

ANIMATION OF THE VECTOR RELATIONS BY AUTOLISP FUNCTIONS

Dinel POPA¹, Nicolae-Doru STĂNESCU², Claudia-Mari POPA³

¹University of Pitești, Romania, dinel_popa@yahoo.com

²University of Pitești, Romania, s_doru@yahoo.com

³Armand Călinescu College, Pitești, Romania, claudia_mari_1966@yahoo.com

Abstract—In the paper are presented six AutoLisp functions by the aid of which one realizes the kinematic analysis of a jointed quadrilateral. The kinematic analysis bases on graph-analytical methods, the showed AutoLisp functions determining the velocities and accelerations of mechanism's points and elements, with no geometrical construction. This thing performs a great connection between the intuitive graphic methods and the vector work procedures of the CAD soft. As an application, is used a Chebyshev mechanism. By the aid of the described functions we obtain the 360 layers which have the name of crank angle: positions, velocities and accelerations of the elements. In a separate layer we get the curve described by a point of mechanism's shaft. The animations of the mechanism and also of the velocities and accelerations polygonals give thus a qualitative appreciation of the mechanism and its functionality too.

Keywords—animation, AutoCAD, AutoLisp, kinematic analysis, mechanism

I. INTRODUCTION

THE appearance and then the exponential development of the personal computers' production developed a real industry in the field of assisted design products, generically called CAD soft (Computer Aided Design). These soft products transformed classic project modalities in complex operations, performed in group, the time between the idea and the final product diminishing very much, all being realized on a competitive market. The products of the projection consists also a database, thing that eases any future activities. When we speak about a CAD system, we think at the three components: man, computer and database. Thus, practically, the CAD system represents the man-computer team, team which has to respect some rules.

The CAD products developed, in the first row, due to the high precision in calculation in drawing. Practically, the CAD soft precision is the same to that of the analytical computer assisted methods, but the advantage is that of the obtaining of graphical results, which are more suggestive.

Further on, our aid is to present an integrated collection of AutoLisp functions which realizes the kinematic and kineto-static analysis of mechanisms. In previous papers were realized applications with such functions. In [1] are presented two AutoLisp functions which determine the velocities of the RRR dyad. In the first row is highlighted the vector way in which we work and stock the entities in

AutoCAD. This mode is much closed to the vector relations used in the graph-analytical analysis of the mechanisms.

In [2] the AutoLisp functions are used for the construction of the curve of centers. The curve of centers, for a good precision, is constructed using hundreds of points. To obtain a point, one makes three or four steps which assume the intersection of two circles or a circle with a straight line. For this reason in this paper we present two AutoLisp functions that return the intersection points of two circles, or a circle with a straight line, with no intervention in the construction process.

Further on, we will exemplify in an application the mode of construction and use of the AutoLisp functions.

II. THE MECHANISM USED AS APPLICATION

The jointed quadrilateral mechanisms are the most used mechanisms in technique. This is, in the first row, because of inferior used joints (rotational joints) and of the simple methods used for the kinematic and kineto-static analysis. There is also a great diversity of methods of synthesis that permit the determinations of the dimensions of mechanism function of imposed positions or trajectories. Even if a study is not performed, probably the jointed quadrilateral mechanism is the most studied mechanism. The synthesis of such mechanisms was the base preoccupation of some mathematicians. The most known is the Russian one (1821 – 1894) which has remarkable realizations in the planar mechanisms' synthesis by approximate analytical methods. He exposed, in 1878 at the exposition in Paris and in 1893 at Chicago, mechanisms which imitate the walk of a horse and which has at their base the jointed quadrilateral mechanism.

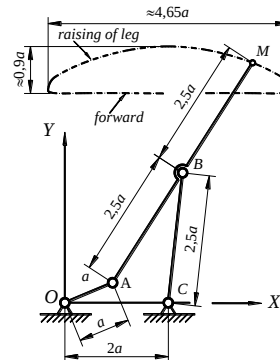


Fig. 1. Walking mechanism of Chebyshev type.

One such mechanism is presented in Fig. 1. Four such mechanisms are used to perform the walk. When point *M* describes a straight line, the body of the horse

forwards, and when point M describes an arc of curve, the horse raises the leg.

Further on, we purpose to perform the kinematic, kineto-static analysis, the animation of the mechanism, of the vector polygonals of velocities, accelerations and (in the future) forces, for such a category of mechanisms. In Fig. 2. is drawn the kinematic schema for such a mechanism for which are known:

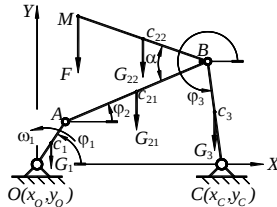


Fig. 2. Linked quadrilateral mechanism.

- dimensions of mechanism in meters: $OA=0.1$, $AB=0.25$, $BC=0.25$, $BM=0.25$,
- angle $\alpha=180^\circ$,
- positions of the linkages connected to basis $O(0; 0)$, $C(0.20; 0)$
- constant angular speed $\omega_1=10$ rad/s of the element 1,
- masses of the elements: $m_1=0.05$ kg, $m_{21}=0.125$ kg, $m_{22}=0.125$ kg, $m_3=0.125$ kg,
- force F that acts at the point M :

$$F = \begin{cases} 0 & \text{for } \varphi \in [0, 80^\circ] \\ 50 \text{ N} & \text{for } \varphi \in [80^\circ, 280^\circ] \\ 0 & \text{for } \varphi \in [280^\circ, 360^\circ] \end{cases}$$

The kinematic and kineto-static analysis is made with constant step of 1° , the angle φ_1 taking values in the interval $[1^\circ, 360^\circ]$.

III. AUTO LISP FUNCTIONS USED IN THE KINEMATIC ANALYSIS OF THE MECHANISM

Positional analysis by a graph method for the 360 positions of the active element imposes the use of a programming language. A solution would be the use of the script files. These are lists of AutoCAD instructions and they can be generated by a programming language. In this way, AutoCAD is used only as graph mode of representation. For this reason we will use AutoLisp and its facilities to obtain analytical solutions for the graph constructions.

When the position of point A is known, to obtain the point B by a classic graph analytical method, one intersects a circle of center at C and radius AB with a circle of center at C and radius BC . The construction is realized at a scale of lengths. The point A is known when is known the angle φ_1 .

In [2] was presented an AutoLisp function called "Int_2C" which returns the intersection points, if they exist, of two circles. In program is selected only the convenient solution. The content of the function is as follows:

```
(Defun Int_2C ()
```

```
(setq A (* 2 R1 (- a1 a2)) B (* 2 R1 (- b1 b2))
C (+ (* (- a1 a2) (- a1 a2)) (* (- b1 b2) (- b1
b2))
(* R1 R1) (* R2 R2 -1))
aa (+ (* A A) (* B B) (* C C -1)))
(If (>= aa 0)
(PROGN
(setq eroare 0)
(setq bb (/ (+ (* -1 B) (Sqrt aa)) (- C A)))
(If (> bb 0)
(setq fi11 (* 2 (atan bb)))
(setq fi11 (+ (* 2 Pi) (* 2 (atan bb))))
)
(setq bb (/ (- (* -1 B) (Sqrt aa)) (- C A)))
(If (> bb 0)
(setq fi12 (* 2 (atan bb)))
(setq fi12 (+ (* 2 Pi) (* 2 (atan bb))))
)
)
)
(setq eroare 1)
)
(If (= eroare 0)
(PROGN
(setq x1 (+ a1 (* R1 (cos fi11)))
y1 (+ b1 (* R1 (sin fi11)))
x2 (+ a1 (* R1 (cos fi12)))
y2 (+ b1 (* R1 (sin fi12)))
P1 (list x1 y1)
P2 (list x2 y2))
)
(Print "Cecurile nu se intersecteaza")
)
)
```

The function is without parameters and local variables and it is called inside other AutoLisp function. The centers of the circles are denoted by $O_1(a_1, b_1)$, $O_2(a_2, b_2)$ and the radii are R_1 and R_2 , respectively. We denote by $P_{1,2}$ the intersection points, the algorithm of the function being as follows:

- one projects the vector equation

$$\overline{OO_1} + \overline{O_1P} = \overline{OO_2} + \overline{O_2P} \quad (1)$$

on the system of axes resulting an equation of the form

$$A \cos \varphi + B \sin \varphi + C = 0, \quad (2)$$

where by φ was denoted the position angle made by the vector $\overline{O_1P}$ with the horizontal. The solution has the form

$$\varphi_{1,2} = 2 \arctan \frac{-B \pm \sqrt{A^2 + B^2 - C^2}}{C - A}. \quad (3)$$

One thus obtains, if there exist, the intersection points $P1(x1, y1)$ and $P2(x2, y2)$ of the two circle and assigns to the variable "eroare" the value 2 for two points of intersection and the value 1 in the case when the two circles do not intersect.

The AutoLisp functions can be directly edited from AutoCAD with the editor Visual Lisp (Tools » AutoLISP » Visual LISP editor). Usually, the AutoLisp functions can be found in the directory "Support" of AutoCAD and they have

the extension "*.lsp"; otherwise we have to find the show the path for the directory which contains the file. The function can be typed in any typing program which saves in ASCII code (Save As.. » simply text (*.txt) » MS-DOS » name_of_function.lsp), but Visual LISP Editor is preferred for its typing facilities. The call of the previous function is realized in an AutoLisp function called "**R_3R.lsp**" with the following contents:

```
(Defun C:R_3R ()
  (Setq Poz (Open "Pozitii.txt" "w"))
  (write-Line "fil,fi2,fi3,xB,yB,xM,yM" Poz)
  (Setq Vit (Open "Viteze.txt" "w"))
  (write-Line "fil,om2,om3,vB,vM" Vit)
  (Setq Acc (Open "Acceleratii.txt" "w"))
  (write-Line "fil,eps2,eps3,aB,aM" Acc)
  (Setq React (Open "Reactiuni.txt" "w"))
  (write-Line "fil,RO,RA,RB,RC,Mel" React)
  (Setq Curba (Open "Curba.scr" "w"))
  (Write-Line "Line" Curba)
  (setq xO 0.0 yO 0.0 xC 0.20 yC 0.0
    Omare (List xO yO) Cmare (List xC yC)
    lOA 0.10 lAB 0.25 lBC 0.25 lBM 0.25 alfaRad pi
    om1 10.0 imic (list 2 2) jmic (list 25 15))
  (Command "Erase" "All" "" "Osnap" "Off" "Ortho"
    "OFF")
  (Command "Zoom" "W" "-0.6,-0.2" "1,1")
  ; (Command "Zoom" "W" "0.5,1" "8,5")
  ; (Command "Zoom" "W" "-50,-15" "100,50")
  (setq fil -1)
  (While (< fil 359)
    (setq fil (+ fil 1) nr (+ fil 1000))
    (Command "-Layer" "N" nr "")
    (Command "-Layer" "S" nr "")
    (Setq filRad (* fil (/ Pi 180))
      Amare (Polar Omare filRad lOA)
      xA (princ (car Amare))
      yA (princ (car (cdr Amare))))
    (Setq a1 xA b1 yA R1 lAB
      a2 xC b2 yC R2 lBC)
    (Int_2C)
    (setq xB x1 yB y1 Bmare (List xB yB)
      fi2 (Angle Amare Bmare)
      fi3 (Angle Bmare Cmare))
    Mmare (Polar Bmare (- (+ fi2 pi) alfaRad) lBM)
    xM (princ (car Mmare))
    yM (princ (car (cdr Mmare)))
    text1 (Strcat (rtos xM 2 6) "," (Rtos yM 2 6)))
    (Write-Line text1 Curba)
    (Command "Pline" Omare "W" "0.001" "" Amare
      Bmare Cmare ""
      "Pline" Amare Mmare "")
    (Setq l 0.025)
    (Scriu Amare Mmare 65 77 32 32)
    (Scriu Cmare Bmare 67 66 32 32)
    (Command "Pline" "0,0" "W" "0.0005" ""
      "0.35,0" "W" "0.01" "0" "L" "0.035" "")
    (Command "Text" "0.4,-0.012" 1 "0" "X" "")
    (Command "Pline" "0,0" "W" "0.0005" ""
      "0,0.5" "W" "0.01" "0" "L" "0.035" "")
    (Command "Text" "-0.012,0.55" 1 "0" "Y" "")
    (Setq vA (* om1 lOA) vc 0.0 alfav (+ (/ Pi 2)
      filRad) betav 0.0)
    (Viteze)
    (setq accA (* om1 om1 lOA) accC 0.0
      alfaa (+ Pi FilRad) betaa 0)
    (Acceleratii)
    (Setq text2 (Strcat (rtos fil 2 6) "," (angtos
      fi2 0 6) "," (angtos fi3 0 6) ","
      rtos xB 2 6) "," (rtos yB 2 6) "," (rtos xM
      2 6) "," (rtos yM 2 6)))
    (Write-Line Text2 Poz)
    (Setq text3 (Strcat (rtos fil 2 6) "," (rtos
      om2 2 8) "," (rtos om3 2 8) ","
```

```
(rtos vitb 2 8) "," (rtos vitM 2 8)))
  (Write-Line Text3 Vit)
  (Setq text4 (Strcat (rtos fil 2 6) "," (rtos
    eps2 2 8) "," (rtos eps3 2 8) ","
    (rtos accb 2 8) "," (rtos accM 2 8)))
  (Write-Line Text4 Acc)
  (Command "-Layer" "Off" nr "Y" "")
  (Reactiuni)
  (Setq text5 (Strcat (rtos fil 2 6) "," (rtos
    RO 2 8) "," (rtos RA 2 8) ","
    (rtos RB 2 8) "," (rtos RC 2 8) "," (rtos
    Mel 2 8)))
  (Write-Line Text5 React)
)
(Write-Line "C" Curba)
(Close Curba)
(Close Poz) (Close Vit) (Close Acc)
(Close React)
)
```

The value obtained after the kinematic and kineto-static analyses are written in the following four files called: "**Pozitii.txt**", "**Viteze.txt**", "**Acceleratii.txt**" and "**Reactiuni.txt**", exactly as at an analytical assisted classic method.

The AutoLisp function begins with the forming of the four files and writing in them the head of the table. We also use the fifth file called "**Curba.scr**", which will contain the values of the trajectory described by point *M*.

After declaring the values assigned to: elements, joints linked to basis and angular speed ω_1 of the active element, in a **while** loop one varies the value of angle ϕ_1 in the interval $[0, 360]$, with the angular step of 1° .

To obtain all the mechanism's positions in one draw, the constructions are obtained, for each angle, in a layer named with the angle's value. To successively obtain the layers, the counting will starts from 1000. Thus, the first layer will contain: the position of mechanism, the velocities, accelerations and forces polygonals at $\phi_1 = 0^\circ$, it will be called 1000, while the last layer, 1359, will contain the same constructions, but for the angle $\phi_1 = 359^\circ$. After the constructions are realized in layers, these close to no visually superpose all the geometric constructions. The layers open when we cote or list. No matter the status in which the layer is (open/closed, frozen/unfrozen, visible/invisible etc.), when is saved the file with the current draw, all the entities are saved.

Knowing the position of point *A*, we obtain the two solutions for the point *B* by intersecting the circle of center $A(x_A, y_A)$ and radius *AB*, with the circle of center $C(x_C, y_C)$ and radius *BC*, by the aid of the AutoLisp function **Int_CD**. One retains the first solution and determines the angles ϕ_2 and ϕ_3 using the function **Angle**. Hence, to angle **fi2** we assign the angle made by the straight line *AB* with the *OX*-axis. The angle is determined with the function **Angle** specifying the end points (**Amare** and **Bmare**) in trigonometric sense. Analogically, to the angle **fi3** we assign the angle made by the straight line *BC* with the horizontal. The point *M* is obtained with the function **Polar**, starting from the

point B with the angle $\varphi_2 + \pi - \alpha$ and the distance BM .

The coordinates of the point $M(x_M, y_M)$ are written in the script file: "**Curba.scr**", which is run using the AutoCAD command SCRIPT, in layer 0. The construction in Fig. 3. appears, overwritten on the mechanism's position, using line with width and the coordinate axes.

To mark the joints of the mechanism, we write an AutoLisp function named "**Scriu**", with the following contents:

```
(Defun Scriu (Cap Coada nr_cap nr_coada semn_cap
semn_coada)
  (Command "Text" (Polar cap (Angle coada Cap) 1)
  1 "0" (Strcat (CHR nr_cap) (Chr semn_cap)))
  (Command "Text" (Polar Coada (Angle Cap Coada)
  1) 1 "0" (Strcat (CHR nr_coada) (Chr
semn_coada)))
)
```

The function has as parameters the coordinates of the points (**Cap** and **Coada**), at which we wish to type the text. The next two parameters are for the ASCII codes of the text letters (integer numbers) and the last two parameters for the second symbol of text, when it appears. If there is no second symbol, then one uses the number 32, the code of the key space. We used two symbols for the notations, taking into account the use of the same function for the points of the acceleration polygonals, where we used the symbol ' after the character (a' , b' , m' etc.). For instance, for the notations of joints of the linkages of element BC one calls the function with the coordinates of point B , the coordinates of point C , the number 66 (the code of the key B), the number (the code of the key C) and twice the number 32 for two spaces. The characters are written at a distance l from the element.

After the written of the characters A , M , C and B one obtains, for the angle $\varphi_1 = 70^\circ$, the representation in Fig. 3., after the selection of layer 1070 which is made visible too. For the appearance of the curve described by point M we run, in layer 0, the script file mentioned above.

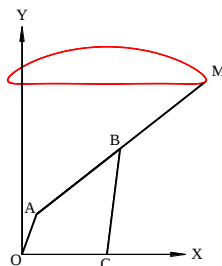


Fig. 3. The position of the linked quadrilateral mechanism at $\varphi_1 = 70^\circ$ and the curve described by point M .

To determine the velocity of the point B we use the Euler relation:

$$\vec{v}_B = \underbrace{\vec{v}_A}_{\perp OA} + \underbrace{\vec{v}_{BA}}_{\perp AB} = \underbrace{\vec{v}_C}_0 + \underbrace{\vec{v}_{BC}}_{\perp BC} \quad (4)$$

In (4) one knows: the speed of point A ($v_A = \omega_1 OA$), the speed of point C ($v_C = 0$) and the directions of the velocities $\vec{v}_{BC}$ and $\vec{v}_{BA}$. Using a graph-analytical method, one represents the vector relation (1), at the speeds scale, point b from the velocities plane resulting at the intersection of the perpendicular to AB and passing through a , (ia is the vector $\vec{v}_A$ at the scale of velocities) and the perpendicular to BC and passing through the pole i of velocities (Fig. 4.). To determine the velocities we created an AutoLisp function called "**Viteze**". It determines the velocities of the RRR dyad when are known the position of the dyad (points A , B and C) and the speeds of the poles A and C . The speeds are known by their magnitude and direction made by the velocity vector and the OX -axis. Thus, for the velocity of point A , the magnitude is $\omega_1 OA$ and the angle of the velocity is $\alpha_V = \varphi_1 + \pi/2$, while the velocity of point C is null and the angle of it and the horizontal is $\beta_V = 0$. The contents of the function "**Viteze**" is:

```
(DEFUN Viteze ()
  (Setq fi2 (angle Amare Bmare)
  fi3 (angle Bmare Cmare)
  amic (Polar imic alfav va)
  rmic (Polar amic (+ fi2 (/ Pi 2)) 1)
  cmic (Polar imic betav vc)
  smic (Polar cmic (+ fi3 (/ Pi 2)) 1)
  bmic (Inters amic rmic cmic smic Nil)
  pctmmic (Polar bmic (+ (Angle bmic amic)
alfaRad) (* (Distance amic bmic) (/ lBM lAB)))
  vitb (Distance imic bmic)
  vitm (Distance imic pctmmic)
  om2 (/ (distance amic bmic) (distance Amare
Bmare))
  om3 (/ (distance bmic cmic) (distance Bmare
Cmare)))
  (Command "Line" imic amic bmic imic cmic bmic
pctmmic "")
  (Command "line" imic pctmmic "")
  (Setq l (/ va 10) g (* 1 0.353))
  (Sageti amic imic) (Sageti pctmmic imic)
  (Sageti bmic imic) (Sageti bmic amic)
  (Sageti pctmmic bmic)
  (Scriu amic pctmmic 97 109 32 32)
  (Scriu imic bmic 105 98 32 32)
)
```

The function is without parameters and local variables. First of all, one determines the positional angles of the elements AB and BC of dyad (φ_2 and φ_3 , Fig. 2.), with the function of multiple assignment **Setq**. We obtain the point a , starting from the pole i of velocities with the angle α_V and distance v_A , by the aid of the AutoLisp function **Polar** (Fig. 4.). Using the same function we determine an arbitrary point r , belonging to the perpendicular to AB , starting polar from the point a with the angle $\varphi_2 + \pi/2$ and an arbitrary distance (we have chosen the value 1). Analogically, one determines the point c and then the point s that belongs to the perpendicular to BC (we start polar from c with the angle $\varphi_3 + \pi/2$ and the arbitrary distance 1).

With the function **Inters** one determines the intersection point, denoted by b , of the segments ar and cs (Fig. 4.). The function **Inters** analyzes two straight lines by their end points and returns, if exists, their intersection point. The distance between i and b is determined with the function **Distance** and it represents the magnitude of the velocity of point b . Thus, we analytically determine the magnitude of the velocity of point B , using a graph method making no geometric construction. The velocity of point M (point m in the velocities plane) is determined using the theorem of similarity. The degenerated triangle ABM is similar to triangle abm . The point m is obtained polar starting **Polar** from b with the angle made by ab with the horizontal at which we add the angle α and the distance $bm = ab \frac{BM}{AB}$. The distance between i and m determined with the function **Distance** represents the magnitude of the speed of point m . The angular speeds of the elements 2 and 3 (ω_2 and ω_3) determine as the ratios between ab and AB , and bc and BC , respectively ($\omega_2 = ab/AB$, $\omega_3 = bc/BC$). If one wishes a graphic representation, one uses the function **Command**, with the aid of which we call AutoCAD functions, drawing lines between the obtained points. The geometric figure completes with arrows that indicates the vectors sense, with the function "**Sageti**":

```
(Defun Sageti (cap coada)
  (Command "Pline" cap "W" "0" g (Polar cap
    (angle cap coada) 1) ""))
)
```

Because the arrows have always the same shape, we have chosen this function to avoid the repetition of the same expression, but with other head and tail points of the arrows. Finally, one completes the figure with notations, using the function "**Scriu**" which was described before. The call of the function "**Viteze**" in the function "**R_3R**" will lead to the obtaining of 360 layers which contain the velocities polygonals. For $\varphi_1 = 70^\circ$ one gets the velocities polygonal in Fig. 4.

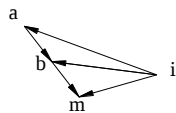


Fig. 4. The velocities polygonal at $\varphi_1 = 70^\circ$.

To obtain the accelerations one calls the Rivals relation:

$$\bar{a}_B = \frac{\bar{a}_A}{A \rightarrow O} + \frac{\bar{a}_{BA}^n}{B \rightarrow A \perp AB} + \frac{\bar{a}_{BA}^t}{0} = \frac{\bar{a}_C}{0} + \frac{\bar{a}_{BC}^n}{B \rightarrow C \perp BC} + \frac{\bar{a}_{BC}^t}{0} \quad (5)$$

We know $a_A = \omega_1^2 OA$, $a_{BA}^n = \omega_2^2 AB$, $a_C = 0$ and $a_{BC}^n = \omega_3^2 BC$. From the accelerations polygonal one determines the point b' at the intersection of the

perpendicular to AB with the perpendicular to BC . As in the previous case we wrote an AutoLisp function to determine the accelerations of the interior point B and of exterior pole M . One knows the parameters of the enter poles A and C by the magnitudes: $a_A = \omega_1^2 OA$ and $a_C = 0$, and the angles of these accelerations with the horizontal $\alpha_A = \varphi_1 + \pi$, respectively $\alpha_C = 0$. The function is called "**Acceleratii**" and its content is as follows:

```
(DEFUN Acceleratii ()
  (Setq aprim(Polar jmic alfaa acca)
    mprim(Polar aprim (+ fi2 Pi) (* om2 om2
      (distance Amare Bmare)))
  (distance Amare Bmare)))
  mmic(Polar mprim (+ fi2 (/ Pi 2)) 10)
  cprim(Polar jmic betaa accc)
  nprim(Polar cprim fi3 (* om3 om3 (distance
    Bmare Cmare)))
  nmic(Polar nprim (+ fi3 (/ Pi 2)) 10)
  bprim(Inters mprim mmic nprim nmic Nil)
  pctmprim(Polar bprim (+ (Angle bprim aprim)
    alfaRad) (* (Distance aprim bprim) (/ lBM lAB)))
  accb(Distance jmic bprim)
  accM(Distance jmic pctmprim)
  eps2(/ (distance mprim bprim) (distance Amare
    Bmare))
  eps3(/ (distance nprim bprim) (distance Bmare
    Cmare)))
  (Command "Line" jmic aprim mprim bprim jmic
    cprim nprim bprim pctmprim "")
  (command "line" jmic pctmprim "")
  (Setq l(/ acca 10) g(* l 0.353))
  (Sageti aprim jmic)
  (Sageti pctmprim jmic)
  (Sageti bprim jmic) (Sageti bprim mprim)
  (Sageti bprim nprim) (Sageti mprim aprim)
  (Sageti nprim cprim) (Sageti pctmprim bprim)
  (Scriu jmic bprim 106 98 32 39)
  (Scriu aprim pctmprim 97 109 39 39)
)
```

As in the previous case the function is without parameter and local variables. From the pole j of accelerations one obtains the point a' starting **Polar** from j with the angle α_A made by the acceleration a_A with the horizontal and with the distance a_A (the magnitude of the acceleration). One continues with the determination of the point m' (Fig. 5.), starting polar from a' with the angle $\varphi_2 + \pi$ and the distance $\omega_2^2 AB$ ($(* om2 om2 (distance Amare Bmare))$). Further on, one uses the previous point to construct a perpendicular to AB : starting polar from m' with the angle $\varphi_2 + \pi/2$, with an arbitrary distance ($(Polar mprim (+ fi2 (/ Pi 2)) 10)$). According to the vector relation (5), we construct from j the acceleration a_C (the segment jc') which is zero in this case and then the normal acceleration a_{BC}^n , starting polar from the point c' , previously determined, with the angle φ_3 and the distance $\omega_3^2 BC$ (the segment $c'n'$ in Fig. 5.). To construct the perpendicular to BC one starts polar from n' , with the angle $\varphi_3 + \pi/2$ and the arbitrary distance 10. It thus results the point **sprim**.

With the aid of the function **Inters** one obtains the point b' , by intersecting the segments $m'r'$ and $n's'$. The distance between j and b' represents the magnitude of the acceleration of point B (**accb(Distance j bprim)**). The acceleration of the point M (the point m' in the accelerations plane) is determined with the theorem of similarity. The degenerated triangle ABM is similar to the triangle $a'b'm'$. The point m' is obtained starting **Polar** from b' with the angle made by $a'b'$ with the horizontal at which is added the angle α and the distance $b'm' = a'b' \frac{BM}{AB}$. The distance between j and m' is determined with the function **Distance** and it represents the magnitude of acceleration of the point M .

The angular accelerations of the elements 2 and 3 (ε_2 and ε_3) are determined as the ratios between the distances $m'b'$ and AB , $b'n'$ and BC , respectively ($\varepsilon_2 = \frac{m'b'}{AB}$, $\varepsilon_3 = \frac{b'n'}{BC}$).

With the aid of the function **COMMAND** one calls the AutoCAD command **LINE** to construct the contour $ja'm'b'jn'b'$.

To complete the acceleration polygonal with arrows one uses the function **"Sageti"**, and with the function **"Scriu"** are positioned the letters of the accelerations polygonal. The calling of the function **"Acceleratii"** in function **"R_3R"** will lead to the obtaining of 360 layers which contain the accelerations polygonals. In layer 1070 there is the accelerations polygonal for the angle $\varphi_1 = 70^\circ$, represented in Fig. 5.

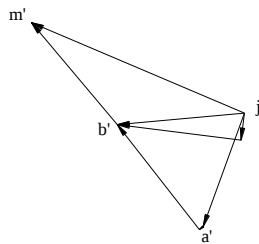


Fig. 5. The accelerations polygonals at $\varphi_1 = 70^\circ$.

At running of the function in AutoCAD, will appear the animation of the mechanism in the window declared for the position (corner left bottom **"-0.6,-0.2"**, corner right up **"1,1"**). After this one can visualize different layers to obtain the position and the velocities and accelerations polygonals for the 360 angles.

If one wishes, one can visualize the animation of the velocities polygonal, in a window of velocities visualization (**(Command "Zoom" "W" "0.5,1" "8,5")**) or of the accelerations polygonal in a window of accelerations visualization (**(Command "Zoom" "W" "-50,-15" "100,50")**).

The animation is not only a show, but it also gives more details about the magnitudes, senses and the rate of growth or diminishing of the vector parameters. Keeping into account that at the ends of the vectors is also

"linked" the letter which moves with the vector simultaneously, we realize the didactical goal of the function too. Modifying the dimensions of the mechanism one can make visual appreciations regarding the performed modifications.

The function **"R_3R"** also calls the function **"Reactiuni"**. With this new AutoLisp function we determine the reactions in the links of the mechanism. Because of the small space dedicated to this paper this function is not presented here. But it can be easily realized with the aid of the algorithm described in [6].

IV. CONCLUSIONS

The present paper highlighted the closing between the logic of the graph methods and the mathematical logic by which AutoLisp works with entities. The six AutoLisp functions presented may be used in the analysis and synthesis of the planar mechanisms. In the present paper they were applied at a Chebyshev type mechanism. We thus realized the kinematic analysis for 360 positions of the mechanism, each position belonging to a layer which has the name given by the angle of the crank. In a layer are presented not only the position of the mechanism, but also the velocities and accelerations polygonals. Even if these constructions were obtained using vector relations, which assumed a scale representation in the graphic classic models, in AutoLisp the obtaining of the numerical results was made without a representation of the vector polygonal. The geometric representation was realized only for a view of these vector parameters and their animation. Due to the fact that the letters accompany the vector one gives a better clarity to the representation, adding also a didactic goal.

To the actual function one may add a new function to determine the reactions in the linkages.

REFERENCES

- [1] D. Popa, M. Stan, C. Popa, "AutoLisp Procedures Used in Kinematic Analysis of Mechanisms", *The XXVII-th National Conference on Solids Mechanics*, Târgoviște 2004, pp. 132 – 136.
- [2] D. Popa, N. Pandrea, "The construction of the centers curve with AutoLisp functions", *The XXX-th National Conference on Solids mechanics*, Constanța Maritime University MECOSOL 2006.
- [3] I. Artobolevski, *Theory of Mechanisms and Machines (Théorie des mécanismes et des machines)*, Ed. MIR, Moscow, 1977.
- [4] Fl. Dudiță, *Mechanisms (Mecanisme)*, Universitatea din Brașov, 1977.
- [5] Fl. Dudiță, D. Diaconescu, *Structural Optimisation of Mechanisms (Optimizarea structurală a mecanismelor)*, Ed. Tehnică, București, 1987.
- [6] V. Handra Luca, I. A. Stoica, *Introduction to the Theory of Mechanisms (Introducere în teoria mecanismelor)*, Ed. Dacia, Cluj-Napoca, 1983.
- [7] N. Pandrea, E. Bărașcu, *Mechanisms (Mecanisme)*, Universitatea din Pitești, 1990.
- [8] Chr. Pelecudi, D. Maroș, V. Merticaru, N. Pandrea, I. Simionescu, *Mechanisms (Mecanisme)*, Ed. Didactică și Pedagogică, București, 1985.
- [9] N. Manolescu, Fr. Kovacs, A. Orănescu, *Theory of Mechanisms and Machines (Teoria mecanismelor și mașinilor)*, Editura Didactică și Pedagogică, București, 1972.
- [10] D. Manolea, *Programming in AutoLisp under AutoCAD (Programarea în AutoLisp sub AutoCAD)*, Editura Alabastră, Cluj-Napoca, 1996.