

Tema curs 13

Olaru Constantin – Alexandru

Grupa 30121

Determinarea funcției de transfer a H_o din răspunsul în frecvență a H_{des}

Am ales $H_{des} = \frac{1}{s} e^{-0.5s}$ pentru a verifica cât mai bine intersecția cu cercurile din care putem scoate pulsațiile ω .

```
% valoarea maxima pentru M
M_max = 2;

% generarea graficului cu 500 de puncte ale raspunsului in frecventa pentru
% un H ales
H = tf(1,[1,0],'iodelay',0.5);
w = logspace(-1,2,500);
[re,im] = nyquist(H,w);
re_1 = re(1,1,:);
im_1 = im(1,1,:);

% generarea cercurilor
figure
for M = 0:0.2:M_max
    theta = linspace(0, 2*pi, 500);
    R = sqrt(M)/(M-1);
    x_cerc = -M/(M-1);
    y_cerc = 0;

    X = x_cerc + R * cos(theta);
    Y = y_cerc + R * sin(theta);

    % stocarea valorilor de pe marginea cercurilor
    puncte_cerc = [X(:),Y(:)];
    intersectii = [];
    for i = 1:length(puncte_cerc)

        % conditia de oprire pentru verificare a intersectiilor
        intersectie_gasita = false;
        if intersectie_gasita
            break;
        end

        % calculul distantei intre puncte pentru validarea intersectiilor
        % cu o limita de 0.01
        x_cerc = puncte_cerc(i,1);
        y_cerc = puncte_cerc(i,2);
        dist = sqrt((re_1(:) - x_cerc).^2 + (im_1(:) - y_cerc).^2);
        [min_dist, idx] = min(dist);
        if min_dist < 0.01
            intersectii = [intersectii; re_1(idx), im_1(idx)];
        end
    end
end
```

```

    % afisarea intersectiilor
    if ~isempty(intersectii)
        plot(intersectii(:,1), intersectii(:,2), 'x', 'LineWidth', 3), hold on
        intersectie_gasita = true;
    end

end

% afisarea cercurilor
plot(X,Y, '--r'), hold on
end

% afisarea axelor, raspunsului in frecventa a lui H si a unei drepte care trece
prin x = 1/2
plot([-10 10], [0, 0], 'b'), plot([0, 0], [-10 10], 'b'), plot([-1/2 -1/2], [-10
10]), grid, hold on
plot(re_1(:), im_1(:), 'LineWidth', 2), axis([-10 10 -10 10])

% repetarea algoritmului pentru N
figure
N_min = -1;
N_max = 1;
for N = N_min:0.2:N_max
    theta = linspace(0, 2*pi, 500);
    R = 1/2 * sqrt(N^2 + 1)/N;
    x_cerc = - 1/2;
    y_cerc = 1/2/N;

    X = x_cerc + R * cos(theta);
    Y = y_cerc + R * sin(theta);

    puncte_cerc = [X(:), Y(:)];
    intersectii = [];
    for i = 1:length(puncte_cerc)

        intersectie_gasita = false;
        if intersectie_gasita
            break;
        end

        x_cerc = puncte_cerc(i,1);
        y_cerc = puncte_cerc(i,2);
        dist = sqrt((re_1(:) - x_cerc).^2 + (im_1(:) - y_cerc).^2);
        [min_dist, idx] = min(dist);
        if min_dist < 0.01
            intersectii = [intersectii; re_1(idx), im_1(idx)];
        end

        if ~isempty(intersectii)
            plot(intersectii(:,1), intersectii(:,2), 'x', 'LineWidth', 3), hold on
            intersectie_gasita = true;
        end
    end
end

plot(X,Y, '--r'), hold on
end

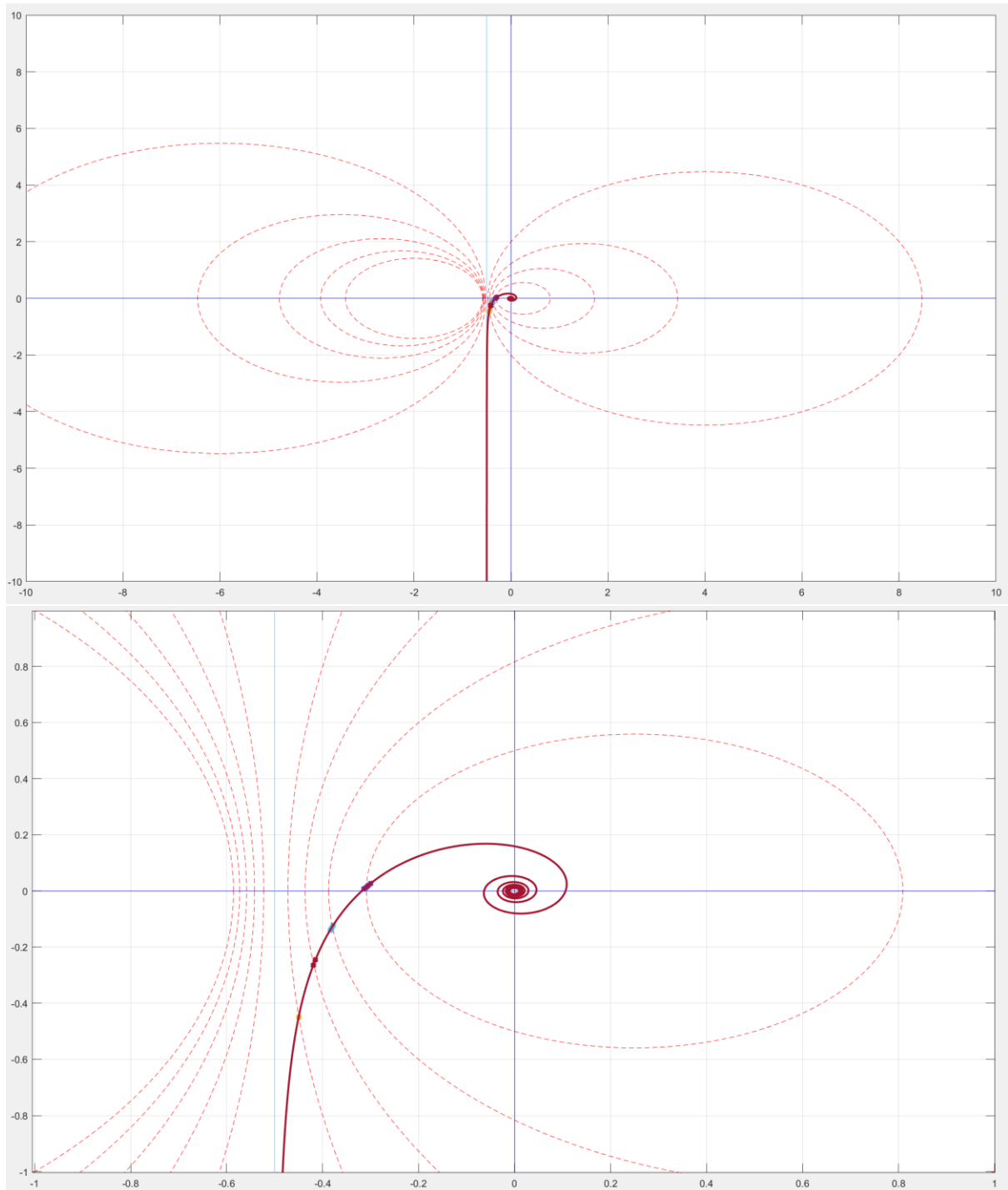
```

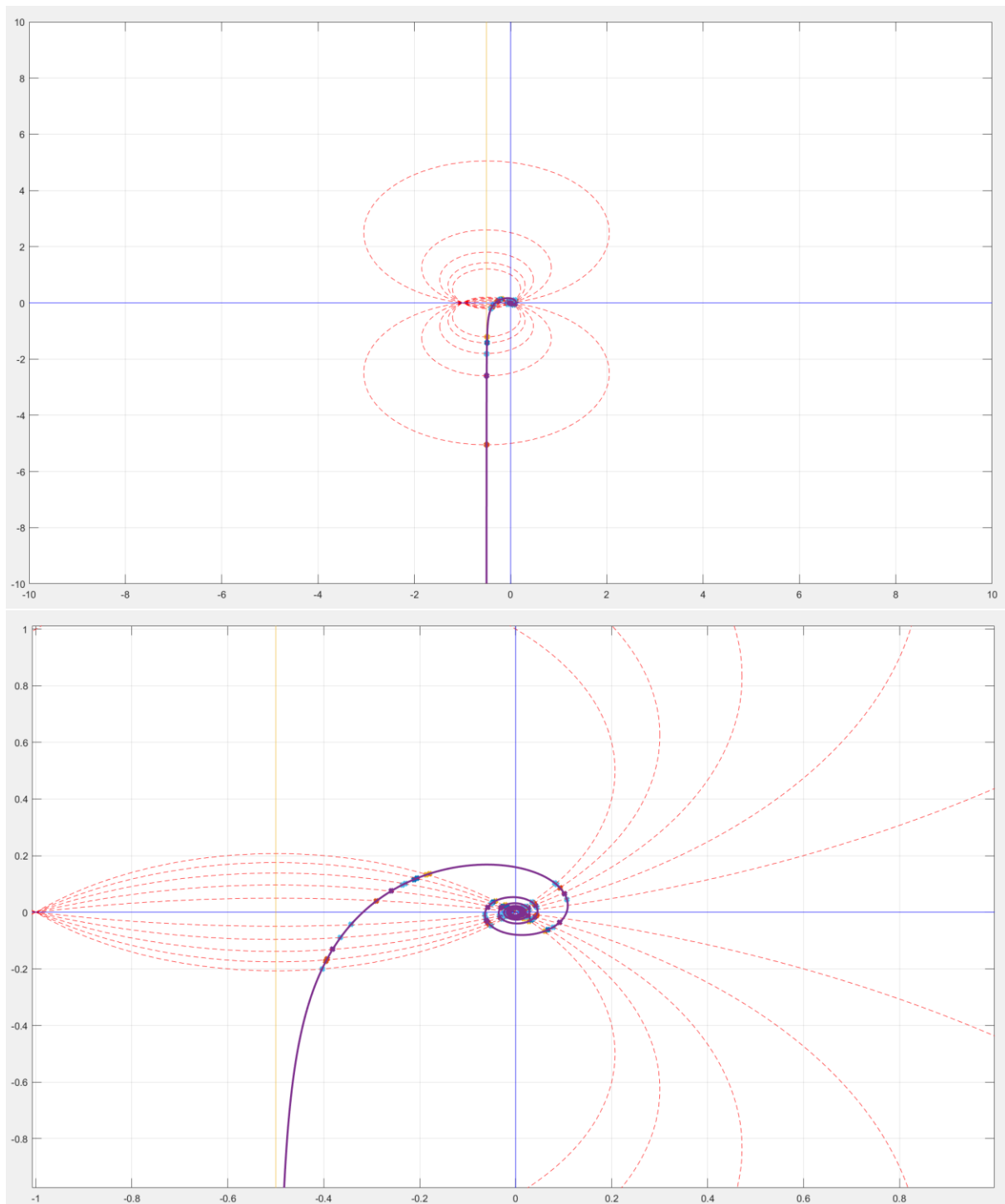
```

plot([-10 10], [0, 0], 'b'), plot([0, 0], [-10 10], 'b'), plot([-1/2 -1/2], [-10
10]), grid, hold on
plot(re_1(:), im_1(:), 'LineWidth', 2), axis([-10 10 -10 10])

%%

```





Din grafice se poate observa raspunsul in frecventa, cercurile influentate de M si N si intersectiile dintre ele cu X de diferite culori si aceeasi grosime. Nu este foarte precisa intersectia, iar din aceasta cauza nu am putut extrage ω la care se petrec acele intersectii.

Pentru a putea extrage acele valori, ar fi trebuit mai multe puncte, ceea ce ar fi incetinit performanta programului.