

РЕАЛИЗАЦИЯ АЛГОРИТМА ДЕЛЕНИЯ ДУГИ ОКРУЖНОСТИ НА УЧАСТКИ РАВНОЙ ДЛИНЫ

Буевич Т.В.¹, к.т.н., доц., Буевич А.Э.², к.т.н., доц., Клебанов С.А.¹, студ.

¹Витебский государственный технологический университет,

²Витебская ордена «Знак Почета» государственная академия ветеринарной медицины, г. Витебск, Республика Беларусь

Реферат. В статье рассмотрены разработка и реализация алгоритма для получения траектории перемещения исполнительного устройства по заданным координатам на дуге окружности. Программный модуль позволяет расширить возможности действующих на предприятиях систем автоматизированного проектирования при разработке управляющих программ к технологическому оборудованию с электронным управлением в производствах легкой промышленности.

Ключевые слова: алгоритм, автоматизированное проектирование, управляющая программа.

Автоматизировать решение производственных задач без привлечения дополнительного программного обеспечения позволяет интегрирование в действующие на предприятии системы автоматизированного проектирования модулей, расширяющих их возможности.

В мехатронных системах часто требуется отработка перемещений исполнительного устройства по заданным координатам. Разработан алгоритм деления траектории в виде дуги окружности на узлы (точки) на заданном расстоянии n_0 друг от друга. На рисунке 1 изображена расчетная схема алгоритма разделения дуги на участки равной длины. При представлении в векторной форме дуга описывается координатами конечных точек 1, 2 и коэффициентом кривизны γ . На рисунке 1 обозначены: точка 1 с координатами (x_1, y_1) , точка 2 с координатами (x_2, y_2) , l_d – длина дуги, l – расстояние между точками 1 и 2, Δl – уточненное расстояние между соседними точками P_i , R – радиус дуги, F – центральный угол.

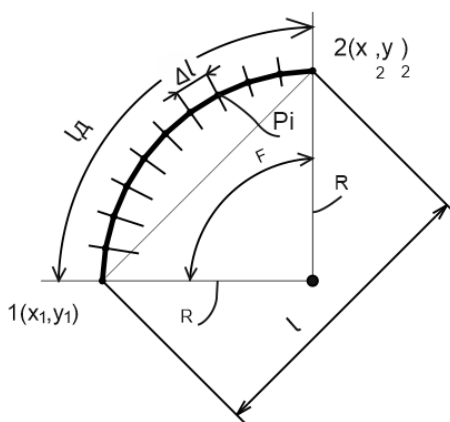


Рисунок 1 – Расчетная схема

Исходные данные о дуге окружности, созданной в системе AutoCAD, содержатся в файле обмена чертежами, который представляет собой обычный текстовый файл типа «.dxf» в кодах ASCII. В «.dxf файле» находится текстовая информация о векторном изображении в специально заданном формате. Координаты точек 1 и 2 начала и конца графического примитива «дуга окружности» (полилинии) содержатся в группах, определяемых соответствующими кодами. Так, код группы «10» указывает на первичную координату X (начальную точку линии или текстового графического примитива, центр окружности и т.д.); код группы «20» указывает на первичную координату Y. Значения второй координаты всегда соответствуют значениям первой координаты и следуют в файле непосредственно за ними. Код «42» означает, что к следующей точке переход осуществляется по дуге. Кривизна дуги γ записана в следующей строчке «.dxf файла» и выражается коэффициентом, который

принимает значения от -1 до 1. Фрагмент программы определения исходных данных о дуге окружности представлен на рисунке 2.

<pre> If st=' 10' then Begin Readln(Var_f,st); Val(st,cr,cod); x2:=cr; end; ... x1:= x2; </pre>	<pre> If st=' 20' then Begin Readln(Var_f,st); Val(st,cr,cod); y2:=cr; end; ... y1:= y2; </pre>	<pre> If st=' 42' then Begin Readln (Var_f,st); Val (st,cr,cod); R:=Cr; end; </pre>
---	---	---

Рисунок 2 – Фрагмент программы для определения исходных данных о дуге

При известных координатах точки 1(x_1, y_1) и точки 2(x_2, y_2) расстояние между точкам 1 и 2 определяется по выражению:

$$l = \sqrt{((x_2 - x_1)^2 + (y_2 - y_1)^2)}. \quad (1)$$

По коэффициенту кривизны r вычисляются радиус R и центральный угол F :

$$R = \frac{r \cdot l + l / r}{4}, \quad (2)$$

$$F = \arccos \frac{(2 \cdot R^2 - l^2)}{2 \cdot R}. \quad (3)$$

Определяется значение вспомогательных углов F_1 , дополняющих с двух сторон центральный угол F до 180 градусов.

Реализация алгоритма вычисления расстояния l между точками 1 и 2, радиуса дуги R , центрального угла F и вспомогательных углов F_1 представлена на рисунке 3.

```

...
l:=sqrt(sqr(x2-x1)+sqr(y2-y1));
R:=((R*l+l/R)/4);
F:=arccos((2*sqr(R)-sqr(l))/(2*sqr(R)));
F1:=(Pi-F)/2;
...

```

Рисунок 3 – Фрагмент программы для вычисления l , R и F

Рассчитывается длина дуги l_d и находится целое число N отрезков длины n_0 , которое помещается в найденной длине:

$$l_d = F * R; \quad (4)$$

$$N = \left\lfloor \frac{l_d}{n_0} \right\rfloor. \quad (5)$$

Определяется угловой шаг Δf и линейные приращения Δx , Δy по координатным осям:

$$\Delta f = \frac{F}{N}; \quad (6)$$

$$\Delta x = R * \cos(\Delta f); \quad (7)$$

$$\Delta y = R * \sin(\Delta f). \quad (8)$$

На рисунке 4 представлен фрагмент программы для определения числа N отрезков и уточнения их длины. Переменная L_d описывает длину дуги радиуса R , переменная Lim

описывает предварительно заданное пользователем расстояние p_0 , переменная *Koef* уточненное число отрезков длины p_0 , которые помещаются в длину дуги между точками 1 и 2, переменные *Fi*, *Xrel*, *Yrel* описывают, соответственно, угловой шаг и линейные приращения по координатным осям

```

...
Ld:=F*R;
Koef:=Floor(Ld /Lim);
if Koef<=1 then begin Koef:=1 end;

Fi:=Fi/Koef;
Koef:=abs(Koef);

Xrel:=R*cos(F1);
Yrel:=R*sin(F1);
...

```

Рисунок 4 – Фрагмент программы для определения числа и уточнения длины отрезков

После этого определяются координаты соседних точек $P_i(x_i, y_i)$ дуги:

$$x_i = x_{i-1} + \frac{\Delta x(x_2 - x_1)}{l} - \frac{\Delta y(y_2 - y_1)}{l}, \quad (9)$$

$$y_i = y_{i-1} + \frac{\Delta y(x_2 - x_1)}{l} + \frac{\Delta x(y_2 - y_1)}{l}. \quad (10)$$

На рисунке 5 представлен фрагмент программы для вычисления координат точек P_i , принадлежащих дуге. Переменные *Xr*, *Yr* определяют координаты (x_i, y_i) текущей точки дуги. Значения координат текущей точки находят приращением по дуге координат начальной точки, обозначенных переменными *X1*, *Y1*. После расчета значения координат текущей точки присваиваются новой начальной точке и обозначаются переменными *X2*, *Y2*. Цикл расчета повторяется *Koef* раз. Рассчитанные координаты точек дуги окружности с помощью оператора *Write* сохраняются в файл, который передается в интегрированную САПР.

```

...
Xr:=X1+Xrel*((X2-X1)/l)-Yrel*((Y2-Y1)/l);
Yr:=Y1+Xrel*((Y2-Y1)/l)+Yrel*((X2-X1)/l);

for n:=1 to Koef do
Begin
X2:=Xr;
Y2:=Yr;
Write(Inp_f, "");
Write(Inp_f, X2_p.cod:10, ', ');
Write(Inp_f, Y2_p.cod:10);
WriteLn(Inp_f, "");
End;
...

```

Рисунок 5 – Фрагмент программы для вычисления координат точек дуги

Разработанный программный модуль предназначен для расчета координат точек, расположенных на дуге окружности с заданным шагом, и определения траектории перемещения исполнительного устройства.