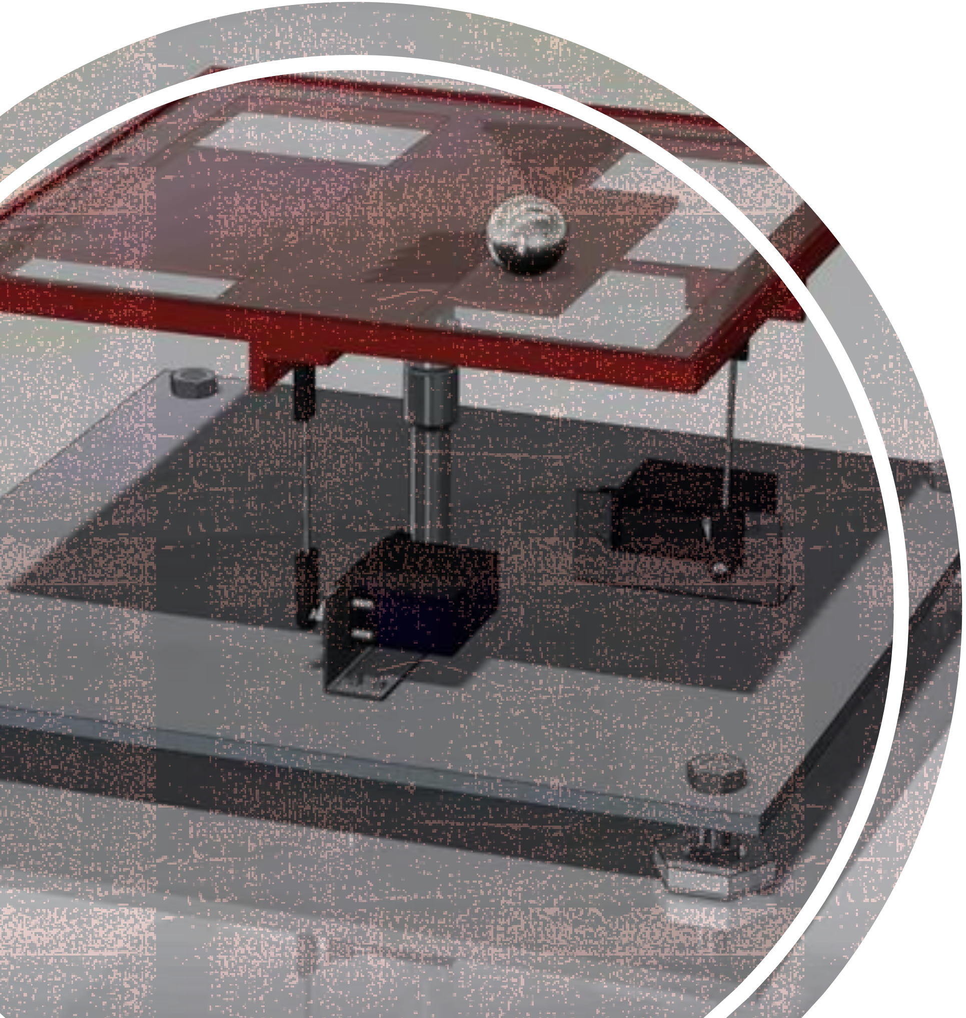
BALL ON A PLATE -MPC-

Pîndichi Elena
Stăvărache Robert - Antonio

CE ESTE MPC?

1



MPC – MODEL PREDICTIVE CONTROL

- -> metoda de control
- -> cautarea comenzii optime pe un orizont finit (orizont de predictie)
- -> minimizarea erorii & magnitudinii comenzii
- -> aplicarea primei comenzi din setul gasit
- -> reluarea procedurii

CE VREM SĂ FACEM ?

Q, R – matrici pondere

N – orizontul de predictive

i – iteratia MPC

$$J_{MPC} = \sum_{k=1}^{N} \left(\|\hat{y}(k+i) - y_{ref}(k+i)\|_Q^2 + \|u(k+i-1)\|_R^2 \right)$$

CUM FACEM?

- Modelarea sistemului
- Liniarizarea ecuatiilor
- Realizarea pe stare
- Discretizarea sistemului
- Prelucrarea sistemului discret sub forma unei constrangeri matriciale

ECUAȚIILE PENTRU FIECARE AXĂ

$$\left[M_b + \left(\frac{I_b}{r_b^2}\right)\right]\frac{\partial^2 x}{\partial t^2} - m_b\left[x_b\dot{\alpha}^2 + y_b\dot{\alpha}\dot{\beta}\right] + m_b g \sin\alpha = 0$$

$$\left[M_b + \left(\frac{I_b}{r_b^2}\right)\right]\frac{\partial^2 y}{\partial t^2} - m_b\left[y_b\dot{\beta}^2 + y_b\dot{\alpha}\dot{\beta}\right] + m_b g \sin\beta = 0$$

Mb –masa bilei
Ib – momentul de inertie al bilei
rb – raza bilei

$\dot{\alpha} \ll 0$ si $\dot{\beta} \ll 0$

Iar $\alpha \ll 1$ $\beta \ll 1$

Intrucat se pot utiliza aproximarile

$$\sin\alpha \simeq \alpha, \ \sin\beta \simeq \beta$$

Ceea ce conduce la sistemul liniar:

$$\frac{5}{7}\ddot{x}_b + g\alpha = 0$$

$$\frac{5}{7}\ddot{y}_b + g\beta = 0$$

REALIZAREA PE STARE

$$\dot{x}(t) = A_c x(t) + B_c u(t),$$
$$y(t) = C_c x(t),$$

Ac =

0	1
0	0

Bc =

0
0.7143

- Ceea ce conduce la sistemul discret (dupa esantionarea de pas Ts = 0.05)

$$x(k+1) = Ax(k) + Bu(k),$$
$$y(k) = Cx(k),$$

Ad =

1.0000	0.0500
0	1.0000

Bd =

0
0.0357

IMPLEMENTAREA OPTIMIZARII 2

OBTINEREA CONSTRANGERILOR

$$Y_1 = AY_0 + Bu_0$$

$$Y_2 = AY_1 + Bu_1 = A^2 Y_0 + ABu_0 + Bu_1$$

$$\vdots$$

$$Y_m = A^m Y_0 + \sum_{k=0}^{m-1} A^{m-1-k} B u_k$$

Matricea M

Matricea V

$$\begin{bmatrix} Y_1 \\ \vdots \\ \vdots \\ Y_m \end{bmatrix} = \begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ A^2B & AB & B \dots & 0 \\ \vdots & & & \\ A^{n-1}B & A^{n-2}B & \dots & B \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{m-1} \end{bmatrix} + \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^n \end{bmatrix} z_0$$

REVENIND LA FUNCTIA DE COST

Inlocuind iesirea prezisa prin dependenta de intrare, dedusa mai devreme, obtinem:

$$\min_{u \in \mathbb{R}^m} \|Mu + V - Y_{ref}\|^2_{\bar{Q}} + \|u\|^2_R$$

$$\bar{Q} = \begin{pmatrix} Q & & 0 \\ & \ddots & \\ 0 & & Q \end{pmatrix} \in \mathbb{R}^{ns \times ns}$$

$$\Rightarrow \min_{u \in \mathbb{R}^m} \frac{1}{2} u^T H u + q^T u$$

$$H = M^T Q M + R \qquad q = (V - Y_{ref})^T Q M$$

COD

```
m = 0.2; % masa bilei
g = 9.8; % acceleratia gravitationala

Ts = 0.05; % perioada

Cx = 5/7;
Cy = 5/7;

Q = [1500 0; ...
      0 10]; % R si Q sunt matrice de weight
R = 5;

N = 15; % orizontul de predictie
```

```
%% Pentru X
Adx = [1    Ts;...
       0    1];

Bdx = [0;  Cx * Ts];
```

```
%% Pentru Y
Ady = [1    Ts;...
       0    1];

Bdy = [0;  Cy * Ts];
```

```
%% Constrangeri si stari
 % Constrangerea de tip box pentru intrare u
 u_lb = -0.5;
 u_ub = 0.5;

 % Starea initiala a sistemului
 z0x = [0.06; -0.1];

 z0y = [0.06; -0.06];

 % Referintele pe care le urmarim
 phi_ref = 0;
 theta_ref = 0;
 z_ref = [0; 0];

 step = 0;
 zx = z0x;
 zy = z0y;

 u_phi = [];
 u_theta = [];

 zfx = z0x;
 zfy = z0y;
 phi = -0.1 * ones(N, 1);
```

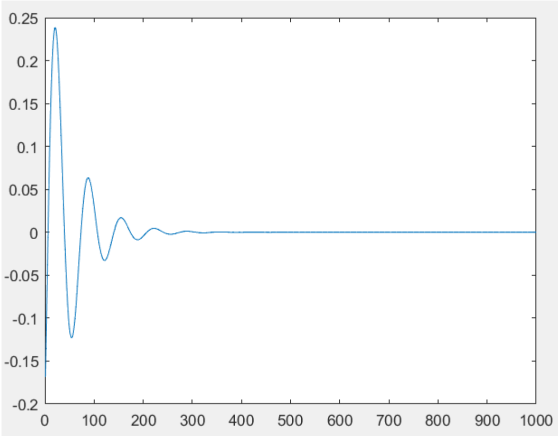
ADUCEREA BILEI ÎN CENTRUL MESEI

```
%%  Aducerea bilei in centrul mesei
 while step < 1000
    [Hx,qx,Cx,dx]= denseMPC(Adx, Bdx, Q, R, zx, N, u_ub, u_lb, z_ref, phi_ref);
    [Hy,qy,Cy,dy]= denseMPC(Ady, Bdy, Q, R, zy, N, u_ub, u_lb, z_ref, theta_ref);
%     phi = quadprog(Hx, qx, Cx, dx);
%     theta = quadprog(Hy, qy, Cy, dy);
    %% GRADIENT PROIECTAT
    epsilon = 1e-5;
    L = max(eig(Hx));
    alfa = 1 / L;
    dF = Hx * phi + qx;
    while norm(dF) > epsilon
        phi = phi - alfa * dF;
        phi = min(u_ub, max(u_lb, phi));
        dF = Hx * phi + qx;
    end
    %% BARIERA
    tau = 1;
    sigma = 0.6;
    stepb = 1;
    theta = zeros(N, 1);
    d1 = u_ub * ones(N,1);
    d2 = u_lb * ones(N,1);

    dF = @(x, tau) Hy * x + qy - tau * (-1 ./ (-x + d1) + (1./(x + d2)));
    ddF = @(x,tau) Hy - tau * (diag(-1 ./ ((-x + d1) .^ 2)) + diag(-1 ./ ((x + d2) .^ 2)));
    while tau >= epsilon
         while norm(dF(theta, tau)) >= epsilon && stepb <= maxIter
            theta = theta - inv(ddF(theta, tau)) * dF(theta, tau);
```

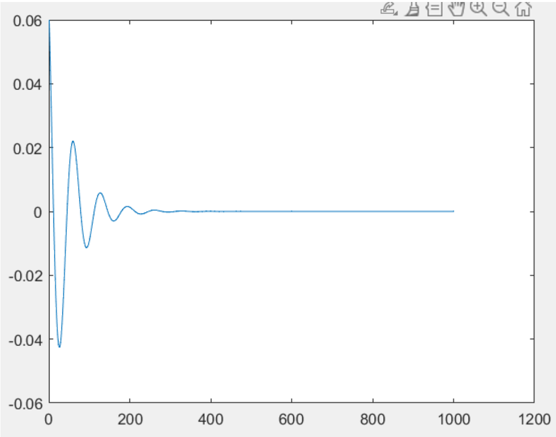
ADUCEREA BILEI ÎN CENTRUL MESEI

```
            stepb = stepb + 1;
            if stepb == maxIter
                disp('S-au atins numarul maxim de iteratii.');
                break;
            end
        end
        tau = sigma * tau;

     end
    u_phi = [u_phi, phi(1,1)];
    u_theta = [u_theta, theta(1,1)];
    znoux = Adx * zx + Bdx * phi(1,1);
    znouy = Ady * zy + Bdy * theta(1,1);
    zx = znoux;
    zy = znouy;
    zfx = [zfx zx];
    zfy = [zfy zy];
    step = step + 1;
  end
```

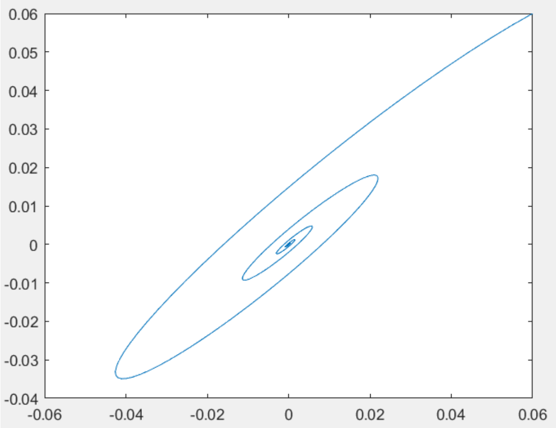
Comanda phi



Pozitia pe x

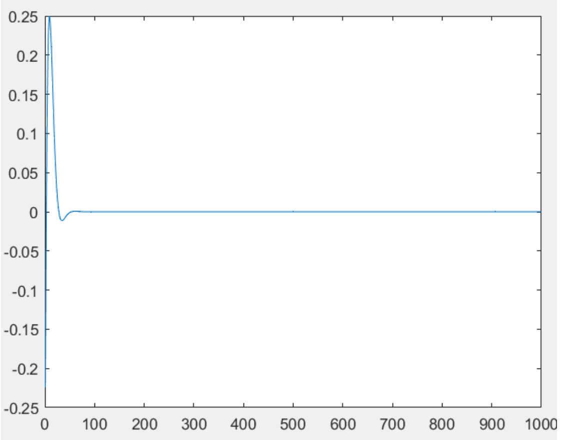


Planul xOy

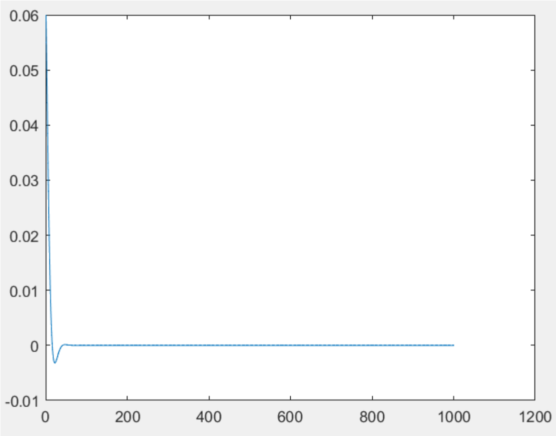


N = 5

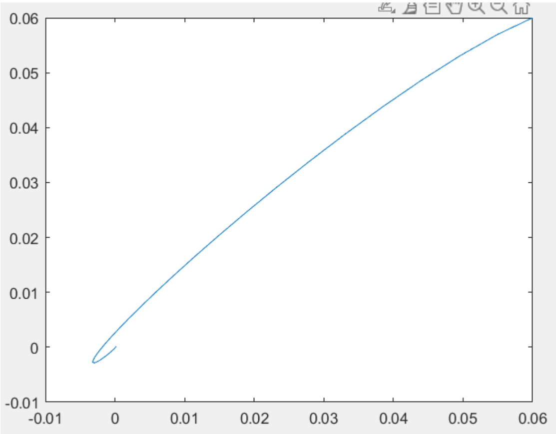
Comanda phi



Pozitia pe x



Planul xOy



N = 15

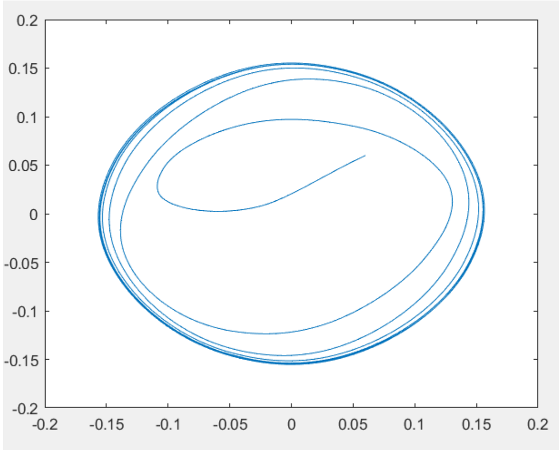
TRAIECTORIE PATRATA

```
%% Traiectorie patrat
 while step < 1000
    tr = puncte_patrat([zx(1), zy(1)], N);


    z_refx = [tr(1, 1); 0];
    z_refy = [tr(1, 2); 0];
    [Hx,qx,Cx,dx]= denseMPC(Adx, Bdx, Q, R, zx, N, u_ub, u_lb, z_refx, phi_ref);
    [Hy,qy,Cy,dy]= denseMPC(Ady, Bdy, Q, R, zy, N, u_ub, u_lb, z_refy, theta_ref);

     phi = quadprog(Hx, qx, Cx, dx);
%    theta = quadprog(Hy, qy, Cy, dy);

    %% CVX
%    cvx_begin quiet
%    variable theta(N)
%    minimize(1/2 * theta' * Hy * theta + qy' * theta);
%    subject to
%    Cy * theta <= dy;
%    cvx_end

%% NEWTON PROIECTAT
     theta = zeros(N, 1);
     ddfinv = inv(Hy);
     df = Hy * theta + qy;
     stepn = 1;
     while norm(df) >= 1e-5 && stepn <= 1e4
          theta = theta - ddfinv * (Hy * theta + qy);
```

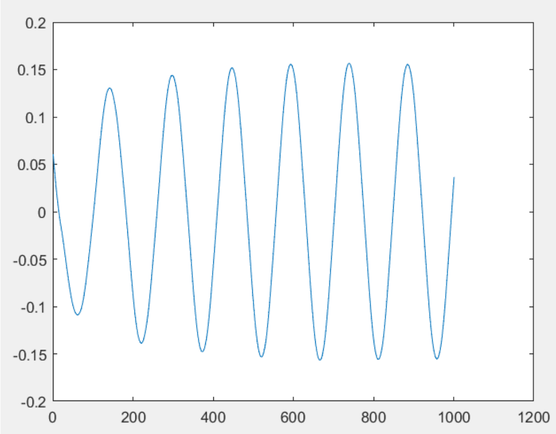
TRAIECTORIE PATRATA

```
        theta = min(u_ub, max(u_lb, theta));
        df = Hy * theta + qy;
        stepn = stepn + 1;
        if stepn == 1e4
                disp('S-au atins numarul maxim de iteratii.');
                break;
        end
    end
    u_phi = [u_phi, phi(1,1)];
    u_theta = [u_theta, theta(1,1)];
    znoux = Adx * zx + Bdx * phi(1,1);
    znouy = Ady * zy + Bdy * theta(1,1);
    zx = znoux;
    zy = znouy;
    zfx = [zfx zx];
    zfy = [zfy zy];
    step = step + 1;
end
```

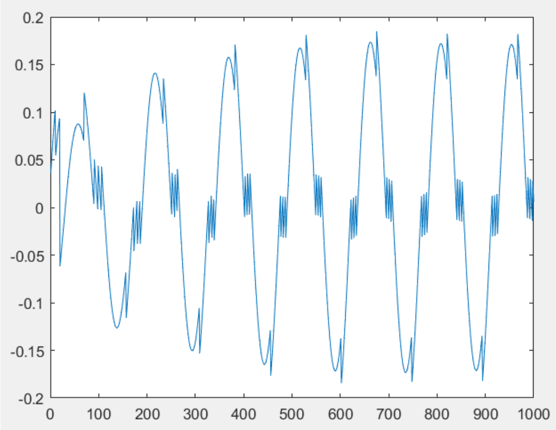
Planul xOy



Pozitia pe x

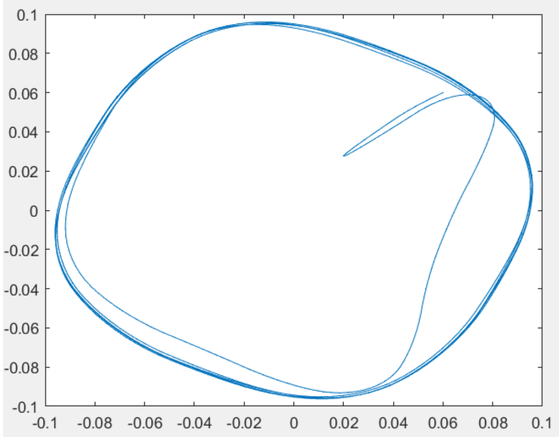


Comanda phi

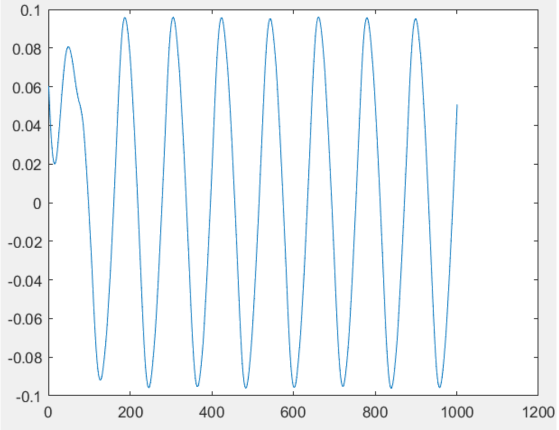


TRAIECTORIE PATRATA CVX

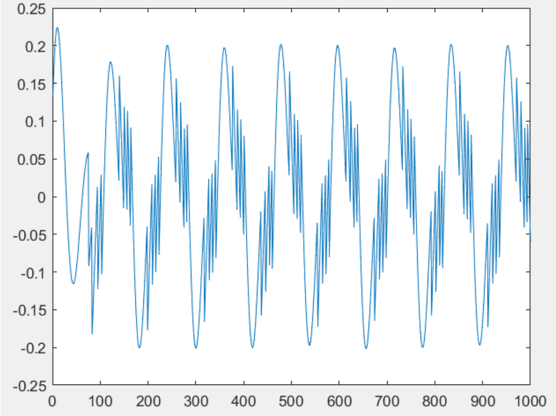
Planul xOy



Pozitia pe x

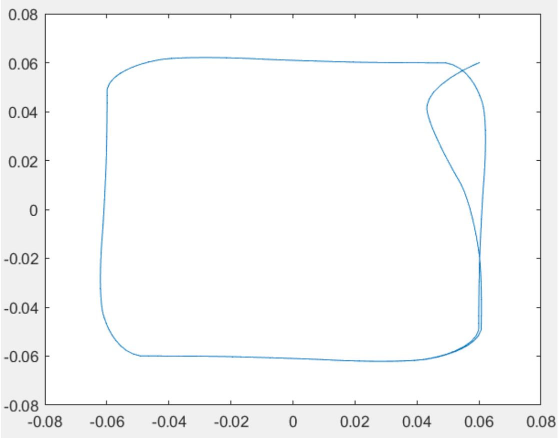


Comanda phi

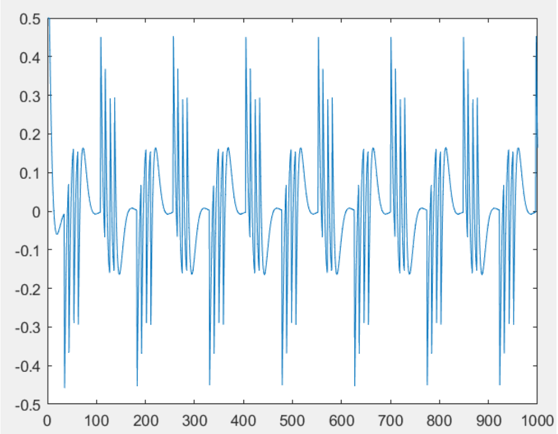


N = 5

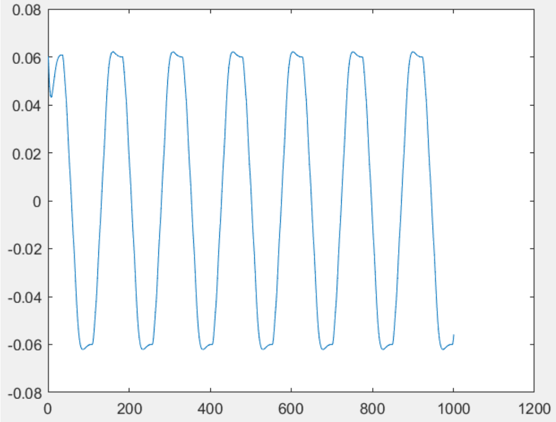
Planul xOy



Comanda phi



Pozitia pe x



N = 15

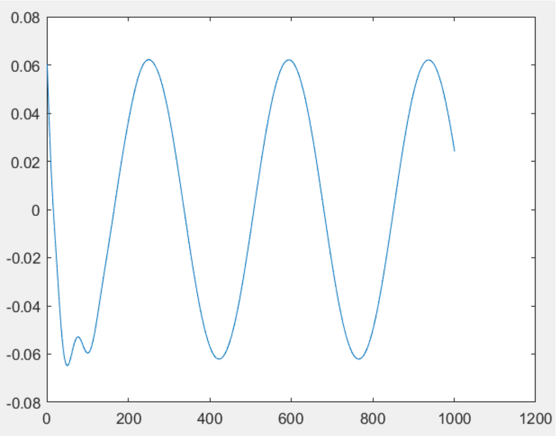
TRAIECTORIE CIRCULARA

```
%% Traiectorie circulara
 while step < 1000
    alfa = atan(zy(1) / zx(1));
    if zx(1) < 0
        alfa = alfa + pi;
    end

    z_refx = [0.06 * cos(alfa + 0.1); 0];
    z_refy = [0.06 * sin(alfa + 0.1); 0];

    [Hx,qx,Cx,dx]= denseMPC(Adx, Bdx, Q, R, zx, N, u_ub, u_lb, z_refx, phi_ref);
    [Hy,qy,Cy,dy]= denseMPC(Ady, Bdy, Q, R, zy, N, u_ub, u_lb, z_refy, theta_ref);
%    phi = quadprog(Hx, qx, Cx, dx);
%    theta = quadprog(Hy, qy, Cy, dy);

    %% GRADIENT PROIECTAT
     epsilon = 1e-8;
     phi = zeros(N, 1);
     L = max(eig(Hx));
     alfa = 1 / L;
     dF = Hx * phi + qx;
     while norm(dF) > epsilon
         phi = phi - alfa * dF;
         phi = min(u_ub, max(u_lb, phi));
         dF = Hx * phi + qx;
     end

     %% BARIERA
     tau = 1;
     sigma = 0.6;
     stepb = 1;
     maxIter = 1e3;
     epsilon = 1e-8;
     theta = zeros(N, 1);
     d1 = u_ub * ones(N,1);
     d2 = u_lb * ones(N,1);
```

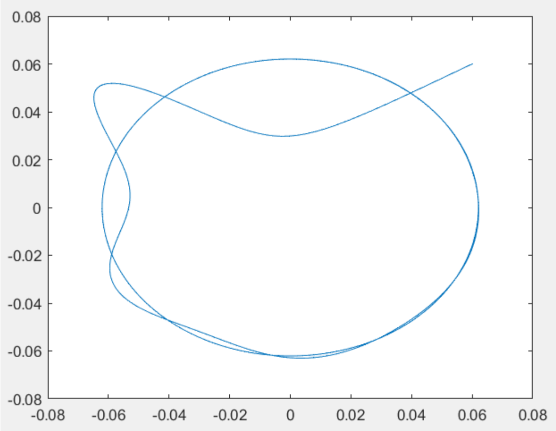
TRAIECTORIE CIRCULARA

```
dF = @(x, tau) Hy * x + qy - tau * (-1 ./ (-x + d1) + (1./(x + d2)));
ddF = @(x,tau) Hy - tau * (diag(-1 ./ ((-x + d1) .^ 2)) + diag(-1 ./ ((x

while tau >= epsilon
     while norm(dF(theta, tau)) >= epsilon && stepb <= maxIter
        theta = theta - inv(ddF(theta, tau)) * dF(theta, tau);
        stepb = stepb + 1;
        if stepb == maxIter
            disp('S-au atins numarul maxim de iteratii.');
            break;
        end
     end
     tau = sigma * tau;

end

u_phi = [u_phi, phi(1,1)];
u_theta = [u_theta, theta(1,1)];
znoux = Adx * zx + Bdx * phi(1,1);
znouy = Ady * zy + Bdy * theta(1,1);
zx = znoux;
zy = znouy;
zfx = [zfx zx];
zfy = [zfy zy];
step = step + 1;
end
```

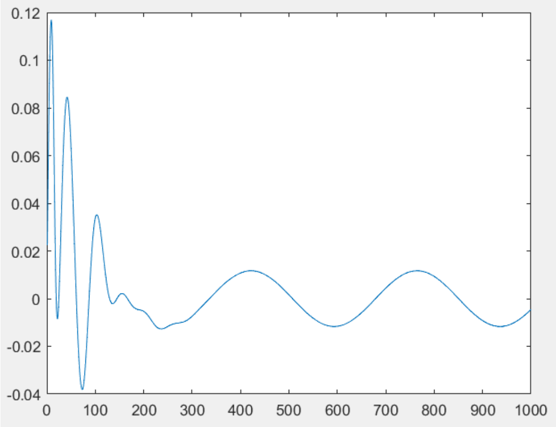
Comanda phi



Planul xOy

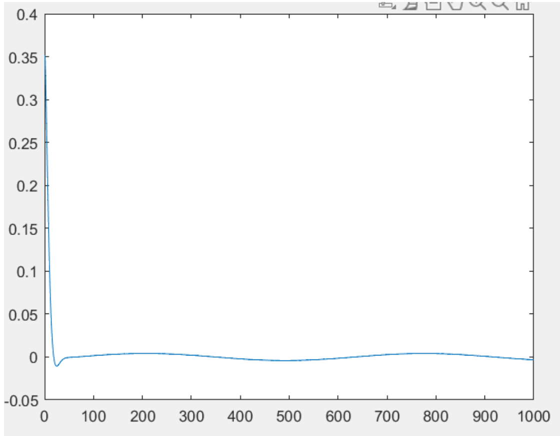


Pozitia pe x

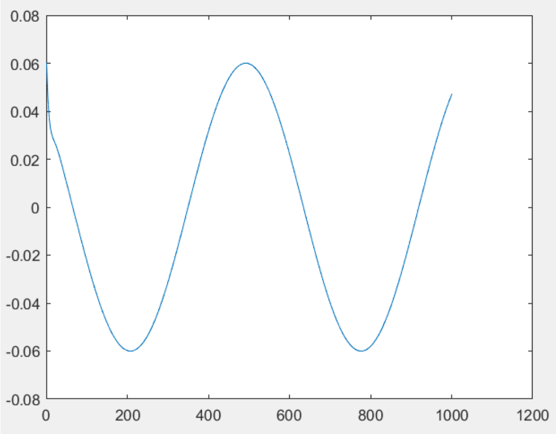


N = 5

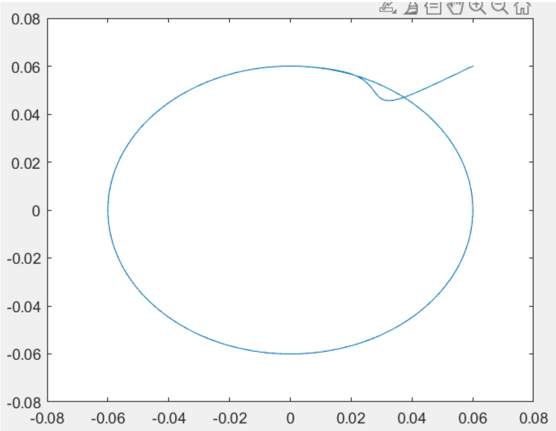
Pozitia pe x



Comanda phi



Planul xOy



N = 15

BIBLIOGRAFIE

- Fast Real-Time Model Predictive Control for a Ball-on-Plate Process - Krzysztof Zarzycki, Maciej Ławry´nczuk – MDPI

- Model Predictive Control of a Ball and Plate laboratory model - Matej Oravec, Anna Jadlovská - IEEE 13th International Symposium on Applied Machine Intelligence and Informatics

- Design and Control of Ball on Plate System - G . Joselin Retna Kumar, N. Showme, M. Aravind and R. Akshay - IJCTA