

АДАПТАЦИЯ ПРОГРАММЫ MACH3 К ЗАДАЧАМ УПРАВЛЕНИЯ ПРОМЫШЛЕННЫМ РОБОТОМ RM-01

Романов В.В., Петровский В.С.

Костромской государственный университет, Кострома

Ключевые слова: кинематика, робот, программа, система управления, обратная задача кинематики, драйвер, компьютер.

Аннотация. Проведена модернизация управляющего контура промышленного робота RM-01 «Пума». Механическая часть с двигателями оставлена неизменной. Установлены новые драйверы приводов звеньев. Управление осуществляется компьютером IBM с использованием программы для станков с ЧПУ. Ввиду отсутствия в открытом доступе специализированных программ для управления роботами, такие программы требуют адаптации к угловым координатам робота. Для этого рассмотрены аналитические решения обратной задачи кинематики для определения обобщенных координат робота-манипулятора RM-01 «Пума». Разработана программа – транслятор для преобразования декартовых координат детали в угловые координаты робота.

ADAPTATION OF THE MACH3 PROGRAM TO THE PROBLEMS OF CONTROL OF THE INDUSTRIAL ROBOT RM-01

Romanov V.V., Petrovsky V.S.

Kostroma State University, Kostroma

Keywords: kinematics, robot, program, control system, inverse kinematics problem, driver, computer.

Abstract. Modernization of the control circuit of the RM-01 "Puma" industrial robot has been carried out. The mechanical part with the engines is left unchanged. New link drive drivers have been installed. The control is carried out by an IBM computer using a program for CNC machines. Due to the lack of publicly available specialized programs for controlling robots, such programs require adaptation to the angular coordinates of the robot. For this purpose, analytical solutions of the inverse kinematics problem for determining the generalized coordinates of the RM-01 "Puma" robot manipulator are considered. A translator program has been developed to convert the Cartesian coordinates of the part into the angular coordinates of the robot.

Анализ литературных данных [1, 2] показывает, что в учебном процессе разных стран активно используется модернизированные промышленные роботы типа Пума. При этом, сохраняя в целом кинематику, модернизации подвергается электронная начинка системы управления и его программное обеспечение. В КГУ в последние годы также ведутся работы в данном направлении. При этом основная цель работы не столько в модернизации имеющегося оборудования, сколько в получении студентами новых компетенций по эксплуатации и модернизации роботов в составе робототехнических комплексов. На первом этапе студенты изучают устройство робота, электронику приводов, драйверы двигателей, программное обеспечение и т.п., конечная цель составление управляющей программы для обхода заданного контура. На момент участия в проекте студенты обладают компетенциями составления управляющих программ для станков с ЧПУ, знакомы с устройством и работой станков. Определенной проблемой при решении поставленной задачи является программное обеспечение

роботов, которое, в отличие от станков с ЧПУ, в свободном доступе отсутствует. По аналогии с работой [3], в качестве базового программного обеспечения робота выбрана программа Mach3, формально рассчитанная на станки с ЧПУ. Однако, при наличии специальной программы – транслятора (или постпроцессора) удалось преобразовать линейные координаты потенциально обрабатываемых деталей в угловые координаты основных звеньев робота. В принципе наша задача свелась к программной реализации решения обратной задачи кинематики для робота с ангулярной системой координат. В литературе известны решения подобной задачи [4]. Однако математический аппарат этих решений оказался достаточно сложным для наших целей. Поэтому была реализована упрощенная модель, учитывающая управление не всеми шестью звеньями робота, а лишь четырьмя – основными. В статье представлен завершающий этап решения – учет модуля качания «пальца». В частном случае «пальцу» необходимо и достаточно быть просто перпендикулярным к базовой плоскости (рис. 1).

В решении используются обозначения:

$\alpha, \beta, \gamma, \delta$ – углы поворота соответственно звеньев 1, 2, 3 и 4 робота;

L – размер «плеча» и «локтя» робота;

P – параметр смещения плоскости качания звена 2 относительно оси симметрии колонны;

r – длина схвата или другого исполнительного инструмента.

Пусть задана целевая точка $M(x, y, z)$, обозначенная на рисунке 1 как (yz) . Тогда точка расположения 3-го звена будет находиться по оси Z на $(+r)$ выше целевой точки. А угол δ определится по формуле (1).

$$\delta = \pi - \beta - \gamma. \quad (1)$$

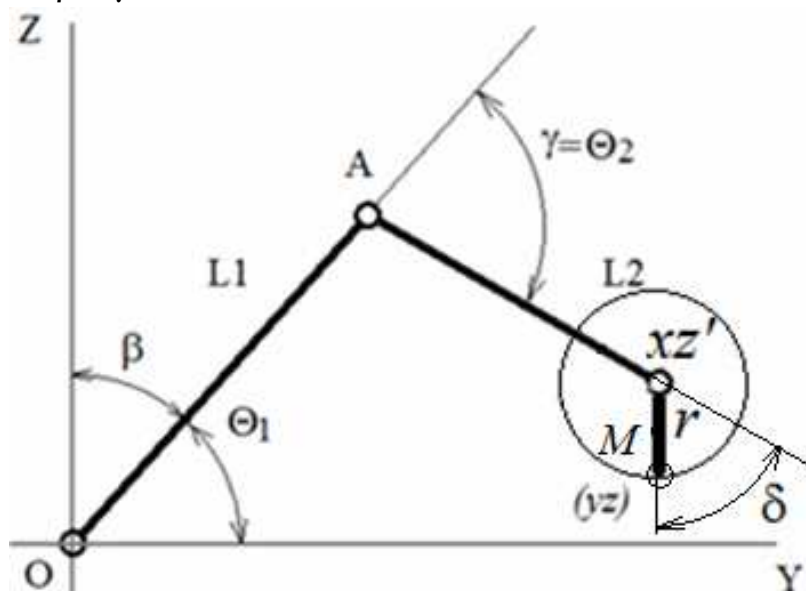


Рис. 1. Решение ОЗК для 3-хзвенного робота

Учитывая, что программа Mach3, применяемая для управления роботом, по сути является программой для станка с ЧПУ, углы поворота звеньев автоматически преобразуются в G-коды ISO: $\text{Alfa} \rightarrow X$, $\text{Beta} \rightarrow Y$, $\text{Gamma} \rightarrow Z$ и $\text{Del} \rightarrow A$. В связи с этим программа – постпроцессор для расчета опорных точек в угловых координатах манипулятора примет вид:

```

Const
  L =432;
  P =138;
  r=80;
Var
  x,y,z,xy,xy1,B:real;
  alf,bet,gam,del:real;
  n:byte;
  XYp:Array [1..10,1..2] of real;
BEGIN
  z:=0+r;
  Write('Количество точек');Readln(n);
  For var i:=1 to n do
    Begin
      Write('X = Y = ');
      Readln(XYp[i,1],XYp[i,2]);
    End;
    For var j:=1 to n Do
      Begin
        x:=XYp[j,1];y:=XYp[j,2];
        xy:=sqrt(sqr(x)+sqr(y));
        xy1:=xy*(cos(arcsin(p/xy)));
        B:=Sqrt(sqr(z)+sqr(xy1)) ;
        alf:=arccos(y/xy1)*180/Pi;
        bet:=(Pi/2-arccos(xy1/B)-arccos(B/2/L))*180/Pi;
        gam:=(Pi-arccos(1-sqr(b)/2/sqr(L)))*180/Pi;
        del:=180-bet-gam;
        Writeln('X',alf:3:2,'Y', Bet:3:2,'Z',Gam:3:2,'A',del:3:2);
      End;
    END.

```

Таким образом, появилась возможность трансформации задачи обхода линейного контура из кодов ISO 7-bit в угловые координаты опорных точек робота (рис. 2).

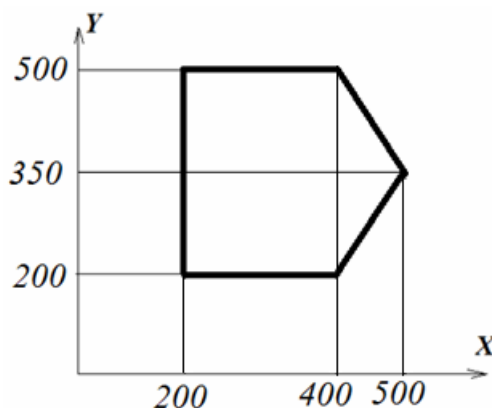


Рис. 2. Заданный линейный контур

Сравнение программ для станка с ЧПУ и робота для пяти опорных точек представлено в таблице 1.

Данная задача решена для линейного контура. При наличии в детали элементов контура, отличных от линейных, необходимо дополнить постпроцессор функцией аппроксимации [5]. Выбор метода аппроксимации дуги

окружности — секущими, касательными или хордами — определяется расположением поля допуска. Общим случаем является аппроксимация секущими, когда заданы верхнее (δ_B) и нижнее (δ_H) отклонения радиуса дуги окружности (рис. 3, а). При $\delta_H = 0$ аппроксимация дуги ведется касательными (рис. 3, б), а при $\delta_B = 0$ — хордами (рис. 3, в).

Табл. 1.

Программа ЧПУ	X	Y	Программа робота
N2 X200 Y200 Z0	200	200	X35.90 Y-0.47 Z145.04 A35.43
N3 Y500	200	500	X16.15 Y28.82 Z104.89 A46.29
N4 X400	400	500	X36.90 Y39.56 Z86.30 A54.14
N5 X500 Y350	500	350	X53.93 Y36.31 Z92.06 A51.64
N6 X400 Y200	400	200	X61.96 Y19.41 Z119.87 A40.72
N7 X200	200	200	X35.90 Y-0.47 Z145.04 A35.43

Центральный угол φ аппроксимируемой дуги окружности равен разности углов (α_K и α_0), образованных радиусами, проведенными в конечную $T_K (x_K, y_K)$ и начальную $T_0 (x_0, y_0)$ точки этой дуги:

$$\varphi = \alpha_K - \alpha_0 = \arccos \frac{x_K}{R} - \arccos \frac{x_0}{R}.$$

Шаг аппроксимации удобно выражать центральным углом $\Delta\varphi$, образованным радиусами, опирающимися на концы звеньев ломаной. При аппроксимации хордами угол $\Delta\varphi$ предварительно вычисляют по формуле

$$\Delta\varphi = 2 \arccos \frac{R - \delta_H}{R}.$$

Затем определяют число звеньев ломаной: $a = \varphi / \Delta\varphi$

Величину a округляют до большего целого числа, после чего уточняют шаг аппроксимации: $\Delta\varphi = \varphi / a$.

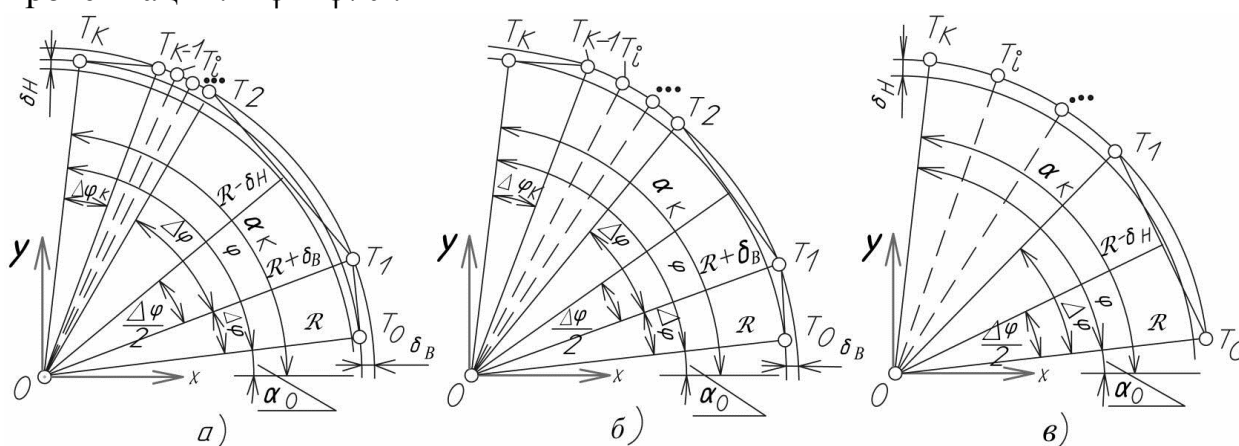


Рис. 3. Аппроксимация дуги окружности секущими (а), касательными (б) и хордами (в)

При аппроксимации секущими или касательными начальное и конечное звенья ломаной определяются углами $\Delta\varphi_0$ и $\Delta\varphi_K$:

$$\Delta\varphi_0 = \Delta\varphi_K = \arccos \frac{R}{R + \delta_B}.$$

Шаг остальных звеньев ломаной предварительно вычисляется по формулам:

– при аппроксимации секущими

$$\Delta\varphi = \arccos \frac{R - \delta_H}{R + \delta_B},$$

– при аппроксимации касательными

$$\Delta\varphi = \arccos \frac{R}{R + \delta_B}.$$

Уточнение этого шага производится так же, как и при аппроксимации хордами.

Углы наклона радиусов, проведенных в опорные точки звеньев ломаной, к оси абсцисс можно определить по формулам:

– при аппроксимации хордами

$$\alpha_i = \alpha_0 + i\Delta\varphi;$$

– при аппроксимации секущими или касательными

$$\alpha_i = \alpha_0 + i\Delta\varphi + (i-1)\Delta\varphi,$$

где i – номер опорной точки ($i = 1, 2, \dots, k$).

Координаты опорных точек звеньев ломаной при аппроксимации хордами вычисляют по формулам:

$$x_i = R \cos(\alpha_i);$$

$$y_i = R \sin(\alpha_i);$$

при аппроксимации секущими или касательными по формулам:

$$x_i = (R + \delta_B) \cos(\alpha_i);$$

$$y_i = (R + \delta_H) \sin(\alpha_i).$$

Программа аппроксимации имеет вид:

```
{Uses CRT;}
```

```
Const
```

```
  R=150;
```

```
  a0=0;
```

```
  ak=pi/3;
```

```
  del_b=0.05;
```

```
  del_n=0.1;
```

```
Var
```

```
  fi,d_fi0,d_fik,d_fi,al:real;
```

```
  x,y:Real;
```

```
  a:Word;
```

```
function ArcCos(X: Real): Real;
```

```
begin
```

```
  if X <= -1.0 then
```

```
    ArcCos := Pi
```

```
  else if X < -Sqrt(2) / 2 then
```

```

    ArcCos := Pi + ArcTan(Sqrt(1 - Sqr(X)) / X)
else if X <= Sqrt(2) / 2 then
    ArcCos := Pi / 2 - ArcTan(X / Sqrt(1 - Sqr(X)))
else if X < 1.0 then
    ArcCos := ArcTan(Sqrt(1 - Sqr(X)) / X)
else
    ArcCos := 0.0
end;

```

BEGIN

```

    fi:=ak-a0;
    d_fi0:=arccos(R/(R+del_b));
    d_fik:=d_fi0;
    d_fi:=2*arccos((R-del_n)/(R+del_b));
    a:=Trunc((fi-(d_fi0+d_fik))/d_fi);
    d_fi:=(fi-(d_fi0+d_fik))/a;
    For var i:=0 to a+1 do
        Begin
            al:=a0+d_fi*i);
            x:=R*cos(al);
            Y:=R*sin(al);
            Writeln(x = ',x:3:3,' y =',y:3:3)
        End;
    Readln;

```

END.

Правильность программы аппроксимации проверена контрольным просчетом, показавшим совпадение с эталоном [5].

Список литературы

1. Using low-cost open source hardware to control puma560 motors / P. Reguera, S. Alonso, M. Domínguez, M.A. Prada, A. Morán, J.J. Fuertes / www.sciencedirect.com.
2. Development of a new controller with FPGA for PUMA 560 robot / Dejan Ž. Joki, Slobodan D. Lubura, Stevan Stankovski / 12th IFAC Conference on Programmable Devices and Embedded Systems The International Federation of Automatic Control September 25-27, 2013. Velke Karlovice, Czech Republic.
3. <http://www.giden.ru/news/274-modernizatsiya-robota-kuka>.
4. Фу К., Гонсалес Р., Ли К. Робототехника: – М.: Мир, 1989. – 624 с.
5. Дерябин А.Л. Программирование технологических процессов для станков с ЧПУ. – М.: Машиностроение, 1984. – 216 с.

Сведения об авторах:

Романов Владимир Викторович – к.т.н., доцент кафедры «Технология машиностроения»;

Петровский Владимир Сергеевич – д.т.н., профессор, заведующий кафедрой «Технологии машиностроения».