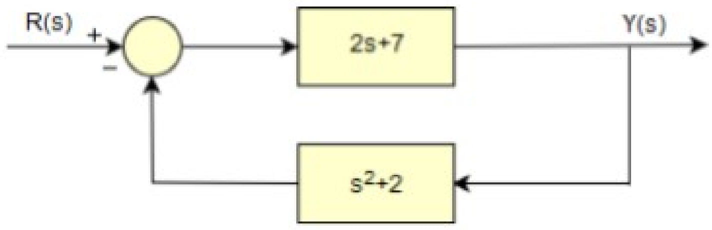
Γρηγοριάδης Παναγιώτης
Εργασία 3, AEM:1833

Έστω το παρακάτω σύστημα



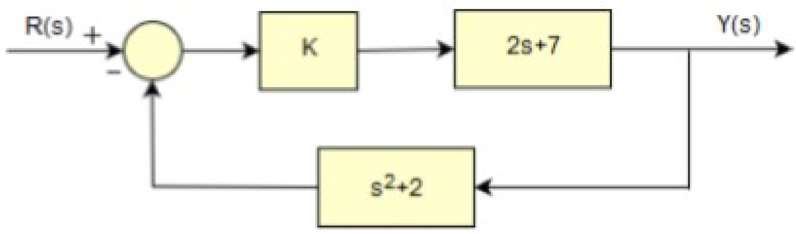
Σχήμα 1

(α) Να υλοποιήσετε τον αλγόριθμο της μεθόδου Ruth και με βάση τον αλγόριθμο αυτό να διερευνήσετε την ευστάθεια του συστήματος αυτού.

Απαντάται λεπτομερώς στον κώδικα.
Un-stable system! (There are changes of signs in the first column)

Number of right hand side poles =2

(β) Προσθέστε έναν ελεγκτή με κέρδος Κ και προσδιορίστε το ελάχιστο Κ με ακρίβεια 0.1, ούτως ώστε η υπερύψωση της μοναδιαίας βηματικής απόκρισης του συστήματος να μην υπερβαίνει το 5%. Με βάση τα ευρήματά σας, σχολιάστε το ρόλο του ελεγκτή στο σύστημα.



Σχήμα 2

Απαντάται λεπτομερώς στον κώδικα.
We used trial and error to find the proper value for k.
If we had a 2nd order system, we would be able to find ζ and through ζ, find k, but since our system is of 3rd order, we cannot do that!
We ended up picking Gain:
k = -0.077800

Our overshooting is: 4.519231%
Our controller was extremely important because it made our system converge to ~0.6, while previously it diverged

Note:

If we had a second order system, we would find ζ like this:
acc = 0.1;
Mp <= 0.05;
Since our Mp <= 0.05
e^(((-π)*ζ)/(sqrt(1-(ζ^2)))) <= 0.05
=> ζ >= 0.69010
Then we would compare our transfer function (with depended variable k) with the standard second order transfer funcion model (where ζ >= 0.69010) and we would find the proper value of k.
However we cannot do that, because our system is 3rd order, so we used trial and error method!

(γ) Για το κέρδος που θα επιλεγεί, να εξεταστεί η ευστάθεια του συστήματος με τη μέθοδο Ruth καθώς και αναλύοντας την καμπύλη Nyquist του συστήματος

Απαντάται λεπτομερώς στον κώδικα.
Stable system! (There are NO changes of signs in the first column)

Number of right hand side poles =0

Nyquist diagram conclusion:
We can see in the Nyquist diagram that we have 0 encirclements of -1 (N=0).
From the Routh Hurwitz method we saw that we have 0 sign changes (P=0).
In order for our system to be stable, Z (closed loop poles) must be 0 (N = Z-P).
So we have: Z = N + P => Z = 0
That means that our system is stable!!

(δ) Στο αρχικό σύστημα του Σχήματος 1 να πραγματοποιήσετε στον κώδικά σας μελέτη της επίδρασης της προσθήκης ενός μηδενικού $(1/a)*(s + a)$ στη συμπεριφορά του συστήματός σας για διάφορες τιμές του α, θεωρώντας σήμα δοκιμής τη μοναδιαία βηματική συνάρτηση και να καταγράψετε τα συμπεράσματά/σχόλιά σας.

Απαντάται λεπτομερώς στον κώδικα.
We tried values of a from waaay negative to waay positive (-500 to +500) and we tests our system stability!
Conclusion:
We can see that our system is unstable for all possible a values!!

(ε) Όπως στο ερώτημα (γ), να πραγματοποιήσετε μελέτη της επίδρασης της προσθήκης ενός πόλου (γs+1) στη συμπεριφορά του συστήματός σας για διάφορες τιμές του γ, θεωρώντας σήμα δοκιμής τη μοναδιαία βηματική συνάρτηση και να καταγράψετε τα συμπεράσματά/σχόλιά σας.

Απαντάται λεπτομερώς στον κώδικα.
We took as input the transfer function of b'.
Conclusion:
We tested our system for a wide variety of γ (-500 to +500) and we can see that our system is unstable for γ<0 and stable for γ>=0 !!

Κώδικας

Main:

```
Clear
clc
close all


%syms s
s = tf('s');

%Find transfer function
g = 2*s + 7; %tf([2 7])
h = s^2 + 2; %tf([1 0 2])
fprintf('Transfer function:')
f = feedback(g,h)

%Plot step response
figure()
step (f,10000) %we can see that our step response diverges!
grid on;

%NOTE: PRESS F5 TO CONTINUE FROM BREAKPOINTS!!

%A.
fprintf ('A.\n')
a_Grigoriadis_1833(f)

%B.
fprintf ('B.\n')
f_new = b_Grigoriadis_1833(g, h)

%C.
fprintf ('C.\n')
c_Grigoriadis_1833(f_new)

%D.
fprintf ('D.\n')
d_Grigoriadis_1833(f)

%E.
fprintf ('E.\n')
%NOTE: I take as input the function from b, not the initial function.
e_Grigoriadis_1833(f_new)

%If you want, you can run the following command, to do the same (e) for
the
%initial function. It will work perfectly, but the printing messages
will
%be wrong, because they are depended on the results!
%You can easily extract analogous information from the plots.
%The system will be unstable for all γ values again (boring) and the
Nyquist diagrams a
```

```
%bit more interesting.

%e_Grigoriadis_1833(f)
```

A:

```
function [] = a_Grigoriadis_1833(f)
%A.

    %get numerator and denominator coefficients from our transfer
function
    [n, d] = tfdata(f,'v');
    fprintf ('Characteristic equation coefficients:')
    d

    %Check stability with Routh-Hurwitz method
    %Coefficients: [-2    -7    -4   -15]
    %these are our characteristic equation coefficients!
    ruth_Grigoriadis_1833(d)

    fprintf('\n')

end
```

B:

```
function [ f_new ] = b_Grigoriadis_1833( g, h )
%B.

    %If we had a second order system, we would find  ζ like this:
    %acc = 0.1;
    %Mp <= 0.05;
    %Since our Mp <= 0.05
    %e^( ((-π)*ζ)/(sqrt(1-(ζ^2))) ) <= 0.05
    % => ζ >= 0.69010
    %Then we would compare our transfer function (with depended
variable k)
    %with the standard second order transfer funcion model (where ζ >=
0.69010)
    %and we would find the proper value of k.
    %However we cannot do that, because our system ir 3rd order, so we
used
    %trial and error method!

    %I used trial and error to conclude what value k should get
    fprintf(['We used trial and error to find the proper value for
k.\n',...
        'If we had a 2nd order system, we would be able to find  ζ
',...
        'and through ζ, find k,\n but since our system is of 3rd order,
we cannot do that!\n'])
    k = -0.0778 ;
    fprintf ('We ended up picking Gain: k = %f\n',k)

    %calculate new transfer function
    f_new = feedback(g*k,h) ;
```

```
    %monitor overshooting %
    S = stepinfo(f_new);
    fprintf ('\nOur overshooting is: %f%%\n',S.Overshoot)

    figure()
    step (f_new) %our step response now converges to ~0.6
    fprintf(['Our controller was extremely important because it
made\n',...
        'our system converge to ~0.6, while previously it diverged\n'])
    grid on;

    fprintf('New transfer function:')

end
```

C:

```
function [ ] = c_Grigoriadis_1833( f_new )
%C.

    %get numerator and denominator coefficients from our transfer
function
    [num, den] = tfdata(f_new,'v');
    num = -num;
    fprintf ('Characteristic equation coefficients:')
    den = -den

    %Check stability with Routh-Hurwitz method
    %Coefficients: [0.1556    0.5446    0.3112    0.0892]
    %these are our characteristic equation coefficients!
    ruth_Grigoriadis_1833(den)

    %Check stability with Nyquist plot
    figure()
    nyquist (f_new)
    grid on;

    fprintf(['\nNyquist diagram conclusion:\n',...
        'We can see in the Nyquist diagram that we have 0 encirclements
of -1 (N=0).\n',...
        'From the Routh Hurwitz method we saw that we have 0 sign
changes (P=0).\n',...
        'In order for our system to be stable, Z (closed loop poles)
must be 0 (N = Z-P).\n',...
        'So we have: Z = N + P => Z = 0\n',...
        'That means that our system is stable!!\n\n'])

end
```

D:

```
function [] = d_Grigoriadis_1833( f )
%D.

    s = tf('s');

    %Trial and error for many 'a' values
```

```
    fprintf(['Is our system with the addition of a zero stable (in our
plot)?',...
    '\n\t1: stable\n\t0: unstable\n'])
    stable = [] ;
    for a = -100:2:100
        if a == 0
            a = 0.000001;
        end
        stable = [stable, check_stability_d(a,f)];
    end

    %print our stability for various 'a' values!
    x = linspace(-100,100,101);
    figure()
    subplot(6,1,1)
    plot (x,stable)
    ylim([-1.5 1.5])
    xlabel('a')
    ylabel('stability (0 or 1)')
    title('Stability of system for various a. 0:unstable, 1:stable')

    %a=0: Can't be, because we have division with 0!!
    %a!=0: Our system is unstable
    %Plot our step response for 5 values of a (0, -1, 1, -100, 100).

    subplot(6,1,2)
    a = 0.000001;
    r = (s+a)/a;
    step(f*r)
    title('Step response. Unstable system (for a--->0')

    subplot(6,1,3)
    a = -1;
    r = (s+a)/a;
    step(f*r)
    title('Step response. Unstable system (for a!=0, here a=-1)')

    subplot(6,1,4)
    a = 1;
    r = (s+a)/a;
    step(f*r)
    title('Step response. Unstable system (for a!=0, here a=1)')

    subplot(6,1,5)
    a = -100;
    r = (s+a)/a;
    step(f*r)
    title('Step response. Unstable system (for a!=0, here a=-100)')

    subplot(6,1,6)
    a = 100;
    r = (s+a)/a;
    step(f*r)
    title('Step response. Unstable system (for a!=0, here a=100)')
```

```
    fprintf(['\nConclusion:\nWe can see that our system is
unstable',...
        ' for all possible a values!!\n\n'])

end
```

D2:

```
function [stable] = check_stability_d( a,f )
%check stability for question  d

    s = tf('s');

    r = (s+a)/a;
    f_d = f*r;
    stable = isstable(f_d);

end
```

E:

```
function [] = e_Grigoriadis_1833( f )
%E.

    s = tf('s');

    %Trial and error for many 'gamma' values
    fprintf(['Is our system with the addition of a pole stable (in our
plot)?',...
    '\n\t1: stable\n\t0: unstable\n'])
    stable = [] ;
    for gamma = -100:2:100
        stable = [stable, check_stability_e(gamma,f)];
    end

    %print our stability for various 'gamma' values!
    x = linspace(-100,100,101);
    figure()
    subplot(4,2,[1 2])
    plot (x,stable)
    ylim([-1.5 1.5])
    xlabel('gamma')
    ylabel('stability (0 or 1)')
    title('Stability of system for various gamma. 0:unstable,
1:stable')

    %gamma: Our system is stable for any value of 'gamma'
    %Plot our step response for 3 values of gamma (0, -5, 5).

    subplot(4,2,3)
    gamma = 0;
    r = (gamma*s)+1;
    step(f*(1/r))
    title('Step response. Stable system (for gamma=0)')

    subplot(4,2,5)
    gamma = -5;
```

```
    r = (gamma*s)+1;
    step(f*(1/r))
    title('Step response. Unstable system (for gamma=-5)')

    subplot(4,2,7)
    gamma = 5;
    r = (gamma*s)+1;
    step(f*(1/r))
    title('Step response. Stable system (for gamma=5)')

    %Now we want to confirm our results through Routh-Hurwitz method
and
    %the Nyquist diagram. We will do that again for 3 values of gamma
    %(-5, 0, 5)

    %gamma = 0
    fprintf('\nγ = 0 (stable system):\n')
    subplot(4,2,4)
    gamma = 0;
    r = (gamma*s)+1;
    f_new = f*(1/r);
    nyquist (f_new)
    title('Nyquist: gamma=0')

    %get numerator and denominator coefficients from our transfer
function
    %to perform Routh Hurwitz
    [num, den] = tfdata(f_new,'v');
    num = -num;
    fprintf ('Characteristic equation coefficients:')
    den = -den
    ruth_Grigoriadis_1833(den)

    %gamma = -5
    fprintf('\nγ = -5 (UNstable system):\n')
    subplot(4,2,6)
    gamma = -5;
    r = (gamma*s)+1;
    f_new = f*(1/r);
    nyquist (f_new)
    title('Nyquist: gamma=-5')

    %get numerator and denominator coefficients from our transfer
function
    %to perform Routh Hurwitz
    [num, den] = tfdata(f_new,'v');
    num = -num;
    fprintf ('Characteristic equation coefficients:')
    den = -den
    ruth_Grigoriadis_1833(den)

    %gamma = 5
    subplot(4,2,8)
    fprintf('\nγ = 5 (stable system):\n')
    gamma = 5;
```

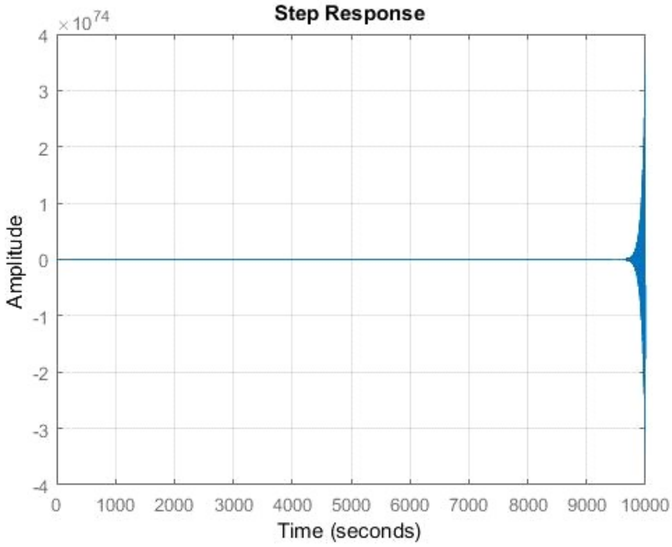
```
    r = (gamma*s)+1;
    f_new = f*(1/r);
    nyquist (f_new)
    title('Nyquist: gamma=5')

    %get numerator and denominator coefficients from our transfer
function
    %to perform Routh Hurwitz
    [num, den] = tfdata(f_new,'v');
    num = -num;
    fprintf ('Characteristic equation coefficients:')
    den = -den
    ruth_Grigoriadis_1833(den)

    fprintf(['\nConclusion:\nWe can see that our system is unstable
for',...
        ' γ<0 and stable for γ>=0 !!\n\n'])

end
```

E2:

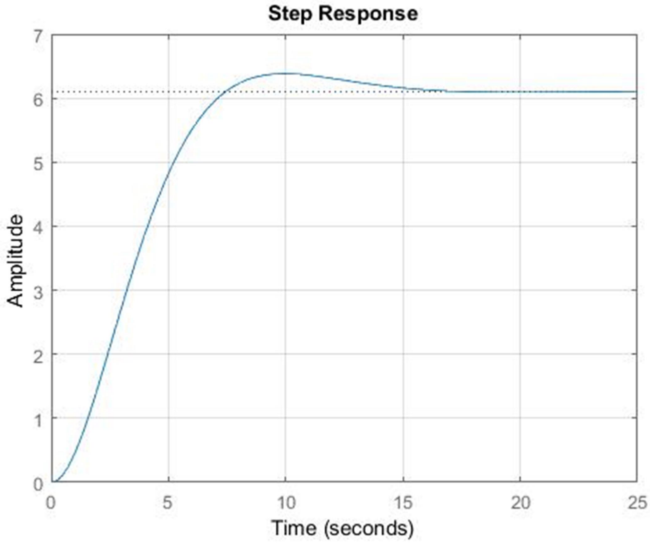
```
function [stable] = check_stability_e( gamma,f )
%check stability for question  e

    s = tf('s');

    r = (gamma*s)+1;
    f_e = f*(1/r);
    stable = isstable(f_e);

end
```

<u>Αποτελέσματα (plots) κώδικα:</u>

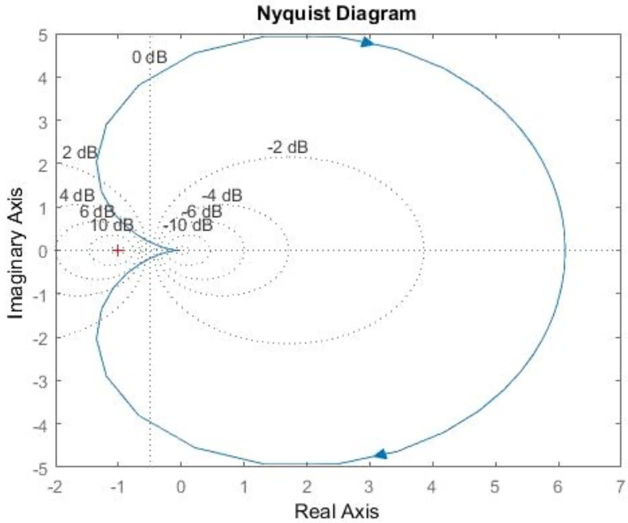
A)



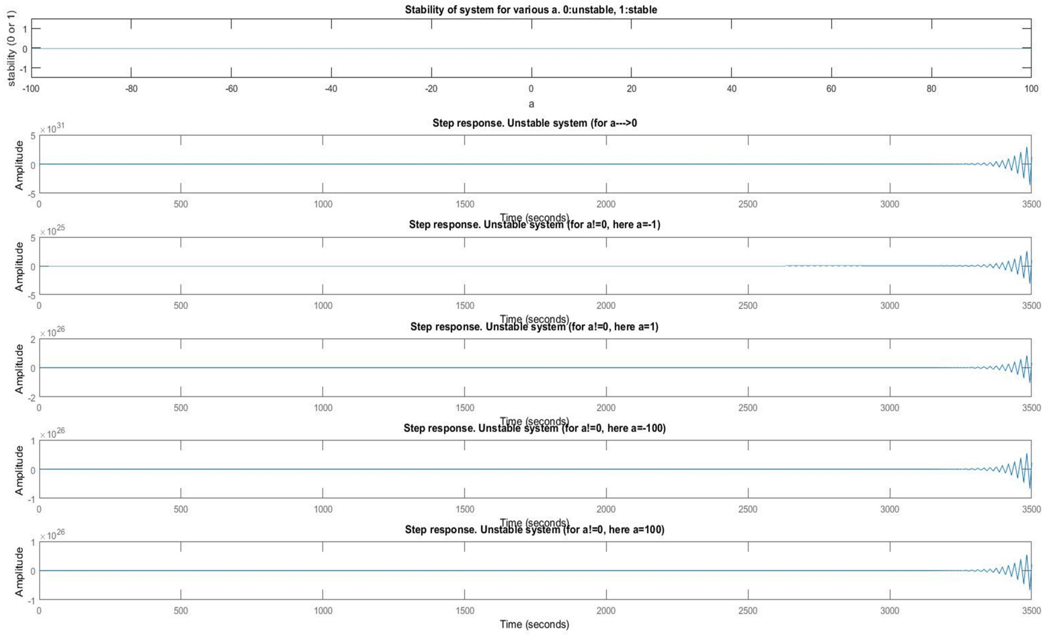
B)



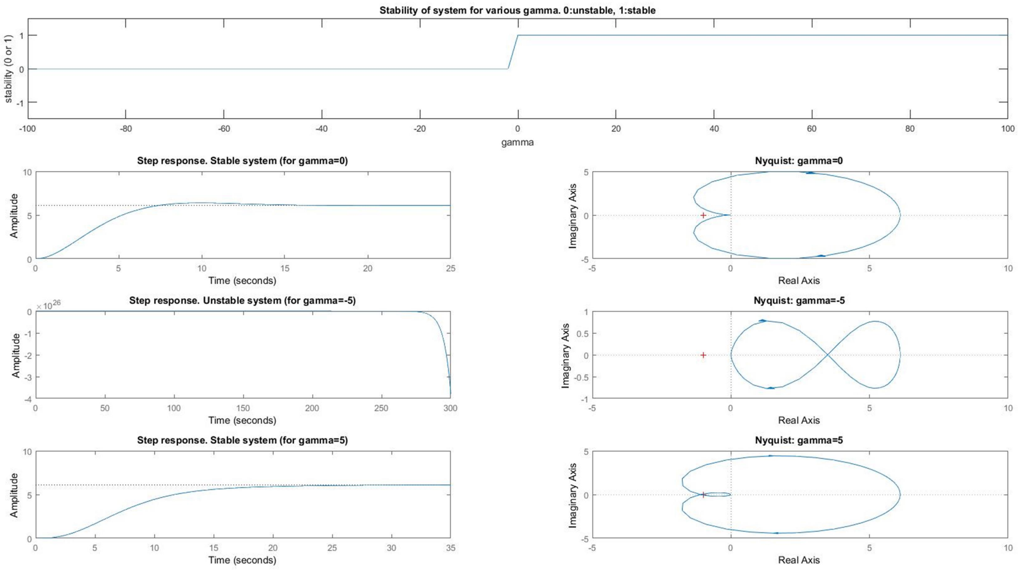
Γ)


Nyquist Diagram
Imaginary Axis
Real Axis
0 dB
-2 dB
2 dB
4 dB
6 dB
10 dB
-4 dB
-6 dB
-10 dB

Δ)


Stability of system for various a. 0:unstable, 1:stable
stability (0 or 1)
a
Step response. Unstable system (for a--->0
Step response. Unstable system (for a!=0, here a=-1)
Step response. Unstable system (for a!=0, here a=1)
Step response. Unstable system (for a!=0, here a=-100)
Step response. Unstable system (for a!=0, here a=100)
Amplitude
Time (seconds)

E)


Stability of system for various gamma. 0:unstable, 1:stable
stability (0 or 1)
gamma
Step response. Stable system (for gamma=0)
Amplitude
Time (seconds)
Nyquist: gamma=0
Imaginary Axis
Real Axis
Step response. Unstable system (for gamma=-5)
Nyquist: gamma=-5
Step response. Stable system (for gamma=5)
Nyquist: gamma=5