# Συστήματα Αυτομάτου Ελέγχου

## Προσομοίωση Συστημάτων

## Χρήση της εργαλειοθήκης του Matlab

(Μέρος Β)

## Control Systems Toolbox

Διδάσκοντες : Αν Καθηγήτρια Ο. Κοσμίδου
Καθηγητής Ι. Μπούταλης
Εργαστήριο Σ.Α.Ε. – Δ.Π.Θ.



## Μοντέλα στο χώρο καταστάσεων

- Για ένα Γραμμικό Χρονικά Αμετάβλητο (ΓΧΑ) σύστημα με *n* καταστάσεις, *p* εισόδους και *m* εξόδους:

Μοντέλο χώρου κατάστασης

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t)+\mathbf{B}\mathbf{u}(t)\\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t)+\mathbf{D}\mathbf{u}(t)\end{aligned} \qquad \begin{cases}\mathbf{x}(t): n \text{ x } 1, \quad \mathbf{u}(t): p \text{ x } 1, \quad \mathbf{y}(t): m \text{ x } 1\\ \mathbf{A}: n \text{ x } n, \quad \mathbf{B}: n \text{ x } p, \quad \mathbf{C}: m \text{ x } n, \quad \mathbf{D}: m \text{ x } p\end{cases}$$

Συνάρτηση Μεταφοράς

$$\mathbf{H}(s) = \mathbf{C}(s\mathbf{I}-\mathbf{A})^{-1}\mathbf{B}+\mathbf{D}$$

Ευστάθεια συστήματος: Πόλοι συν. Μεταφοράς = ρίζες χαρακτηριστικού Πολυωνύμου = Ιδιοτιμές πίνακα **A**

$$\det(s\mathbf{I}-\mathbf{A}) = 0$$

Οι μεταβλητές κατάστασης δεν είναι μοναδικές: Αν **z=Px**

$$\begin{aligned}\dot{\mathbf{z}}(t) &= \hat{\mathbf{A}}\mathbf{z}(t)+\hat{\mathbf{B}}\mathbf{u}(t)\\ \mathbf{y}(t) &= \hat{\mathbf{C}}\mathbf{z}(t)+\hat{\mathbf{D}}\mathbf{u}(t)\end{aligned} \Rightarrow \boxed{\begin{aligned}&\hat{\mathbf{A}} = \mathbf{P}\mathbf{A}\mathbf{P}^{-1}, \quad \hat{\mathbf{B}} = \mathbf{P}\mathbf{B}\\ &\hat{\mathbf{C}} = \mathbf{C}\mathbf{P}^{-1}, \quad \hat{\mathbf{D}} = \mathbf{D}\end{aligned}}$$

Μετασχηματισμός ομοιότητας

Οι ιδιοτιμές δεν μεταβάλλονται με τους μετασχηματισμούς ομοιότητας

## Χειρισμός μοντέλων χώρου κατάστασης

Το Matlab διαθέτει την εντολή **ss** (state-space) για τον ορισμό και χειρισμό μοντέλων Χ.Κ. Αφού οριστούν μπορούμε να πάρουμε και άλλες μορφές των συστημάτων, τις ιδιότητές τους ή τη χρονική τους απόκριση:

```
% Script e10 Enter a State-Space model
clear % removes all variables from the workspace
A = [-4 1 2; 1 -5 3; 2 0 -6]; % Enter A, B, C, D matrices
B = [1; 0.5; 2]; C = [2 1 2]; D =0;
Gss = ss(A,B,C,D) % create ss model
Gtf = tf(Gss) % convert to transfer function form
pause
Gss_from_tf = ss(Gtf) % convert back to state-space form
Gzpk = zpk(Gss) % convert to ZPK form
eigA = eig(A) % get eigenvalues of system matrix A
% they should be the same as the poles of Gtf
pause
Gss_from_zpk = ss(Gzpk) % convert back to s-s form
pzmap(Gss) % get pzmap from s-s object
```



## Χειρισμός μοντέλων χώρου κατάστασης

Έχοντας ορίσει μοντέλα Χ.Κ., μπορούμε να πάρουμε συνδυασμούς τους (serial, parallel, feedback) και τη χρονική απόκριση του συστήματος:

```
% Script e11 Get complex forms and time response of ss
models
clear % removes all variables from the workspace
A1 = [0 1; -3 -5]; B1 = [0; 1];  %  G1 state matrices
C1 = [1 2]; D1 = 0;
G1 = ss(A1,B1,C1,D1) % G1(s) as SS object
pause
A2 = [-3 1; 0 -4]; B2 = [0; 4]; %  G2 state matrices
C2 = [3 0]; D2 = 2;
G2 = ss(A2,B2,C2,D2); % G2(s) as SS object
pause; disp('Series connection Ts(s)');
Ts = G1*G2 %Series connection Ts(s)
pause; disp('Parallel connection Tp(s)');
Tp = G1+G2 %Parallel connection Tp(s)
pause; disp('Feedback connection Tf(s)');
Tf = feedback(G1,G2) %Feedback connection Tf(s)
pause;  t=[0:0.1:20]';  impulse(Tf,t)
pause
step(Tf,t)
```

# Διαγράμματα Ευστάθειας

- Τα εποπτικά κριτήρια ευστάθειας αναφέρονται σε συστήματα μιας εισόδου – μιας εξόδου. Απορρέουν από ειδικά διαγράμματα, που σχεδιάζονται από τη συνάρτηση μεταφοράς ανοικτού βρόχου $G(s)F(s)$

Γεωμετρικός τόπος ριζών: Μελέτη της ευστάθειας και σχετικής ευστάθειας του συστήματος. Η σταθερά ενίσχυσης *K* παίζει σημαντικό ρόλο, επειδή διαφορετικές τιμές της προκαλούν μετατόπιση των πόλων στο μιγαδικό επίπεδο.

Κριτήριο και διαγράμματα Nyquist: Δίνει πληροφορία για την απόλυτη και τη σχετική ευστάθεια του συστήματος και προσδιορίζει την απόκριση του συστήματος στο πεδίο της συχνότητας

Αρμονική απόκριση και διαγράμματα Bode: Η αρμονική απόκριση μας δίνει την έξοδο του συστήματος στη μόνιμη κατάσταση, όταν η είσοδός του είναι κάποια ημιτονοειδής συνάρτηση



# Γεωμετρικός τόπος ριζών

Το Matlab διαθέτει τις εντολές **rlocus** και **rlocfind.** Η πρώτη δίνει το γεωμετρικό τόπο, ενώ η δεύτερη επιτρέπει στο χρήστη να βρει το κέρδος οποιουδήποτε σημείου του γ.τ.ρ, στο οποίο κάνει κλικ με το ποντίκι.

$$G(s)H(s) = \frac{K(s+5)}{(s+1)(s+3)(s+12)}$$

```
% Script e12 Root locus for a system with real poles
clear % removes all variables from the workspace
disp('Create zero and pole arrays of G(s)')
  zOL = -5 % open-loop zero
  pOL = [ -1; -3; -12 ] % open-loop poles
  pause
  G = zpk(zOL,pOL,1) %Create G(s) as a ZP object with gain K=1
  pause;  disp('>>> Next, draw root-locus plot with default area')
  rlocus(G);   grid; title('Root Locus with default area')
  pause
  disp('Redraw root-locus plot with smaller area ')
  axis equal
  axis([-10 10 -10 10]);
  title('Root Locus with smaller area')
  disp('Use mouse or arrow keys to select ')
  disp('a point on the locus that will appear.')
  [kk,clroots] = rlocfind(G)
```

## Γεωμετρικός τόπος ριζών

Παράδειγμα συστήματος με πραγματικούς και μιγαδικούς πόλους στη συνάρτηση μεταφοράς ανοικτού βρόχου

$$G(s)H(s) = \frac{K(s+8)}{s(s+2)(s^2+8s+12)}$$

```
% Script e13 Root locus for a system with real and complex poles
clear % removes all variables from the workspace
den = conv([1 2 0],[1 8 32])  % denominator coefficients (4 poles)
G = tf([1 8],den)
pause
disp('zero, poles, and gain of G(s)')
Gzp = zpk(G)
pause
rlocus(G) % compute root locus
axis('equal')             % same scaling for both axes
axis([-15 5 -10 10]); % adjust plotting area
grid;title('Root Locus plotted for -15 < real < 5 & equal scaling')
[kk,clroots] = rlocfind(G) % calculate gain values and poles
```



## Διαγράμματα απόκρισης συχνότητας

- Το Matlab δίνει τη δυνατότητα σχεδιασμού διαγραμμάτων Bode, Nichols, και Nyquist.

Διάγραμμα Bode : Η εντολή **`bode(G, omega)`** παράγει ημιλογαριθμικό διάγραμμα του μεγέθους |G(jω) | (σε decibels) σε σχέση με το ω και ξεχωριστό ημιλογαριθμικό διάγραμμα arg(G(jω) ) (σε μοίρες) σε σχέση με το ω

Διάγραμμα Nyquist: Η εντολή **`nyquist(G, omega)`** παράγει διάγραμμα του G(jω) με άξονες τα Re[G(jω) ] και Im[G(jω) ].

Διάγραμμα Nichols: Η εντολή **`nichols(G, omega)`** παράγει διάγραμμα του μεγέθους |G(jω) | (σε decibels) σε σχέση με τη γωνία φάσης arg(G(jω) ) (σε μοίρες)

# Απόκριση σε ημιτονοειδή είσοδο

Η απόκριση (μόνιμη κατάσταση) σε ημιτονοειδή είσοδο $u(t) = a\cos(\omega_0 t + \theta)$

Δίνεται από τη σχέση $y_{ss}(t) = a|G(j\omega_0)|\cos(\omega_0 t + \theta + \arg[G(j\omega_0)])$

Βρίσκουμε την απόκριση του: $G(s) = \dfrac{10s+50}{s^2+4s+3}$ στην είσοδο $u(t) = 2\cos(5t+30^o)$

```
% Script e14 Sinusoidal state space response
clear % removes all variables from the workspace
G = tf([ 10 50 ],[ 1 4 3 ]) % 'Create G(s) as a TF object
t = [0:0.06:6]'; % time vector with 0.06 sec. increment
u = 2*cos(5*t + 30*pi/180);
y = lsim(G,u,t);
disp('magnitude and phase of frequency response at w = 5 rad/s')
[mag,phase] = bode(G,5)
pause
disp('magnitude and phase of steady-state response')
y_mag = 2 * mag
y_phase = 30 + phase
y_ss = 2*mag*cos (5*t + (30+phase)*pi/180);
plot(t,u,'-',t,y,'--',t,y_ss,'-.')
grid, title('u (solid), y (dashed) & yss (dashdot)')
xlabel('Time (secs)'), ylabel('u, y, yss'), text(0.7,2,'u(t)')
text(1.1,-6,'y(t)'), text(0.4,4,'yss(t)')
```



# Διαγράμματα Bode-Nichols-Nyquist

Χρησιμοποιούμε τις εντολές **bode**, **nichols** και **nyquist** για να πάρουμε την απόκριση συχνότητας G(jω) του συστήματος με συνάρτηση μεταφοράς:

$$G(s) = \frac{1280s+640}{s^4+24.2s^3+1604.81s^2+320.24s+16}$$

```
% Script e15 Bode, Nichols, Nyquist plots
clear % removes all variables from the workspace
numG = [ 1280 640 ]; denG = [ 1 24.2 1604.81 320.24 16 ];
G = tf(numG,denG) %Create G(s) as a TF object
pause
disp('Make Bode plot directly from the bode command')
w = logspace(-2,3,100)'; % logarithmically-spaced points as column
bode(G,w);
pause
disp('Make Nichols plot directly from the nichols command')
disp('  The encirclement direction is indicated')
nichols(G,w), grid, ngrid
axis([-270 0 -40 40])
pause
disp('Make Nyquist plot directly from the nyquist command')
nyquist(G);
grid, axis([-10 50 -30 30]), axis equal
```

# Βιβλιογραφία


1. R. Dorf and R. Bishop, “Σύγχρονα Συστήματα Αυτομάτου Ελέγχου”, Εκδόσεις Τζιόλα (11η Έκδοση στα Ελληνικά), 2009.
2. Dean Frederick and Joe Chow, “Feedback Control Problems using Matlab”, Brooks/Cole publishing, 2000.
3. B. Lurie and P. Enright, “Classical Feedback Control with Matlab”, Marcel Dekkerm, 2000.