VOL. 13, NO. 5, MARCH 2018 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

ALGORITHMIC SUPPORT OF PROBLEMS OF ELECTRONIC KINEMATICS

Inga Nikolaevna Bulatnikova and Natalja Nikolaevna Gershunina
Department of Applied Mathematics, Kuban State Technological University, Moskovskaya, Krasnodar, Russia
E-Mail: inkras@yandex.ru

ABSTRACT

This article discusses digital simulation and possible implementation of kinematic mechanisms and units on the basis of simple computational means (microprocessors). The proposed digital simulation is supported by geometrical plotting methods developed in descriptive geometry and successfully applied in engineering graphics. These methods, aided by compass, ruler, protractor and triangular ruler, make it possible to plot any trajectory, for instance, kinematic one. Technological basis of this simulation is a set of rapid algorithms based on integer arithmetic carried out by RISC microprocessors. Possibilities of this approach to designing of algorithmic support are exemplified by evolvent of circle. The obtained set of algorithms does not contain operations of multiplication and division; nevertheless, quite complex trajectories are obtained without trigonometric computations. The efficiency of the proposed procedure of algorithm designing is based on their fast operation, low cost of designing and simplicity of software implementation using microprocessors.

Keywords: mechatronics, integer algorithms, microprocessors, machinery designs.

1. INTRODUCTION

Academician Frolov, a prominent engineer, introduced the term "electronic kinematics". Its essence is comprised of provision of preset synchronous motions of working tools not by means of conventional mechanic kinematic tools (joints, rods, crankshafts, cams and so on) but by power drives (usually electric) synchronously controlled by computers [1].

This simplifies designing and manufacturing of complex mechanic systems reducing it to synthesis of existing algorithms and their implementation by computers.

Taking into consideration widespread application of computers we recommend to use microprocessors.

In this regard an urgent issue is efficient algorithmic support of microprocessors which are used for solution of electron kinematic problems.

Herewith, it is required to account for microprocessor architecture which differs significantly from regular PC architecture.

In our works we proved the efficiency of microprocessor information technologies based on integer arithmetic [2, 3, 4].

Such integer algorithms are very fast and accurate, making them efficient upon implementation by microprocessors, for instance, of RISC architecture.

Since all units of mechanisms are distributed and move in space, then it is important to proceed from static algebraic computational models to dynamic geometric models.

Moreover, it should be taken into account that many algebraic problems are solved better by geometrical methods. Such approach would provide sharp increase in processing rate of control information not only due to internal clock rate (in mHz) but mainly due to algorithm operation speed.

Development of such algorithms is aided by geometrical plotting methods developed in descriptive geometry and widely applied in engineering graphics.

Thus, on the basis of integer algorithms of digital linear interpolation (ruler) and circular interpolation (compass), algorithms of measurement of angular motions (protractor), determination of perpendicular to straight line (triangular ruler) it is possible proceed to integer arithmetic and, hence, to relieve sharply requirements of architecture of microprocessors which carry out such integer algorithms.

The most important requirement to relieve requirements to microprocessors is elimination of multiplication and division in their system of commands. This increases their operation speed and data processing accuracy in solving problems of electron kinematics.

2. MAIN COMPUTATIONAL METHODS ON THE BASIS OF INTEGER ARITHMETIC

Tools of integer data processing for electronic kinematic problems are listed below.

2.1 Difference iterative methods

Difference iterative methods are non-analytical computational methods eliminating multiplication and division [2] required for solution of electronic kinematic problems.

For instance, the function $z = \frac{(x-y)^2}{y}$ can be computed using such algorithm (for $x > 0, \quad y > 0$):

$$q_{i-1} = signZ_{i-1} = \begin{cases} +1, \ if \ Z_{i-1} > 0; \\ -1, \ if \ Z_{i-1} < 0; \\ 0, \ if \ Z_{i-1} = 0, \ \textbf{stop} \ ; \end{cases} \qquad (1)$$

$$Z_0 = x - y,\ Z_i = Z_{i-1} - q_{i-1} \cdot y \cdot 2^{-i},\ Z_n \to 0;$$

$$Y_0 = x - y,\ Y_i = Y_{i-1} + q_{i-1} \cdot x \cdot 2^{-i},\ Y_n \to \frac{(x-y)^2}{y}.$$

Here and below $i$is the number of iteration, $i = 1,2,\ldots,n-1;$ $n$is the binary digits of operands x and y.

2.2 Vector pseudo-rotation methods

These methods were proposed in [3,4,5]. The "digit-by-digit" methods can be exemplified as follows (at $y > 0,\ \ x > 0$):

$$\xi_{i-1} = signY_{i-1} = \begin{cases} +1, \text{ if } Y_{i-1} \ge 0; \\ -1, \text{ if } Y_{i-1} < 0; \end{cases} \tag{2}$$

$$Y_0 = y,\ Y_i = Y_{i-1} - \xi_{i-1} \cdot X_{i-1} \cdot 2^{-(i-1)},\ Y_n \to 0;$$

$$X_0 = x,\ X_i = X_{i-1} + \xi_{i-1} \cdot Y_{i-1} \cdot 2^{-(i-1)},\ X_n \to K \cdot \sqrt{x^2 + y^2};$$

$$\theta_0 = 0,\ \theta_i = \theta_{i-1} + \xi_{i-1} \cdot arctg\, 2^{-(i-1)},$$

$$Q_n \to arctg\, \frac{y}{x},$$

where K is the coefficient:

$$K = \prod_{m=0}^{n-1} \sqrt{1 + 2^{-2m}}.$$

Algorithm for rotation of coordinates by angle α is available, similar to algorithm (2).

2.3 Digital linear interpolation

There are numerous algorithms of digital interpolation of straight lines. Some of them, for the sake of simplicity, permit absolute error equaling to one step of interpolation (Δ).

We developed the so-called algorithm of digital linear interpolation with twice as lower error $\Delta/2$ and double operation speed in comparison with non-optimized algorithms [6]. This provides more accurate geometrical simulation of kinematic systems upon their designing.

2.4 Digital circular interpolation

The situation with circular interpolation is the same as with linear one: there are numerous non-optimized algorithms with absolute error up to Δ.

We substantiated and proposed optimum algorithm of digital circular interpolation with absolute error equaling to $\Delta/2$. Operation speed is also doubled [6].

This is achieved by that the algorithm estimates deviation (evaluation function) in two adjacent alternative interpolation nodes and selects the node with minimum absolute value of evaluation function. The evaluation function is adjusted by means of integer expressions, such as:

$$F_{i+1} = \begin{cases} F_i + 2x_i + 1 \ \textbf{upon movement along } x \textbf{ axis,} \\ F_i + 2x_i - 2y_i + 2 \ \textbf{upon movement along diagonal,} \\ F_i - 2y_i + 1 \ \textbf{upon movement along } y \textbf{ axis;} \end{cases} \tag{3}$$

where x_i and y_i are the coordinates of current node of circle interpolation.

Algorithm of digital circular interpolation is readily modified for angular rotations in the range of 0÷360° as well as for reverse motion along arc of circle (clockwise or counterclockwise).

2.5 Angular rotation of radius vector

Using integer algorithm of measurement of actual rotation of radius vector, digital circular interpolation determines angles of rotation of moving parts. This is required, for instance, upon simulation of complex kinematic curves (cycloid, epi- and hypocycloid).

This algorithm is based on consideration for interrelation between the value of angular sector and its area. The latter is calculated by increments at each step of digital interpolation using simple equation (such as area of rectangular triangle with unit base, interpolation step Δ) [7]

$$\Delta S_i = \begin{cases} 0.5 \cdot y_i \ \textbf{upon movement along } x \textbf{ axis,} \\ 0.5 \cdot (x_i + y_i) \ \textbf{upon movement along diagonal,} \\ 0.5 \cdot y_i \ \textbf{upon movement along } \ y \textbf{ axis.} \end{cases} \tag{4}$$

2.6 Alignment of perpendicular to straight line

Equation of perpendicular to this straight line in predefined point (x_0, y_0) (or passing via predefined point) is based on known expressions of analytical geometry. Their angular coefficients are correlated as k and $-1/k$ (where k is the angular coefficient of preset straight line) and perpendicular offset b_2 is calculated as follows:

$$b_2 = y_0 + x_0/k \tag{5}$$

Since simulation of perpendicular as straight line requires only for expression of its evaluation function, then ΔX and ΔY are important (parameters of digital linear interpolation) which are correlated with the angular coefficient k as follows:

$$k = -\frac{\Delta X}{\Delta Y}. \tag{6}$$

This correlation is used, for instance, upon determination of parameters of digital interpolation of tangent to circle.

2.7 Dynamic multiplication of variables

This procedure makes it possible to multiply variables in integer format. This multiplication is considered as dynamic because each variable varies in time (step-by-step), but with condition that in each step it varies discretely (integer arithmetic) by ± 1 or does not vary (zero increment).

Such variables can be, for instance, coordinates of successive node of digital interpolation of this or that line (straight line, circle and so on). This procedure is carried out according to the recursive equation:

$$y_{i+1} = y_i + x_{1i}(t)\cdot\delta_{2i} + x_{2i}(t)\cdot\delta_{1i} + \delta_{1i}\cdot\delta_{2i}, \quad (7)$$

where i is the step number, i= 0, 1, 2, 3, …; $x_{1i}(t)$, $x_{2i}(t)$, are the current values of the first and the second variables; δ_{2i}, δ_{1i} are the increments of the first and the second variables in the i-th step equaling to the elements of set {0, 1, –1}.

At the initial moment $(i=0)$ the variable $y_0 = x_{10}(t)\cdot x_{20}(t)$ is a parameter (constant) of integer algorithm.

2.8 Concept of geometric simulation of kinematic systems

Geometric simulation is based on the thesis: many algebraic problems are solved simpler by geometrical method. Well-known plotting methods of descriptive geometry are successfully applied in engineering graphics and make it possible to determine complex configurations and curves. Four major graphic tools are used for this purpose: ruler, compass, protractor, triangular ruler.

All these tools can be substituted by integer algorithms of digital linear, circular interpolation, algorithm of measurement of angles and algorithm of determination of parameters of digital interpolation of perpendicular as a straight line.

Some other algorithms are also used including logical algorithms.

Advantage of integer algorithms is based on their operation speed due to implementation by efficient computing means: microprocessors, for instance, of RISC architecture. Another advantage of such algorithms is their improved accuracy due to eliminated errors. This is achieved by non-use of floating point operations as well as multiplication and division.

Using these algorithms, it is possible to carry out trajectories of separate elements of kinematic systems and, essentially, to control positions of required movable elements (for instance, manipulator gripper).

This is one of electronic kinematic problems declared at the beginning of this article [10].

3. CASE STUDY OF EVOLVENT DIGITAL INTERPOLATION

In the case study of this curve - evolvent of circle - we will show how, using integer algorithms and model properties of this curve, to achieve motion of working element (for instance, cutting tool) along the evolvent trajectory.

This is based on algorithms of digital circular interpolation [6] and generalized algorithm of digital interpolation of arbitrary curves [8]. The essential concept of the algorithm is illustrated in Figure-1. The control of evolvent steps is comprised of determination of the fact that the current point C of evolvent reaches the tangential AC after one or several steps of digital interpolation of evolvent (arc BC).

This is determined by the sign of evaluation function for the tangential AC passing via the point A (x_i, y_i) with angular coefficient $k = -x_i / y_i$. Evaluation function of the tangential is as follows:

$$F(x, y) = y - kx - b, \text{ where } b = y_i - kx_i \quad (8)$$

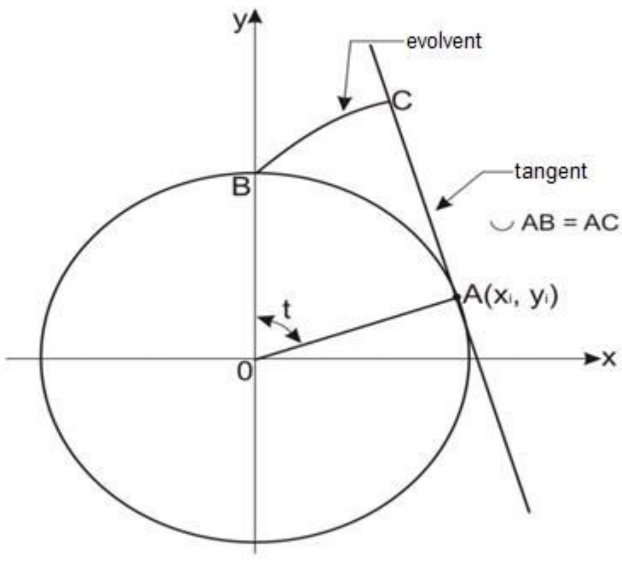

Figure-1. Algorithm of evolvent interpolation.

In the points above the tangential it is positive, and below the tangential it is negative. On the tangential it is zero. This zero crossing denotes the end of steps of digital interpolation after each step of digital interpolation of circle.

Them, the next step of circular interpolation is performed and the process is repeated. Herewith, this is accompanied by variation of the coordinates of point A (not more than by ±1), the angular coefficient k and, respectively, the value of evaluation function $F(x, y)$ of new tangential AC.

Digital interpolation of evolvent is performed according to generalized algorithm [8] with varying ΔX and ΔY equaling to x_i and y_i, respectively. Finally, the evaluation function of tangential for the first quadrant is as follows:

$$F(x, y) = y_i \cdot y + x_i \cdot x - (x_i^2 + y_i^2). \tag{9}$$

The sum in brackets, Equation (9), equals approximately to a^2, where a is the circle radius. It exactly equals to a^2 for the nodes of circular interpolations on the circle, and for remaining points it is:

$$(x_i^2 + y_i^2) - \Delta f_i = a^2, \tag{10}$$

where Δf_i is the excess of $(x_i^2 + y_i^2)$ over a^2 due to that the i-th node is outside the circle. In this case $\Delta f_i > 0$, if it is inside, then $\Delta f_i < 0$.

In the proposed algorithm Δf_i, equaling to zero at the beginning, for other nodes is calculated according to iterative integer expression and subtracted from the evaluation function $F(x, y)$.

The algorithm of adjustment is as follows. If the step prior to the i-th one was coordinate ($x_i = x_{i-1} + 1$ or $y_i = y_{i-1} - 1$), then Δf_i will equal to $(2 \cdot x_{i-1} + 1)$ or $(-2 \cdot y_{i-1} + 1)$, and if it was diagonal ($x_i = x_{i-1} + 1$ and $y_i = y_{i-1} - 1$), then $\Delta f_i = 2(x_{i-1} - y_{i-1} + 1)$. These equations follow from Eq. (1.3).

Let us temporarily omit Δf_i from the considerations. Then, we have in simplified form:

$$F(x, y) = y_i \cdot y + x_i \cdot x - a^2. \tag{11}$$

Zero of Eq. (11) follows from equation of circle with the radius a for all points of the tangential AC. In order to avoid misunderstanding in further conclusions let us denote coordinates of evolvent as w (abscissa) and u (ordinate) for all its current points.

Then, the coordinates of the next point (interpolation node) will be as follows:

$$w_{j+1} = w_j + dw_j,\ u_{j+1} = u_j + du_j, \tag{12}$$

where dw_j and du_j are the possible increments of evolvent coordinates upon interpolation equaling to 0 or 1 at the j-th step of interpolation according to generalized algorithm for arbitrary curves.

Let us determine the increment $\Delta F(w_{j+1}, u_{j+1})$ as difference:

$$F(w_{j+1}, u_{j+1}) - F(w_j, u_j) =$$
$$= \Delta F(w_{j+1}, u_{j+1}) = y_i \cdot du_j + x_i \cdot dw_j \tag{13}$$

Multiplication operations in Equation (13) by 0 or 1 are substituted with additions (if by 1).Then, the evaluation function is adjusted for the last point (node) on evolvent according to the expression similar to Equation (13) using increment:

$$\Delta F(x_{j+1}, y_{j+1}) = u_i \cdot dy_j + w_i \cdot dx_j, \tag{14}$$

where dy_j и dx_j are the increments of coordinates of current node of circular interpolation (0 or ±1). Equation (14) also does not contain operations of multiplication. Initial value of evaluation function $F(x, y)$ in the point B (origin of exponent) is zero.

Therefore, the algorithm of digital interpolation of evolvent is separated into two parts:

a)the driving algorithm is that of circular interpolation (exactly one step is performed and the second part is activated);

b)the driven algorithm of digital interpolation (one or several steps are performed until the evaluation function of tangential AC $F(x_i, y_i)$ varies its sign).

Herewith, parameters for the second part of algorithm are provided by the first part in the form of successive point (x_i, y_i) of circular interpolation. After change of sign of evaluation function of tangential as a consequence of action of the second part the action of the first part is transferred.

4. RESULTS AND DISCUSSIONS

Finally, completely integer algorithm (without multiplication and division) is obtained, oriented at rapid implementation by microprocessors, for instance, of RISC architecture.

The obtained algorithm of digital interpolation of evolvent was approbated on PC using Delphi by emulation of command system of modern RISC microprocessors. Quantitative results (maximum absolute errors) of digital interpolation of evolvent are summarized in Table-1.

Table-1. Error of digital interpolation of evolvent (in interpolation steps Δ = 0.1 mm).

a=100	*a*=250	*a*=500	*a*=2000	*a*=5000
0.793	0.814	0.825	0.916	0.990

The developed integer algorithm of digital interpolation of evolvent is recommended for application in numerical control machines, automated systems of designing and simulation of complex kinematic systems.

5. CONCLUSIONS

Conversion to integer algorithmization of information technologies of data processing for solution of problems of electronic kinemics is very promising. On the basis of simple technical aids (microprocessors) and using rapid integer algorithms it permits to implement not only simulation but physical replacement of elements of complex kinematic systems [7, 11, 12].

REFERENCES

[1] K.V. Frolov and V.I. Babitskii. 1986. Mehanikaiiskusstvokonstruirovaniya v epohu EVM. [Engineering and designing in PC age] Izobretatel` iratsionalizator. 12: 16-17.

[2] A.M. Oranskii. 1977. Apparatnyemetody v tsifrovoivychislitel`noitekhnike [Hardware in digital computing technique]. (BGU, Minsk).

[3] I.N. Bulatnikova and V.I. Clyuchko. 2011. Informatsionnyetekhnologii s ispol`zovaniemtselochislennoiarifmetiki [Information technologies using integer arithmetic]. Geoengineering. NIPI InzhGeo.2:54-57.

[4] V.D. Baikov and V.B. Smolov. 1985. Spetsializirovannyeprotsessory: iteratsionnyealgoritmyistruktury [Specialized processors: iterative algorithms and structures]. (Radio isvyaz`, Moscow).

[5] Adir, E. Almog, L. Fournier, E. Marcus, M. Rimon, M. Vinov, A. Ziv. 2004. Genesys-Pro: Innovations in Test. Program Generation for Functional Processor Vertification. Design and Test.

[6] N.S. Anishin. 1990. Metodologiyaproektirovaniyaalgoritmicheskogoobespecheniyamikroprotsessornykhustroistvlokal`noiavtomatiki (funktsional`noepreobrazovanieiobrabotkainformatsii) [Designing procedures of algorithmic support of microprocessors of local automation (functional transformation and data processing]: Doctoral thesis, specialty, Kiev.

[7] Morris. 1985. Coordinate transformations and programming for small revolute - coordinate robots. Microprocessors and Microsystems. 9(6): 290-295.

[8] I.N. Bulatnikova. 2011. Tsifrovyeinterpolyatorykrivolineinykhtraektorii. [Digital interpolators of curved paths] Izv.vuzov. Sev.-Kavkazskii region. Ser. Tekhnicheskienauki. 2: 16-19.

[9] N.S. Anishin. 2013. Modelirovaniekinematikiploskikhmehanizmovnabazetselochislennykhalgoritmov. [Simulation of kinematics of plain mechanisms on the basis of integer algorithms] Izv.vuzov. Sev.-Kav. region. Ser. Tekhnicheskienauki. 4: 22-25.

[10] K.P. Corwell, et al. 1985. Computers, complexity and controversy. IEEE Computer. 18(4): 8-19.

[11] E.Vitrant, Canudas-De-Vit C., D. Georges, 2004. Alamir Remote stabilization via time - varying communication network delays. IEEE Conf. in Control Applications, Taiwan. Sept 2-4.

[12] R.E. Carlyle, 1985. Riscy business. Datamation. 18(4): 18-19.