей программе.

В Таблице 2 представлены подпрограммы для вычисления коэффициентов *A*, *B*, *C*, *D* в уравнениях плоскостей и координат точек.

Таблица 2

Доступные подпрограммы для промежуточных вычислений

Вызов функции и входные параметры	Выходной параметр
PLN_Through3PNT(PNT1,PNT2,PNT3,PLNX) PNT1, PNT2, PNT3 – три точки, не лежащие на одной прямой (тип данных FRAME)	PLNX – плоскость, проходящая через три точки (тип PLABCD_TYPE)
PLN_Perp2PLN_ Through1PNT(PL1,PL2,PNT1,PLNX) PLN1, PLN2 – две перпендикулярные плоскости (тип PLABCD_TYPE)	PLNX – плоскость, проходящая через точку PNT1, перпендикулярно плоскостям PLN1 и PLN2 (тип PLABCD_TYPE)

Окончание таблицы 2

PLN_Perp1PLN_Through2PNT(PLN1,PNT1,PNT2,PLNX) PLN1 – плоскость (тип PLABCD_TYPE); PNT1 и PNT2 – точки в пространстве, прямая через которые не перпендикулярна плоскости PLN1 (тип данных FRAME)	PLNX – плоскость, перпендикулярная плоскости PLN1 и проходящая через точки PNT1 и PNT2 (тип PLABCD_TYPE)
PLN_Par1PLN_Through1PNT(PLN1,PNT1,PLNX) PLN1 – плоскость (тип PLABCD_TYPE); PNT1 – точки в пространстве, не лежащие на плоскости PLN1 (тип данных FRAME)	PLNX – плоскость, параллельная плоскости PLN1 и проходящая через точки PNT1 (тип PLABCD_TYPE)
PNT_Mid2PNT(PNT1,PNT2,PNTX) PNT1, PNT2 – две несовпадающие точки в пространстве (тип данных FRAME)	PNTX – середина между двумя точками (тип FRAME)
PNT_Hole4PNT(PNT1X,PNT2X,PNT1Y,PNT2Y,PNTX) PNT1X и PNT2X – точки, лежащие на диаметральных сторонах отверстия вдоль оси *X* (тип данных FRAME); PNT1Y и PNT2Y – точки, лежащие на диаметральных сторонах отверстия вдоль оси *Y* (тип данных FRAME)	PNTX – центр отверстия, измеренного по четырем точкам (тип FRAME)

Источник: таблица составлена авторами.

Заключение

Программы автоматического базирования деталей для роботов-манипуляторов KUKA, основанные на представленных в данной статье математических выкладках и алгоритмах, позволяют сократить подготовительное время на переналадку технологической оснастки в среднем до 20–25 минут (установка и закрепление оснастки – 15–20 минут, выполнение программы – 5 минут). Таким образом, производительность данной операции увеличилась более чем в 7 раз.

Подготовка специального постпроцессора для CAM-среды и соответствующих руководящих материалов для технолога-программиста позволяет на одном рабочем месте создавать как программы фрезерной обработки, так и программы автоматического базирования вне зависимости от возможностей CAM-пакета, что в конечном итоге в разы сокращает время на технологическую подготовку и настройку оборудования.

Литература / References

1. Schleth G., Kuss A., Kraus W. (2018) Workpiece localization methods for robotic welding – A review. In: *ISR 2018; 50th International Symposium on Robotics*, Munich, Germany, 20–21 June 2018. Pp. 1–6. URL: https://ieeexplore.ieee.org/document/8470567 (accessed 17.01.2024).

2. Qin H., Li Y., Xiong X. (2019) Workpiece pose optimization for milling with flexible joint robots to improve quasi-static performance. *Applied Sciences*. No. 9. Article ID 1044. DOI: https://doi.org/10.3390/app9061044

3. Du Y., Liang Z., Chen S., Huang H., Zheng H., Gao Z., Zhou T., Liu Z., Wang X. (2023) Dynamic modeling and stability prediction of robot milling considering the influence of force-induced deformation on regenerative effect and process damping. *Metals.* Vol. 13. No. 5. Article ID 974. DOI: https://doi.org/10.3390/met13050974

4. Caro S., Dumas C., Garnier S., Furet B. (2013) Workpiece placement optimization for machining operations with a KUKA KR270-2 robot. In: *2013 IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, 06–10 May 2013. Pp. 2921–2926. DOI: https://doi.org/10.1109/ICRA.2013.6630982

5. Qin H., Li Y., Xiong X. (2019) Workpiece pose optimization for milling with flexible joint robots to improve quasi-static performance. *Applied Sciences*. No. 9. Article ID 1044. DOI: https://doi.org/10.3390/app9061044

6. Balci B., Donovan J., Roberts J., Corke P. (2023) Optimal workpiece placement based on robot reach, manipulability and joint torques. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*, London, United Kingdom, 29 May – 02 June 2023. Pp. 12302–12308. DOI: https://doi.org/10.1109/ICRA48891.2023.10161031

7. Liao Z.-Y., Wang Q.-H., Xie H.-L., Li J.-R., Zhou X.-F., Pan T.-H. (2022) Optimization of robot posture and workpiece setup in robotic milling with stiffness threshold. In: *IEEE/ASME Transactions on Mechatronics*. Vol. 27. No. 1. Pp. 582–593. DOI: https://doi.org/10.1109/TMECH.2021.3068599

8. Liao Z., Li J.-R., Xie H.-L., Wang Q.-H., Zhou X. (2020) Region-based toolpath generation for robotic milling of freeform surfaces with stiffness optimization. *Robotics and Computer-Integrated Manufacturing*. Vol. 64. Article ID 101953. DOI: https://doi.org/10.1016/j.rcim.2020.101953

9. Lin Y., Zhao H., Ding H. (2017) Posture optimization methodology of 6R industrial robots for machining using performance evaluation indexes. *Robotics and Computer-Integrated Manufacturing*. Vol. 48. Pp. 59–72. DOI: https://doi.org/10.1016/j.rcim.2017.02.002

10. Gotlih J., Brezocnik M., Karner T. (2021) Stiffness-based cell setup optimization for robotic deburring with a rotary table. *Applied Sciences*. Vol. 11. Article ID 8213. DOI: https://doi.org/10.3390/app11178213

11. Xiong G., Ding Y., Zhu L. (2019) Stiffness-based pose optimization of an industrial robot for five-axis milling. *Robotics and Computer-Integrated Manufacturing*. Vol. 55. Pp. 19–28. DOI: https://doi.org/10.1016/j.rcim.2018.07.001

12. Zargarbashi S.H.H., Khan W., Angeles J. (2012) Posture optimization in robot assisted machining operations. *Mechanism and Machine Theory*. Vol. 51. Pp. 74–86 22https://doi.org/10.1016/j.mechmachtheory.2011.11.017

13. Xue Y., Sun Z., Liu S., Gao D., Xu Z. (2022) Stiffness-oriented placement optimization of machining robots for large component flexible manufacturing system. *Machines*. Vol. 10. No. 5. Article ID 389. DOI: https://doi.org/10.3390/machines10050389

14. Qu X., Wan M., Shen C.-J., Zhang W.-H. (2023) Profile error-oriented optimization of the feed direction and posture of the end-effector in robotic free-form milling. *Robotics and Computer-Integrated Manufacturing*. Vol. 83. Article ID 102580. DOI: https://doi.org/10.1016/j.rcim.2023.102580