

Calculation method linkages. Universal approach.

A. Ivanov.

Description ideas.

Linkages will be considered from the point of view of the geometric relations between the points of the mechanism, at the same time mechanisms will not be divided into planar and spatial. As a mathematical model describing the positions of the points of mechanism proposed underdetermined system of equations or a system of equations with free variables. Kinematics calculation is based on the normalization of the provisions of all points depending to the position of the entry point of mechanism. Underdetermined systems of equations will be solved on the basis of the method Draghilev solving systems of nonlinear equations.

Draghilev's method develops the method of continuation on parameter solutions. Continuation of the parameter solutions involves the use as an independent variable parameter artificially entered in or selected from a number of variables in the system. Take one of the easiest options for the continuation method on parameter. Let (1) a system of equations with respect to X :

$$\begin{aligned} F(X) &= 0; \\ F &= (f_1, \dots, f_n), \quad f_i \in R^1, \quad f_i = f_i(X), \quad X = (x_1, \dots, x_n), \quad x_i \in R^1; \end{aligned} \quad (1)$$

At point $X_0 = (x_{01}, \dots, x_{0n})$ the right side of the system (1) is $F(X_0) = F_0$. In the new system of equations:

$$\begin{aligned} F(X) - t \cdot F_0 &= 0; \\ t &\in R^1; \end{aligned} \quad (1a)$$

realized variant of the method of continuation on parameter solutions [1], [2]. When $t = 0$, (1a) is transformed to (1). Assuming that $x_i = x_i(t)$, $i = 1, \dots, n$, $t \in [0, 1]$, then changing the value of t from 1 to 0, it is possible to obtain a solution of (1), moving from the solutions (1a). A.V. Draghilev proposed a numerical analytical method for producing the functional dependence of all variables in (1a) with respect to curve length in R^{n+1} [3]. Using this dependency, we can find the solution of systems of the form (1), and obtain an infinite set of solutions of equations of the form:

$$\begin{aligned} F(X) &= 0; \\ F &= (f_1, \dots, f_n), \quad f_i \in R^1, \quad f_i = f_i(X), \quad X = (x_1, \dots, x_{n+m}), \quad m \geq 1, \quad x_j \in R^1; \end{aligned} \quad (1b)$$

on connected areas of subsets of solutions previously knowing any one point $X_0 = (x_{01}, \dots, x_{0n+m})$ belonging to this subset. Solution (1b) we will seek as the solution of the Cauchy problem of autonomous ODE system. Let $m = 1$, we obtain:

$$\begin{aligned} F(X) &= 0; \\ F &= (f_1, \dots, f_n), \quad f_i \in R^1, \quad f_i = f_i(X), \quad X = (x_1, \dots, x_{n+1}), \quad x_j \in R^1; \end{aligned} \quad (2)$$

We are interested in the case when (2) has an infinite number of solutions. Geometrically, the solution of this system can be regarded as the set of points of the curve in space. Let the length of the curve t , assuming that the variables (the coordinates of the points of the curve) are continuously differentiable functions of t . Let us differentiate (2) with respect to t .

$$\left\{ \begin{array}{l} \frac{\partial f_1}{\partial x_1} \frac{dx_1}{dt} + \dots + \frac{\partial f_1}{\partial x_{n+1}} \frac{dx_{n+1}}{dt} \\ \cdot \\ \cdot \\ \cdot \\ \frac{\partial f_n}{\partial x_1} \frac{dx_1}{dt} + \dots + \frac{\partial f_n}{\partial x_{n+1}} \frac{dx_{n+1}}{dt} \\ x_j(0) = x_{0j}, j = 1, \dots, n+1; \end{array} \right. = 0; \quad (3)$$

The solution of equation (3) is a solution of equation (2), and this solution describes the part of the curve or the whole curve from any single starting point $x_j(0) = x_{0j}$, $j = 1, \dots, n+1$, which belong to this curve.

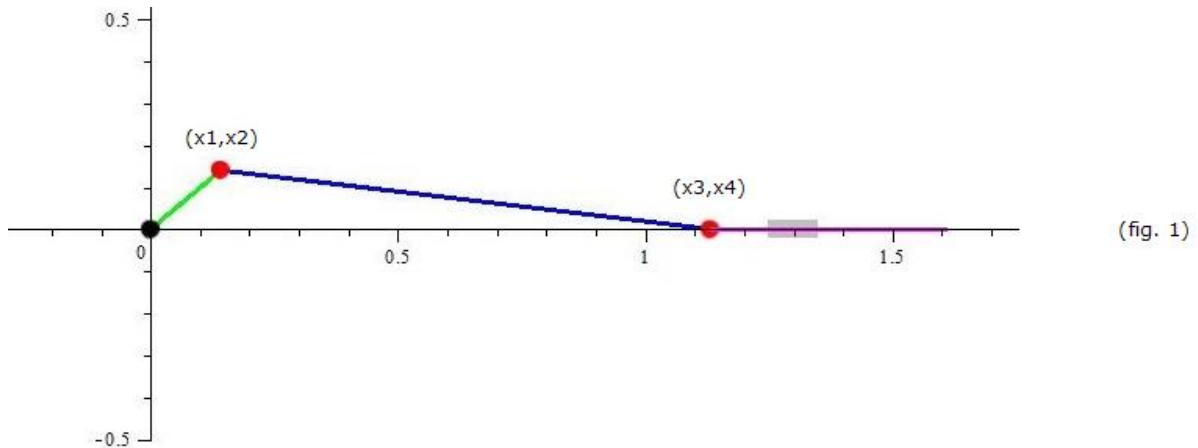
In the calculations for obtain one-to-one correspondence between the length of the curve and parameter t we will set a step h by parameter t and equate $h = \left(\sum_{j=1}^{n+1} dx_j^2 \right)^{\frac{1}{2}}$. When we need the projection of the curve onto any subspace, then we will use only the coordinates of this subspace (this issue will be discussed in greater detail in the first application example).

In order to in (3) to express derivatives $\frac{dx_j}{dt}$ used Cramer's rule for solving systems of linear equations. As a free variable may be any of the derivatives $\frac{dx_j}{dt}$. Let the free variable, we will always be, for example, $\frac{dx_{n+1}}{dt}$. To remove the denominators from the system (3), A.V. Draghilev proposed to equate $\frac{dx_{n+1}}{dt}$ to main determinant of the system (3). Then (3) takes the form:

$$\left\{ \begin{array}{l} \frac{dx_j}{dt} = \det \left[\begin{array}{ccc} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_{n+1}} \dots \frac{\partial f_1}{\partial x_n} \\ \cdot & & \cdot \\ \cdot & \text{(j-column)} & \cdot \\ \cdot & & \cdot \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_{n+1}} \dots \frac{\partial f_n}{\partial x_n} \end{array} \right], j = 1, \dots, n; \\ \frac{dx_{n+1}}{dt} = -\det \left[\begin{array}{ccc} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{array} \right]; \\ x_j(0) = x_{0j}, j = 1, \dots, n+1; \end{array} \right. \quad (3a)$$

Examples. (see attachment)

Let us turn directly to the calculation of the kinematics of linkages based solutions underdetermined systems of equations. Consider some examples. The first example is a classic planar slider-crank mechanism (fig 1):



(Text figure 1.mw program in the attachment. Maple 15.)

The fixed point of the crank is at the origin, the movable point has coordinates (x_1, x_2) . Conrod points have coordinates (x_1, x_2) and (x_3, x_4) . Point of slider (x_3, x_4) moves along oX axis. We write the equations that describe the motion of the mechanism. The first equation defines the rotation of the crank (length 0.2) around the origin: $x_1^2 + x_2^2 - 0.2^2 = 0$ the second equation corresponds to the constant length 1 of conrod: $(x_3 - x_1)^2 + (x_4 - x_2)^2 - 1^2 = 0$. Providing a point conrod (slider) belongs the oX axis corresponds to the equation: $x_4 = 0$. We have a system of three equations for the four variables. This corresponds to one degree of freedom mechanism. To solve the system we use the method Draghilev. At first we will find any arbitrarily fixed position of mechanism. For example, the coordinates: $x_{01} = 0.14142136$, $x_{02} = 0.14142136$, $x_{03} = 1.13137085$, $x_{04} = 0$ corresponds to the solution of the system, and these values we use as initial data for the solution of the Cauchy problem (3a). (see attachment)

It has been said that we consider in more detail how to obtain the functional dependence of the desired coordinates (x_1, x_2) and (x_3, x_4) with respect to length of the curve. In this case, it is the functional dependence of the desired coordinates with respect to projection of the curve on the R^2

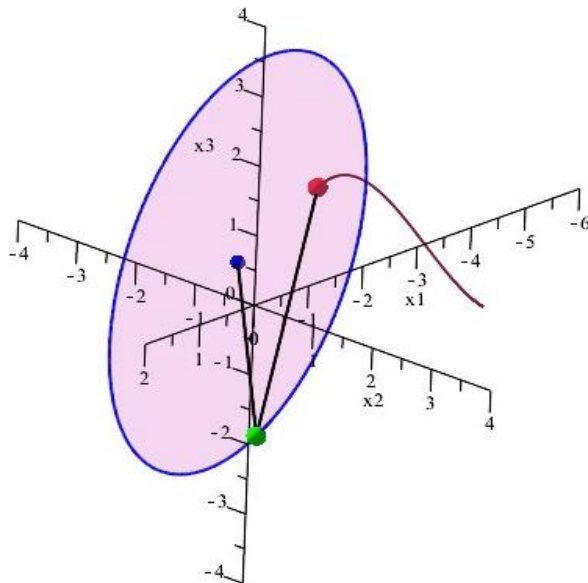
(space of the variables (x_1, x_2)). To do this, we assign a $h = \left(\sum_{j=1}^2 dx_j^2\right)^{\frac{1}{2}}$, where the h is step of

integration, and then all the position of the mechanism depends on length of the curve in space R^2 .

Since we have $h = \text{constant}$, and at the same $h = \left(\sum_{j=1}^2 dx_j^2\right)^{\frac{1}{2}}$, therefore the crank rotation at a constant

angular velocity. Step movement along the trajectory of the mobile point of the crank change the angle of the circle. As can be seen from the solution, we have the dependence of all the variables with respect to t : $x_1 = x_1(t)$, $x_2 = x_2(t)$, $x_3 = x_3(t)$, $x_4 = x_4(t)$, where t can be represented either as the angle, and as the curve length. In the program the integration interval specified as **smín: = 0 .;** **smax: = evalf (2 * Pi * r)**, the integration step **h: = 0.025**, that is, if we reduce the length of r times, you will only have to change the angle of the circle.

In the second example performs calculation of spatial slider-crank mechanism with the movement of slider along the spatial transcendental trajectory (fig 2):



(fig. 2)

(Text figure 2.mw program in the attachment. Maple 15.)

Despite its apparent complexity of the slider trajectory, the calculation program itself is not fundamentally different from the previous program of the planar mechanism. The system of equations describing the moving of mechanism: the trajectory of the mobile point of the crank is given by the first two equations: $f1$ and $f2$, $f3$ is length of conrod, $f4$ and $f5$ describe the trajectory of slider movement (red point).

In the program we can add command of output kinematics based on numerical differentiation, because we got the functional dependence of coordinates with respect to length of the selected path (in the above examples this is the trajectory of the moving point of the crank). Obviously, that easy to calculate any geometric parameters of mechanism and process of their change.

Final part.

We have considered the case where the system (1b) $m = 1$, which corresponds to one degree of freedom of the mechanism. When $m > 1$, the method Draghilev operates similarly as in the case $m = 1$. Only the solution of system is partitioned into a sequence of solutions of systems, each system when $m = 1$. The number of such sequences infinitely, starting with $m = 2$, which is in practice will correspond to the number of the sorting paths from $K \cdot N$, where K number of calculated points, for example, "the first system with $m = 1$ ", and the N number of points "second system with $m = 1$ ". To avoid selecting from all possible paths $K \cdot N$, can alternatively be added immediately tentative trajectory in the form of additional equations to the original system.

References.

1. А. В. Баранов, Об одном методе вычисления комплексных корней системы нелинейных уравнений, Ж. вычисл. матем. и матем. физ., 1972, том 12, номер 1, 199–203
2. Дж. Ортега и В. Рейнболдт. Итерационные методы решения нелинейных систем уравнений со многими неизвестными. Издательство "Мир", Москва 1975 г. (James M. Ortega and Werner C. Rheinboldt. Iterative Solution of Nonlinear Equations in Several Variables. Academic Press, New York, 1970.)
3. Иванов А. Б., Пахоменков Ю. М. О применении численного метода решения уравнения Драгилева при синтезе квазигармонических сигналов. // Системы управления и обработки информации: Научн.-техн. сб. / ФНПЦ «НПО «Аврора». СПб., 2007. — Вып. 13. — С. 125-134

Attachment.

figure 1.mw Maple 15

```
restart;
with(LinearAlgebra): with(plots):
RR := proc (t1, t2, t3, t4) options operator, arrow; ((t1-t2)^2+(t3-t4)^2)^.5 end proc;
r := .2; OA1 := 0; OA2 := 0; rL := 1; N := 70; smin := 0.; h := 0.25e-1; smax := evalf(2*Pi*r);
```

```

f1 := (x1-OA1)^2+(x2-OA2)^2-r^2;
f2 := (x3-x1)^2+(x4-x2)^2-rL^2;
f3 := x4; x01 := .14142136;
x02 := .14142136;

x03 := 1.13137085;
x04 := 0.;
n := 3;
x := seq(eval(cat('x', i)), i = 1 .. n+1);
x0 := seq(eval(cat('x0', i)), i = 1 .. n+1);
F := [seq(unapply(eval(cat('f', i)), [x]), [x]), i = 1 .. n)];
A := MTM:-jacobian(F(x), [x]);
A := MTM:-subs(A, [x], [x(t)]);
ics := seq(x[i](0) = x0[i], i = 1 .. n+1);
for i to n do b[i] := simplify(Determinant(DeleteColumn(ColumnOperation(A, [i, n+1]), n+1))) end do;
b[n+1] := simplify(-Determinant(DeleteColumn(A, n+1)));
deqs := seq(diff(x[i](t), t) = b[i]/(b[1]^2+b[2]^2)^.5, i = 1 .. n+1);
sln := dsolve([deqs, ics], numeric, method = rkf45, abserr = (1/10)*h, maxfun = 10000, range = smin .. smax);
a1 := sln(smin+smax*nn/N)[2];
a2 := sln(smin+smax*nn/N)[3];
a3 := sln(smin+smax*nn/N)[4];
a4 := sln(smin+smax*nn/N)[5];
AB := seq(plottools[line]([rhs(a1), rhs(a2)], [OA1, OA2], color = green, thickness = 3), nn = 0 .. N);
BC := seq(plottools[line]([rhs(a1), rhs(a2)], [rhs(a3), rhs(a4)], color = blue, thickness = 3), nn = 0 .. N);
CE := seq(plottools[line]([rhs(a3), rhs(a4)], [rhs(a3)+.5*rhs(a3)/RR(OA1, rhs(a3), OA2, rhs(a4)), 0.], color = RGB(1, .1, 4), thickness = 3), nn = 0 .. N);
EE := plottools[line]([1.3-0.5e-1, 0.], [1.3+0.5e-1, 0.], color = grey, thickness = 12);
OOO := seq(pointplot([rhs(a1), rhs(a2)], [rhs(a3), rhs(a4)]), color = [red], symbol = solidcircle, symbolsize = 12), nn = 0 .. N);
OO := pointplot([OA1, OA2], color = [black], symbol = solidcircle, symbolsize = 12);
E := seq(display(AB[k], BC[k], OO, EE, OOO[k], CE[k]), k = 1 .. N);
display(E, insequence = true, view = [-.5 .. 2, -.5 .. 2]);

```

figure 2.mw Maple 15

```

restart;
with(LinearAlgebra): with(plots):
r := 3; h := 0.25e-1; N := 120; smin := 0; smax := (2*3.05)*r;
f1 := x1^2+(x2+1)^2+(x3-.5)^2-r^2;
f2 := x1-.5*x2+.5*x3;
f3 := (x1-x4)^2+(x2-x5)^2+(x3-x6)^2-19;
f4 := sin(x4)-x5;
f5 := sin(2*x4)-x6;
x01 := 1.18325;
x02 := 1.18325;
x03 := -1.18325;
x04 := -2.07809;
x05 := -.87406;
x06 := .84926;
n := 5;
x := cat(x, 1 .. n+1);
x0 := cat(x0, 1 .. n+1);
F := [seq(unapply(eval(cat('f', i)), [x]), [x]), i = 1 .. n)];
A := MTM:-jacobian(F(x), [x]);
A := MTM:-subs(A, [x], [x(t)]);
for i to n do b[i] := simplify(Determinant(DeleteColumn(ColumnOperation(A, [i, n+1]), n+1))) end do;
b[n+1] := simplify(-Determinant(DeleteColumn(A, n+1)));
deqs := seq(diff(x[i](t), t) = b[i]/(b[1]^2+b[2]^2+b[3]^2)^.5, i = 1 .. n+1);
ics := seq(x[i](0) = x0[i], i = 1 .. n+1);
sln := dsolve([deqs, ics], numeric, method = rkf45, abserr = (1/100)*h, maxfun = 10000, range = smin .. smax);
a1 := sln(smin+smax*nn/N)[2];
a2 := sln(smin+smax*nn/N)[3];
a3 := sln(smin+smax*nn/N)[4];
a4 := sln(smin+smax*nn/N)[5];
a5 := sln(smin+smax*nn/N)[6];
a6 := sln(smin+smax*nn/N)[7];
AB := seq(plottools[line]([rhs(a1), rhs(a2), rhs(a3)], [rhs(a4), rhs(a5), rhs(a6)], color = black, thickness = 2), nn = 0 .. N);
AO := seq(plottools[line]([rhs(a1), rhs(a2), rhs(a3)], [-.5, -.75, .25], color = black, thickness = 2), nn = 0 .. N);
LL := seq(pointplot3d([rhs(a1), rhs(a2), rhs(a3)], symbolsize = 16, symbol = solidsphere, color = green), nn = 0 .. N);
LLL := seq(pointplot3d([rhs(a4), rhs(a5), rhs(a6)], symbolsize = 16, symbol = solidsphere, color = red), nn = 0 .. N);
P := pointplot3d([-.5, -.75, .25], symbolsize = 12, symbol = solidsphere, color = blue);
B := `~`[rhs](Matrix(`~`[sln](seq(i, i = smin .. smax+20*h, h)))(() .. (), 1 .. n+2));

```

```
m := ArrayNumElems(B[()] .. (), 1); m; k := 1; for j to n+1 do L[j] := B[()] .. (), k+j] end do:
S := [seq(seq(L[j][i], j = 1 .. 3), i = 1 .. m)]:
SS := [seq(seq(L[j][i], j = 4 .. 6), i = 1 .. m)]:
FACE := plottools[polygon]([seq([L[1][i], L[2][i], L[3][i]], i = 1 .. m)], color = RGB(12, .1, .5), transparency = .7, style = patchnogrid):
MS := pointplot3d(S, color = blue, style = line, thickness = 2, axes = normal, labels = ["x1", " x2", " x3"]):
MSS := pointplot3d(SS, color = RGB(.5, .1, .21), style = line, thickness = 1):
E := seq(display(MSS, MS, AB[k], AO[k], P, FACE, LL[k], LLL[k]), k = 1 .. N):
display(E, insequence = true, view = [-6 .. 2, -4 .. 4, -4 .. 4], axes = normal)
```