

MODELAREA CINEMATICĂ A UNUI MECANISM MONOMOBIL ÎN MEDII SPECIFICE DE SIMULARE (MATHCAD / MATLAB)

APOSTOL Ionuț¹, BOZEANU Ana Lucia², PRIOTEASA Andrada-Mădălina³ și
SOLOMON Maria-Georgiana⁴

Conducător științific: Prof.dr.ing. **Adriana COMĂNESCU**

REZUMAT: Prezenta lucrare este împărțită în 8 capitole. Primul capitol conține o scurtă introducere a proiectului, în care sunt prezentate obiectivele lucrării și rezultatele obținute. Cel de-al doilea capitol prezintă pe scurt cele 2 medii de simulate MathCAD și MATLAB, precum și schema cinematică a prototipului ales pentru analiză. De asemenea, acest capitol menționează modalitatea de determinare a traiectoriilor extremităților picioarelor gândacului în MathCAD. Al treilea capitol se axează pe contribuțiile originale din cadrul lucrării, și anume realizarea programului în MATLAB. Capitolul 4 prezintă rezultatele obținute în urma apelării programului. Capitolul 5 prezintă principalele diferențe dintre cele 2 programe. Capitolul 6 conține concluziile analizei. Capitolul 7 este dedicat mulțumirilor adresate doamnei profesor coordonator. Capitolele 8 și 9 cuprind bibliografia și notațiile utilizate în cadrul cercetării.

CUVINTE CHEIE: MathCAD, MATLAB, mecanism monomobil, simulare, modelare

1 INTRODUCERE

Scopul lucrării îl reprezintă modelarea cinematică a unui sistem monomobil în medii specifice de simulare, pornind de la lucrarea cu titlul "Gândac" din cadrul laboratorului ASSMM. Acest program își propune să determine traiectoriile pentru punctele extreme ale picioarelor gândacului ales. Pentru a atinge acest scop s-a realizat un program modularizat în mediul de simulare MATLAB care are ca finalitate prezentarea traiectoriilor menționate anterior.

2 STADIUL ACTUAL

Cele doua medii de simulare pe care le analizăm, prin prisma proiectului ales sunt MATHCAD și MATLAB. MATHCAD este un software orientat pe document, conținând un mediu de calcul extrem de puternic, care permite crearea unor documente complexe, într-un format user-friendly. MATLAB este un software care conține un pachet de programe dedicate calculului numeric. Pentru programul ce va fi prezentat în capitolele următoare MATLAB integrează rezolvarea unor vectori de funcții neliniare și vizualizarea grafică a soluțiilor obținute.

¹ Specializarea Ingineria și Managementul Sistemelor Tehnologice, Facultatea IMST;

E-mail: ionut_ap@yahoo.com;

E-mail: anabozeanu@gmail.com

E-mail: andrada.prioteasa@yahoo.com;

E-mail: georgianna.solomon@yahoo.com;

2.1. Program realizat în MATHCAD

Pentru realizarea acestei lucrări în mediul de simulare MathCad sunt necesare următoarele etape:

- inițializarea parametrilor geometrici constanți, pe care îi vom prelua din schema cinematică
- inițializarea parametrilor independenți
- calculul parametrilor punctului B
- calcularea diadei RRR(2,3)
- calcularea diadei RRR(5,4)
- calcularea diadei RRR(6,7)
- parametrii punctelor T3, T5 și T7 asociați
- simularea traiectoriilor extremităților picioarelor gândacului.

2.1.1 Schema cinematică a mecanismului monomobil ales

Calculul traiectoriilor și descrierea funcțiilor neliniare au fost realizate pornind de la schema cinematică a prototipului de gândac.

Schema cinematică a gândacului a fost realizată în AutoCAD, în varianta 2D, versiunea 2014. Schema cuprinde notațiile cu ajutorul cărora vom putea descrie ecuațiile matematice necesare determinării traiectoriilor extremităților picioarelor gândacului.

**MODELAREA CINEMATICĂ A UNUI MECANISM MONOMOBIL ÎN MEDII SPECIFICE DE SIMULARE
(MATHCAD / MATLAB)**

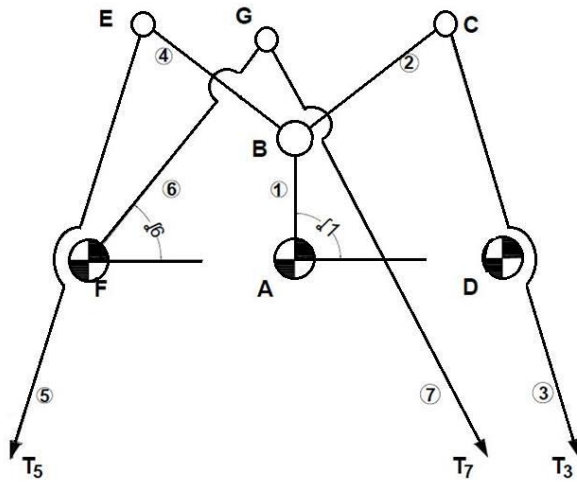


Fig. 1. Schemă cinematică gândac

Date specifice schemei cinemactice

- A(1,0)
- B(1,2)
- B(1,7)
- C(2,3)
- D(3,0)
- E(4,5)
- F(5,0)- baza
- F(6,0)
- G(6,7)
- $M=3 \cdot m - 2 \cdot i = 21 - 20 = 1$

2.1.2 Program MathCAD

Parametrii geometrici constanți utilizați în programul din MathCad sunt:

- $XA := 0$
- $YA := 0$
- $XD := 0.012$
- $YD := 0$
- $XF := -XD$
- $YF := YD$
- $AB := 0.002$
- $BC := 0.017$
- $DC := 0.007$
- $DT3 := 0.018$
- $BE := BC$
- $FE := DC$

$$FT5 := DT3 + 0.018$$

$$FG := 0.02$$

$$BG := 0.01$$

$$BT7 := 0.025 - 0.005$$

Parametrii independenți utilizați în program sunt:

$$k := 0..36$$

$$\phi_{1k} := k \cdot \frac{\pi}{18}$$

Calculul parametrilor punctului "B" în program se calculează cu ajutorul ecuațiilor:

$$XB_k := XA + AB \cdot \cos(\phi_{1k})$$

$$YB_k := YA + AB \cdot \sin(\phi_{1k})$$

Calculul diadei RRR(2,3) se realizează în MathCad astfel:

$$\phi_{20} := 10$$

$$\phi_{30} := 120$$

$$\phi_2 := \phi_{20} \cdot \frac{\pi}{180}$$

$$\phi_3 := \phi_{30} \cdot \frac{\pi}{180}$$

Given

$$XB_k + BC \cdot \cos(\phi_2) - XD - DC \cdot \cos(\phi_3) = 0$$

$$YB_k + BC \cdot \sin(\phi_2) - YD - DC \cdot \sin(\phi_3) = 0$$

$$sol_k := \text{Find}(\phi_2, \phi_3)$$

$$\begin{pmatrix} \phi_{2k} \\ \phi_{3k} \end{pmatrix} := sol_k$$

$$\begin{pmatrix} \phi_{20k} \\ \phi_{30k} \end{pmatrix} := sol_k \cdot \frac{180}{\pi}$$

Utilizând soluțiile obținute anterior se pot scrie ecuațiile care duc la aflarea parametrilor punctului T3:

$$XT_{3k} := XD + DT3 \cdot \cos(\phi_{30k} + \pi)$$

$$YT_{3k} := YD + DT3 \cdot \sin(\phi_{30k} + \pi)$$

Analog se calculează diadele RRR(5,4) și RRR(6,7), precum și parametrul punctelor T5 și T7.

3 PROCEDURI/ FUNCȚII/ SCRIPTURI MATLAB


3.1. Script principal (modalitate de apelare)

Pentru apelarea programului se va scrie în linia de comandă în ordine, după cum urmează:

```
>> param_geom_cst  
>> param_indep  
>> param_pct_B  
>> RRR23  
>> RRR54  
>> RRR67  
>> traiect
```


De asemenea, după fiecare grafic afișat, respectiv după fiecare animație, dacă sunt rulate toate scripturile deodată trebuie dat “space” pentru a se face trecerea la următorul grafic.

```
>> param_geom_cst
```



```
%% Parametrii geometrici constanti  
  
XA=0;  
YA=0;  
XD=0.012;  
YD=0;  
XF=-XD;  
YF=YD;  
AB=0.002;  
BC=0.017;  
DC=0.007;  
DT3=0.018;  
BE=BC;  
FE=DC;  
FT5=DT3+0.018;  
FG=0.02;  
BG=0.01;  
BT7=0.02;
```

```
>> param_indep
```



```
%% Parametru independent  
  
k=0:36;  
FI1(k+1)=k*(pi/18);
```

MODELAREA CINEMATICĂ A UNUI MECANISM MONOMOBIL ÎN MEDII SPECIFICE DE SIMULARE (MATHCAD / MATLAB)

>> param_pct_B



```

%% Parametrii punctului "B"

XB(k+1)= XA + AB*cos(FI1(k+1));
YB(k+1)= YA + AB*sin(FI1(k+1));
plot(k,XB,'r',k,YB,'b','LineWidth',3)
xlabel('k')
ylabel('XBk,YBk')
axis([0 40 -0.002 0.002])
set(gca,'YTick',[-0.002 -0.0016 -0.0012 -0.0008 -0.0004 0 0.0004 0.0008 0.0012 0.0016 0.0020])
grid on
w = waitforbuttonpress;
if w
figure
plot(XB,YB,'r','LineWidth',3)
axis([-0.002 0.002 -0.002 0.002])
set(gca,'YTick',[-0.002 -0.0016 -0.0012 -0.0008 -0.0004 0 0.0004 0.0008 0.0012 0.0016 0.0020])
set(gca,'XTick',[-0.002 -0.0016 -0.0012 -0.0008 -0.0004 0 0.0004 0.0008 0.0012 0.0016 0.0020])
grid on
hold on
for j=0:36
    h1=plot(XB(j+1),YB(j+1),'b o','LineWidth',4);
    xlabel('XBk')
    ylabel('YBk')
    pause(0.4);
    delete(h1)
end
end
    
```

>> RRR23



```

%% Diada RRR(2,3)
%Valori de initializare
FI20=10;
FI30=120;
FI2=FI20*pi/180;
FI3=FI30*pi/180;
for i = 2:k(end)+1
    FI23 = fsolve(@(x) fun_1(x,XB(i),BC,XD,DC,YB(i),YD), [FI2(i-1);FI3(i-1)]);
    FI2 = [FI2 ; FI23(1)];
    FI3 = [FI3 ; FI23(2)];
end
w = waitforbuttonpress;
if w
figure
plot(0:35,180/pi*FI2(2:end),'r',0:35,180/pi*FI3(2:end),'LineWidth',3)
ylabel('FI20k, FI30k')
xlabel('k')
axis([0 40 -50 100])
set(gca,'YTick',[-50 -35 -20 -5 10 25 40 55 70 85 100])
grid on
%parametrii punctului T3
XT3(k+1) = XD + DT3*cos(FI3(k+1)+pi);
YT3(k+1) = YD + DT3*sin(FI3(k+1)+pi);
end
w = waitforbuttonpress;
if w
figure
plot(0:35,XT3(2:end),'r',0:35,YT3(2:end),'b','LineWidth',3)
ylabel('XT3k, YT3k')
xlabel('k')
axis([0 40 -0.02 0.01])
grid on
end
    
```

```
function FI23 = fun_1(x,XBK,BC,XD,DC,YBK,YD)
```

```
FI23 =[XBK + BC*cos(x(1)) - XD - DC*cos(x(2)); YBK + BC*sin(x(1)) - YD - DC*sin(x(2))];
end
```

>> RRR54



```
%% Diada RRR(5,4)
```

```
FI50 = 30;
FI40 = 120;
FI5 = FI50*(pi/180);
FI4 = FI40*(pi/180);
for i = 2:k(end)+1
    FI54 = fsolve(@(x) fun_2(x,XF,FE,XB(i),BE,YF,YB(i)), [FI5(i-1);FI4(i-1)]);
```

```
    FI5 = [FI5 ; FI54(1)];
    FI4 = [FI4 ; FI54(2)];
```

```
end
```

```
w = waitforbuttonpress;
```

```
if w
```

```
figure
```

```
plot(0:35,180/pi*FI5(2:end), 'r', 0:35,180/pi*FI4(2:end), 'LineWidth', 3)
```

```
ylabel('FI50k, FI40k')
```

```
xlabel('k')
```

```
axis([0 40 -50 100])
```

```
set(gca, 'YTick', [-50 -35 -20 -5 10 25 40 55 70 85 100])
```

```
grid on
```

```
end
```

```
%Parametrii punctului T5
```

```
XT5(k+1) = XF + FT5*cos(FI5(k+1)+pi);
```

```
YT5(k+1) = YF + FT5*sin(FI5(k+1)+pi);
```

```
w = waitforbuttonpress;
```

```
if w
```

```
figure
```

```
plot(0:35,XT5(2:end), 'r', 0:35,YT5(2:end), 'b', 'LineWidth', 3)
```

```
ylabel('XT5k, YT5k')
```

```
xlabel('k')
```

```
axis([0 40 -0.2 0.2])
```

```
grid on
```

```
end
```

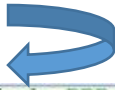
```
function FI54 = fun_2(x,XF,FE,XBK,BE,YF,YBK)
```

```
FI54 =[XF + FE*cos(x(1)) - XBK - BE*cos(x(2)); YF + FE*sin(x(1)) - YBK - BE*sin(x(2))];
```

```
end
```

MODELAREA CINEMATICĂ A UNUI MECANISM MONOMOBIL ÎN MEDII SPECIFICE DE SIMULARE
(MATHCAD / MATLAB)

>> RRR67



```

%% Diada RRR(6,7)

FI60 = 30;
FI70 = 120;
FI6 = FI60*(pi/180);
FI7 = FI70*(pi/180);
for i = 2:k(end)+1
    FI67 = fsolve(@(x) fun 3(x,XF,FG,XB(i),BG,YF,YB(i)), [FI6(i-1);FI7(i-1)]);
    FI6 = [FI6 ; FI67(1)];
    FI7 = [FI7 ; FI67(2)];
end
w = waitforbuttonpress;
if w
figure
plot(0:35,180/pi*FI6(2:end), 'r',0:35,180/pi*FI7(2:end), 'LineWidth',3)
ylabel('FI60k, FI70k')
xlabel('k')
axis([0 40 0 100])
set(gca,'YTick',[0 10 20 30 40 50 60 70 80 90 100])
grid on
end
%Parametrii punctului T7
XT7(k+1) = XB(k+1)' + BT7*cos((FI7(k+1)) + pi);
YT7(k+1) = YB(k+1)' + BT7*sin((FI7(k+1)) + pi);
w = waitforbuttonpress;
if w
figure
figure
plot(0:35,XT7(2:end), 'r',0:35,YT7(2:end), 'b', 'LineWidth',3)
ylabel('XT7k, YT7k')
xlabel('k')
axis([0 40 -0.025 0])
grid on
end

```

```
function FI67 = fun 3(x,XF,FG,XBK,BG,YF,YBK)
```

```

FI67 =[XF+ FG*cos(x(1))- XBK- BG*cos(x(2));YF+ FG*sin(x(1))- YBK- BG*sin(x(2))];
end

```

>> traieci



```

%%Traieci

%% Grafic final 1
w = waitforbuttonpress;
if w
figure
plot(XT3(2:end),YT3(2:end),'r',XT5(2:end),YT5(2:end),'g',XT7(2:end),YT7(2:end),'b','LineWidth',3)
ylabel('YT3k, YT5k, YT7k')
xlabel('XT3k, XT5k, XT7k')
axis([-0.05 0.01 -0.02 0.005])
set(gca,'YTick',[-0.02 -0.015 -0.01 -0.005 0 0.005])
grid on
hold on
for j=0:35
    h1=plot(XT3(j+2),YT3(j+2),'k o','LineWidth',4);
    h2=plot(XT5(j+2),YT5(j+2),'k o','LineWidth',4);
    h3=plot(XT7(j+2),YT7(j+2),'k o','LineWidth',4);
    pause(0.2);
    delete(h1)
    delete(h2)
    delete(h3)
end
end

%% Grafic final 2
%w = waitforbuttonpress;
%if w
figure
plot(0:35,YT3(2:end),'r',0:35,YT5(2:end),'g',0:35,YT7(2:end),'b','LineWidth',3)
ylabel('YT3k, YT5k, YT7k')
xlabel('k')
axis([0 40|-0.02 0.005])
set(gca,'YTick',[-0.02 -0.0175 -0.015 -0.0125 -0.01 -0.0075 -0.005 -0.0025 0 0.0025 0.005])
grid on
hold on

for j=0:35
    h1=plot(j,YT3(j+2),'k o','LineWidth',4);
    h2=plot(j,YT5(j+2),'k o','LineWidth',4);
    h3=plot(j,YT7(j+2),'k o','LineWidth',4);
    pause(0.2);
    delete(h1)
    delete(h2)
    delete(h3)
end
end

```

4 DETERMINAREA TRAIECTORIILOR PENTRU PUNCTELE EXTREME ALE PICIOARELOR GÂNDACULUI - MATLAB

4.1. Inițializare parametrii constanți

Pentru inițializarea parametrilor constanți s-au realizat 2 scripturi în MATLAB. Cele 2 scripturi se numesc param_geom_cst, respectiv param_indep.

Dupa executarea celor 2 scripturi, în workspace-ul MATLAB-ului se vor salva constantele pe care le vom utiliza în cadrul programului. Acest lucru se poate observa în figura nr. 2.

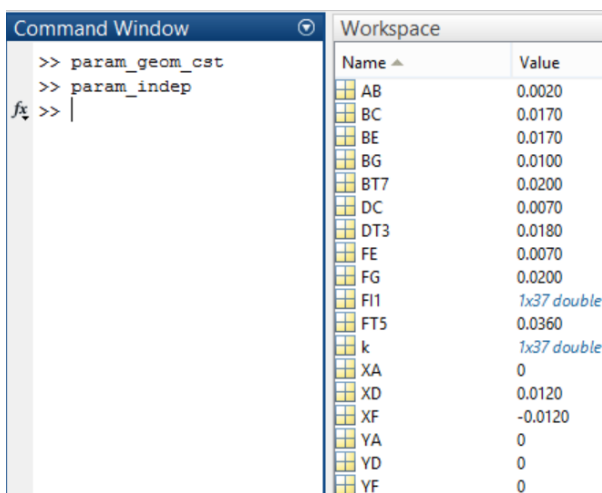


Fig. 2. Inițializare parametrii constanți

4.2. Parametrii punctului „B”

Pentru a determina și afișa grafic parametrii punctului „B” în funcție de parametrul independent „k”, este necesară rezolvarea unor vectori de funcții liniare. Soluțiile ecuației vor fi afișate în figura nr. 3

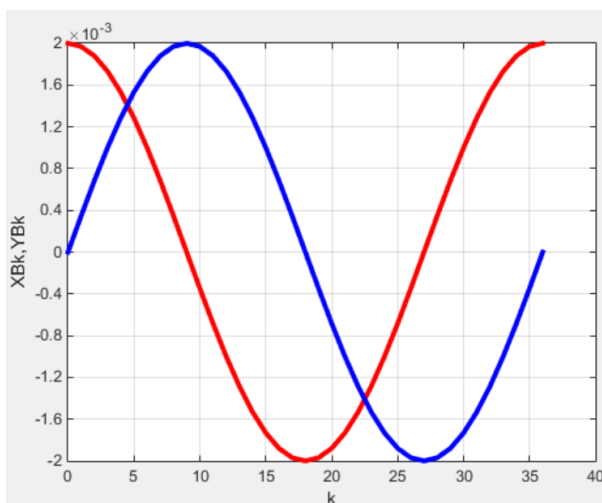


Fig. 3. Parametrii punctului „B”

De asemenea, acest script realizează un grafic animat, care evidențiază evoluția parametrilor YBk față de XBk. Acest lucru se poate observa în figura nr. 4.

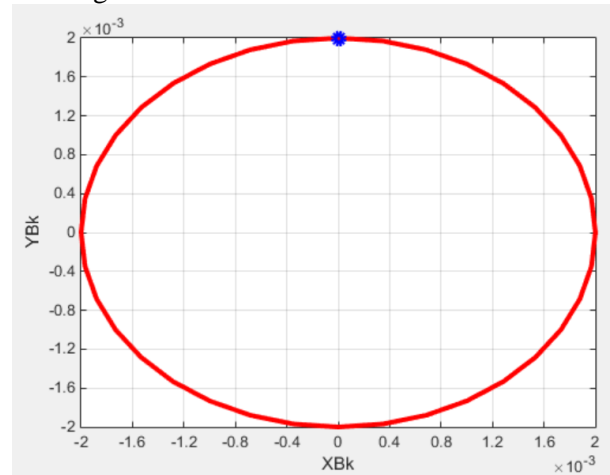


Fig. 4. YBk(XBk)

4.3. Diada RRR (2,3)

Pornind de la parametrii punctului „B” și de la unghiul constant ϕ_{10} se vor determina vectorii de soluții pentru ϕ_{20} și ϕ_{30} pe care îi vom utiliza pentru a calcula parametrii punctului „T3”. Punctul „T3” reprezintă una dintre cele 3 extremități ale picioarelor gândacului.

Pentru a determina vectorii de soluții ale unghiurilor ϕ_{20} și ϕ_{30} se va utiliza funcția prestabilită din MATLAB “fsolve”. Acești vectori care conțin valorile unghiurilor vor fi reprezentanți în figura 4, unde se va afișa variația lor în funcție de parametrul independent “k”. În figura 5 vor fi reprezentați grafic parametrii XT3k, YT3k, în funcție de parametrul independent “k”.

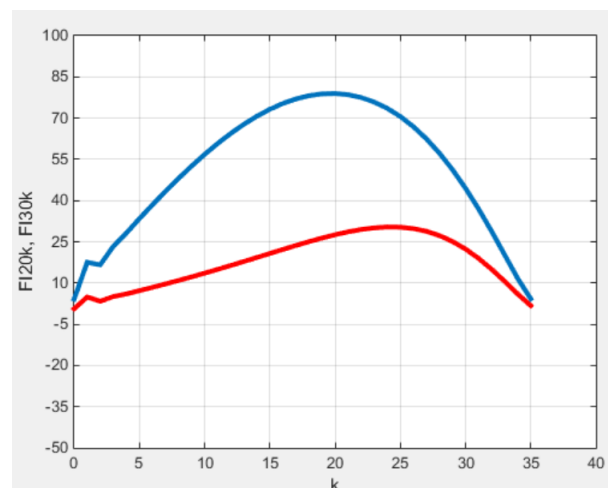


Fig. 4. Vectorii de soluții ale unghiurilor ϕ_{20} și ϕ_{30}

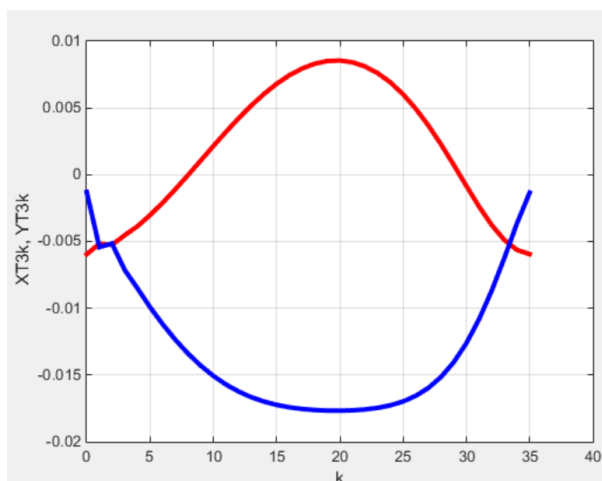


Fig. 5. Parametrii XT3k, YT3k

4.4.Diada RRR (5,4)

Folosind calculele anterioare se vor determina vectorii de soluții pentru ϕ_{50} și ϕ_{40} , pe care îi vom utiliza pentru a calcula parametrii punctului „T5”.

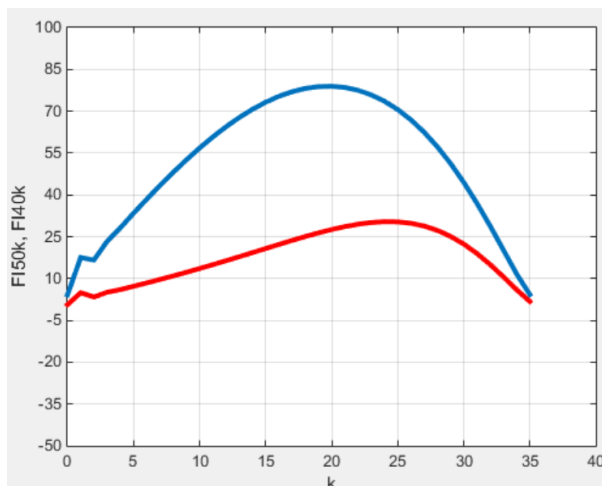


Fig. 6. Vectorii de soluții ale unghiurilor ϕ_{50} și ϕ_{40}

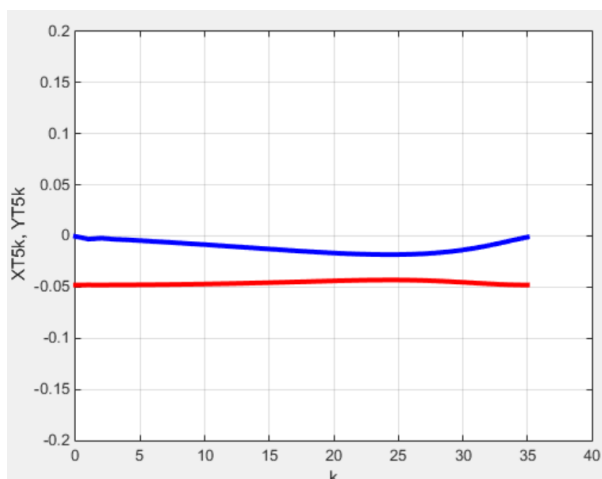


Fig. 7. Parametrii XT5k, YT5k

Punctul „T5” reprezintă cea de-a doua extremitate aferentă picioarelor gândacului. Pentru a determina vectorii de soluții ale unghiurilor ϕ_{50} și ϕ_{40} se va utiliza aceeași funcție menționată anterior:

“fsolve”. Acești vectori care conțin valorile unghiurilor sunt reprezentați în figura 6, unde este afișată variația lor în funcție de parametrul independent “k”. În figura 7 sunt reprezentați grafic parametrii XT5k, YT5k, în funcție de parametrul independent “k”.

4.5.Diada RRR (6,7)

Folosind calculele anterioare se vor determina vectorii de soluții pentru ϕ_{60} și ϕ_{70} pe care îi vom utiliza pentru a calcula parametrii punctului „T7”. Punctul „T7” reprezintă o altă extremitate a picioarelor gândacului.

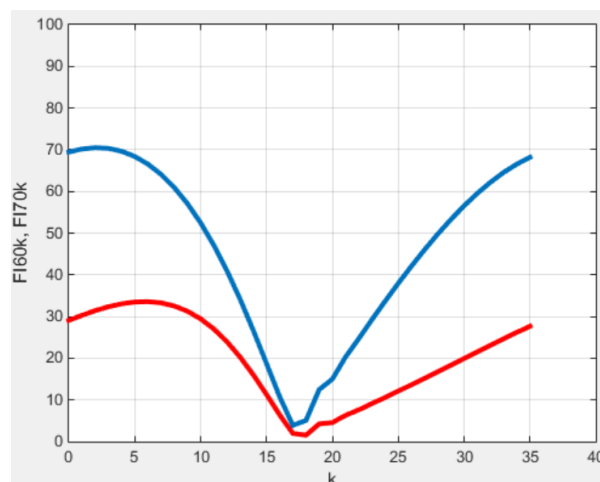


Fig. 8. Vectorii de soluții ale unghiurilor ϕ_{60} și ϕ_{70}

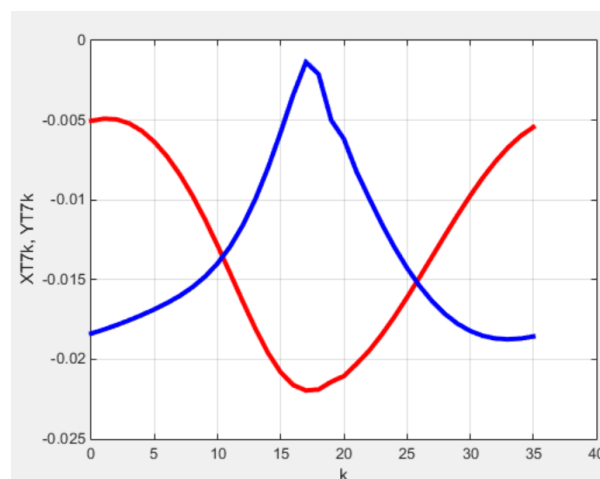


Fig. 9. Parametrii XT7k, YT7k

Pentru a determina vectorii de soluții ale unghiurilor ϕ_{60} și ϕ_{70} se va utiliza funcția „fsolve”. Acești vectori care conțin valorile unghiurilor sunt reprezentați în figura nr. 8, unde este afișată variația lor în funcție de parametrul independent “k”, iar în figura nr. 9 sunt reprezentați grafic parametrii XT7k, YT7k, în funcție de parametrul independent “k”.

MODELAREA CINEMATICĂ A UNUI MECANISM MONOMOBIL ÎN MEDII SPECIFICE DE SIMULARE (MATHCAD / MATLAB)

4.6. Traiectoriile pentru punctele extreme ale gândacului

În figurile nr. 10, respectiv nr.11, se vor prezenta variațiile parametrilor punctelor „T3”, „T5” și „T7”, lucru care oferă o viziune de ansamblu în ceea ce privește traiectoria gândacului. Această traiectorie este definită în funcție de traiectoriile fiecărei extremități în parte.

Pentru fiecare traiectorie în parte, în figura nr. 10, se va afișa evoluția parametrilor $YT3k$, $YT5k$, $YT7k$, în funcție de $XT3k$, $XT5k$, $XT7k$.

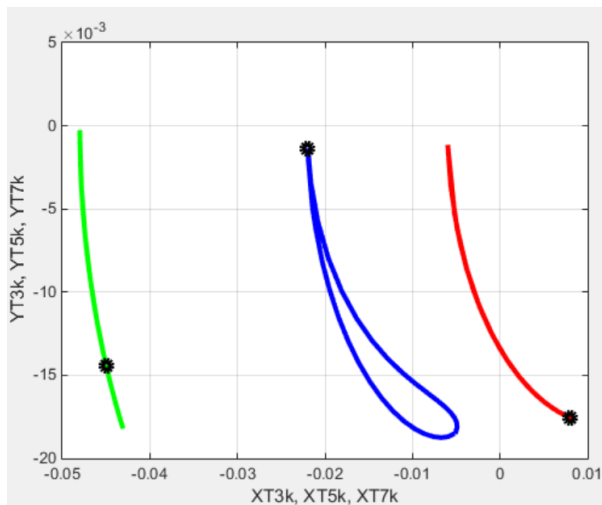


Fig. 10. $YT3k(XT3k)$, $YT5k(XT5k)$, $YT7k(XT7k)$

Figura nr. 11 conține variația parametrilor $YT3$, $YT5$ și $YT7$, în funcție de parametrul independent „k”. Aceste traiectorii se vor reprezenta printr-o animație realizată cu funcția „plot”.

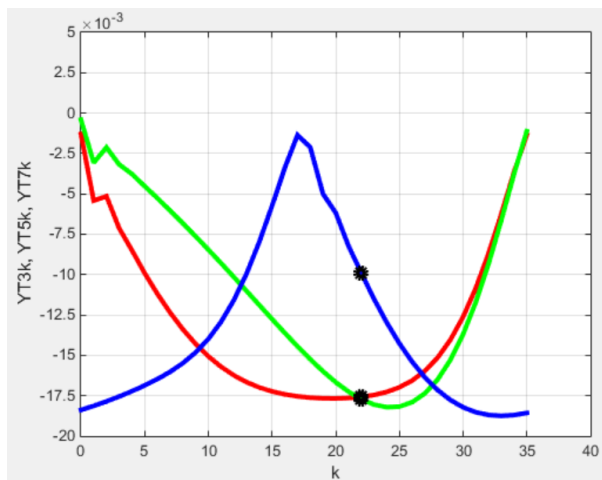


Fig. 11. Parametrii $YT3k$, $YT5k$, $YT7k$

În cele 2 figuri cu nr. 10, respectiv nr. 11, parametrul independent „k” crește cu o unitate, pornind de la valoarea 0, până la valoarea 36.

Aceste traiectorii se vor reprezenta printr-o animație realizată cu funcția plot, a cărei funcționalitate va fi prezentată în rândurile ce urmează.

Funcția „plot”, generează grafice 2D, cu scalare liniară a axelor. Aceasta funcție are diferite forme, în funcție de argumentele de intrare. Dacă de exemplu y este un vector, $plot(y)$ produce un grafic liniar al elementelor lui y versus indexul elementelor sale. Dacă se specifică doi vectori ca argumente, $plot(x,y)$ produce graficul lui y versus x . Se pot realiza grafice multiple utilizând un singur apel al funcției plot. MATLAB-ul realizează automat o reprezentare cu culori diferite pentru a permite distingerea graficelor. Se pot crea diferite tipuri de linii pentru fiecare set de date prin folosirea unor identificatori de tip string în funcția „plot”. Pentru plotarea datelor din matrici, atunci când funcția plot este utilizată cu un singur argument de tip matrice: $plot(Y)$, va fi realizat un grafic pentru fiecare coloana a matricii, cu axa x reprezentând indexul de linie $1:m$, cu m numărul liniilor din Y .

5 DIFERENȚE PRINCIPALE ÎNTRE CELE 2 PROGRAME SOFTWARE: MATLAB/MATHCAD

Principala diferență dintre cele 2 programe software este reprezentată de caracteristica de modularitate. În timp ce MathCAD-ul oferă o soluție care nu se bazează pe modularitate, MATLAB-ul ne oferă posibilitatea de a diviza programul principal în diferite funcții sau scripturi. Așadar, dacă în MATLAB putem scrie un program precum în MathCAD, reciproca nu este valabilă.

Pentru realizarea de animații, cele 2 programe funcționează diferit. Funcția „plot” se poate utiliza în MATLAB pentru a realiza reprezentări grafice, în timp ce în MathCAD graficele se pot configura direct prin intermediul Toolbox.

Mai mult decât atât, în MATLAB, prin suprapunerea consecutivă a unor grafice, se pot realiza scurte animații în vederea obținerii unei perspective de ansamblu în ceea ce privește traiectoriile pe care le descriu extremitățile picioarelor gândacului. În MathCAD, animațiile se pot realiza astfel: după selecția graficului se accesează opțiunea „View”-> „Animate”, urmând ca mai apoi să se configureze parametrii solicitați aferenți animației dorite.

O altă diferență observată pe parcursul lucrării se referă la faptul că ecuațiile de funcții neliniare se

rezolvă diferit în cele 2 programe. Pentru rezolvarea vectorilor de ecuații nelinare în MATLAB se folosește funcția “fsolve”, în timp ce, pentru rezolvarea acestora în MathCAD se folosește funcția “GIVEN”.

6 CONCLUZII

Concluzia principală care se poate trage din această redactare științifică este aceea că, MATLAB, fiind un software modularizat, permite ca pe viitor să i se aducă îmbunătățiri astfel încât traiectoria gândacului să se realizeze în funcție de diverși parametrii preluați din mediu. Acest lucru se poate realiza fără a schimba funcțiile principale.

Spre exemplu, se poate adăuga o funcție care să modifice traiectoria principală a gândacului, primind date externe de la senzori. O altă funcționalitate care se poate adăuga la acest program este controlul la distanță.

Prezenta lucrare descrie determinarea traiectoriilor picioarelor unui gândac, precum și simularea acestora pentru a ne oferi o imagine de ansamblu a funcționalității mecanismului monomobil ales.

7 MULȚUMIRI

Aducem mulțumiri doamnei Profesor dr. ing, Adriana COMĂNESCU, pentru îndrumarea și susținerea acordată, fără de care nu ar fi fost posibilă finalizarea și realizarea acestei lucrări la secțiunea următoare.

Dorim să mulțumim, de asemenea, domnului Conf. dr. ing Marin NEACȘA, domnului Conf. dr. ing. Iulian TABĂRĂ, domnului Profesor dr. ing. Constantin OCNĂRESCU și doamnei Ș.l.dr.ing. Ileana DUGĂEȘESCU, – pentru atenția acordată, dar și pentru susținerea oferită pe tot parcursul anului universitar.

8 BIBLIOGRAFIE

- [1]. Mircea Neagoe, (2002), *Cinematica roboților industriali: modelare și simulare*, Editura Universitatii "Transilvania", ISBN 9736350207, 9789736350207, 267 pagini.
- [2]. Camelia Gavrița, Viorel Petreș, Narcisa Teodorescu, Cristina Nartea, Iuliana Popescu, Alina Elisabeta Sandu, (2014), *MATHCAD-Aplicații modelare și simulare*, Editura Conspress, ISBN 978-973-100-356-6, 220 pagini.
- [3]. Zahariea Dănuț, (2014), *MATLAB-calcul numeric și simbolic*, Editura PIM, ISBN 97860661320912, 554 pagini.
- [4]. Alexandru P. Vișa, I., Alexandru Talabă, *Proiectarea funcțională a mecanismelor*, Editura Lux Libris, Brașov, 1999.

[5]. <http://civile.utcb.ro/cmat/cursrt/mmath.pdf>
[Accesat la data de 01/05/2015]

[6]. <http://www.scribub.com/stiinta/matematica/GRAFICE-SI-INTERFETE-GRAFICE-I91363.php>

[Accesat la data de 05/05/2015]

[7]. Moser, H. C. (2004). *Trends in EDM*, disponibil la: <http://www.mmsonline.com/articles/020001.html>

[Accesat la data de 01/05/2015]

9 NOTAȚII

În prezenta lucrare s-au utilizat următoarele notații:

m= număr de elemente mobile

i= număr de cuple inferioare