Laborator Nr.3

SISTEME INCERTE. NOŢIUNI DE BAZĂ

1. NOŢIUNI INTRODUCTIVE

Principalele obiective ale proiectării sistemelor automate sunt asigurarea stabilităţii sistemelor şi rejectarea efectelor perturbaţiilor interne sau externe.

La ora actuală se cunosc un număr mare de metode şi tehnici de proiectare a sistemelor de reglare automată, toate bazându-se pe următoarele ipoteze:

- Sistemul poate fi modelat, cu structură cunoscută
- Parametrii modelului sunt cunoscuţi

Modelarea unui sistem fizic în vederea proiectării unui sistem de reglare implică un compromis între simplitatea modelului matematic şi acurateţea cu care acesta descrie comportarea sistemului real, astfel cele două ipoteze necesare ridică numeroase probleme. De cele mai multe ori, modelul matematic utilizat este o aproximare a sistemului fizic, valabil doar într-un domeniu restrâns, în jurul punctului staţionar de funcţionare. Parametrii modelului matematic sunt determinaţi de cele mai multe ori experimental. Astfel obţinerea performanţelor impuse este dificilă în condiţiile în care parametrii sau chiar structura procesului pot varia datorită unor perturbaţii.

Sistemele robuste se axează pe proiectarea şi implementarea unor sisteme de reglare în condiţii de incertitudini asupra modelului procesului condus sau pentru procese variabile în timp. Dacă pentru aceste incertitudini se poate stabili o normă matematică, teoria sistemelor robuste face posibil calculul unui regulator capabil să satisfacă anumite specificaţii de performanţă dacă se respectă domeniul incertitudinilor.

2. CONSTRUIREA MODELELOR CU PARAMETRII INCERŢI

Indiferent daca modelul matematic al unui sistem fizic este obţinut prin metode teoretice sau prin identificare experimentală, pentru a fi adecvat proiectării sistemelor de reglare, acesta trebuie simplificat, neglijând intenţionat unele părţi complicate ale sistemului. Diferenţele dintre sistemul real şi modelul nominal al sistemului reprezintă incertitudinile sistemului sau erorile de modelare.

Astfel se proiectează un regulator (compensator) pentru a asigura stabilitatea atât a modelul nominal cât şi a familiei de sisteme care există într-un domeniu de incertitudine.

Pentru a utiliza prezenţa incertitudinilor (erorilor de modelare) în proiectarea regulatoarelor este necesară determinarea unui model matematic a acestora, pe baza informaţiilor disponibile.

Modelarea incertitudinilor se poate realiza fie în domeniul timp, fie în domeniul frecvenţial.

3. UTILIZAREA TOOLBOXUL ROBUST CONTROL/MATLAB PENTRU CREEAREA MODELOR CU PARAMETRII INCERŢI

Elemente incerte - sunt blocurile de construcţie folosite pentru a forma matrici cu obiecte incerte şi sisteme cu obiecte incerte. Există 5 clase:

Function	Description
ureal	Uncertain real parameter
ultidyn	Uncertain, linear, time-invariant dynamics
ucomplex	Uncertain complex parameter
ucomplexm	Uncertain complex matrix
udyn	Uncertain dynamic system

Proprietăţile elementelor incerte pot fi accesate folosind funcţiile ***set*** şi ***get***.

Sintaxa:

p1 = ureal(name, NominalValue, Prop1, val1, Prop2, val2,...);

p2 = ucomplex(name, NominalValue, Prop1, val1, Prop2, val2,...);

p3 = ucomplexm(name, NominalValue, Prop1, val1, Prop2, val2,...);

p4 = ultidyn(name, ioSize, Prop1, val1, Prop2, val2,...);

p5 = udyn(name, ioSize, Prop1, val1, Prop2, val2,...);

Obs 1. Pentru *ultidyn* şi *udyn* parametrul *NominalValue* este valoare fixată în jurul căreia se consideră incertitudinile.

Obs 2. Pentru *ureal, ultidyn, ucomplex* şi *ucomplexm*, comanda *usample* va genera o valoare aleatoare (random) în gama modelată: *usample(p1).*

Parametrii reali incerţi **–** sunt utilizaţi pentru a reprezenta un număr real a cărui valoare este incertă. Parametrii reali incerţi au un nume (proprietatea Name) şi o valoare nominală (proprietatea NominalValue). Mai au o serie de alte proprietăţi (plusminus, Gama, procente) care descriu domeniul de incertitudine.

Properties	Meaning	Class
Name	Internal name	char
NominalValue	Nominal value of atom	double
Mode	Signifies which description (from 'PlusMinus', 'Range', 'Percentage') of uncertainty is invariant when NominalValue is changed	char
PlusMinus	Additive variation	scalar or 1x2 double
Range	Numerical range	1x2 double
Percentage	Additive variation (% of absolute value of nominal)	scalar or 1x2 double
AutoSimplify	'off' | {'basic'} |'full'	char

Proprietatea ***AutoSimplify*** controlează modul în care expresiile care implică parametri reali sunt simplificate. Poate avea valoarea „off” – nu se aplică simplificări, „basic” – se aplică metode elementare de simplificare şi „full” – se aplică tehnici de simplificare complexe.

Exemplu 1. Creaţi un parametru real, incert, cu valoarea nominală de 3, cu valori implicite pentru toate proprietăţile nespecificate. Vizualizaţi proprietăţile şi valorile lor.

```
a = ureal('a',3)
get(a)
Name: 'a'
NominalValue: 3
Mode: 'PlusMinus'
Range: [2 4]
PlusMinus: [-1 1]
Percentage: [-33.3333 33.3333]
AutoSimplify: 'basic'
```

Exemplu 2. Creaţi un parametru real, incert, cu valoarea nominală de 2, cu o variaţie de 20%. Vizualizaţi proprietăţile şi valorile lor.

```
b = ureal('b',2,'percentage',20)
get(b)
Name: 'b'
NominalValue: 2
Mode: 'Percentage'
Range: [1.6000 2.4000]
PlusMinus: [-0.4000 0.4000]
Percentage: [-20.0000 20.0000]
AutoSimplify: 'basic'
```

Modificaţi gama de variaţie a parametrului.

```
b.Range = [1.9 2.3]
get(b)
Name: 'b'
NominalValue: 2
Mode: 'Percentage'
Range: [1.9000 2.3000]
PlusMinus: [-0.1000 0.3000]
Percentage: [-5.0000 15.0000]
AutoSimplify: 'basic'
```

Modificaţi valoarea nominală a parametrului.

```
b.NominalValue = 2.2;
get(b)
Name: 'b'
NominalValue: 2.2000
Mode: 'Percentage'
Range: [2.0900 2.5300]
```

PlusMinus: [-0.1100 0.3300]
Percentage: [-5.0000 15.0000]
AutoSimplify: 'basic'

Exemplu 3. Creaţi un parametru real, incert, cu o variaţie asimetrică în jurul valori nominale Vizualizaţi proprietăţile şi valorile lor.

```
c = ureal('c',-5,'per',[-20 30]);
get(c)
Name: 'c'
NominalValue: -5
Mode: 'Percentage'
Range: [-6 -3.5000]
PlusMinus: [-1 1.5000]
Percentage: [-20 30]
AutoSimplify: 'basic'
```

Exemplu 4. Creaţi un parametru real, incert, cu o variaţie în procente dar proprietatea Mode să aibă valoarea Range. Vizualizaţi proprietăţile şi valorile lor.

```
d = ureal('d',-1,'mode','range','perc',[-40 60]);
get(d)
Name: 'd'
NominalValue: -1
Mode: 'Range'
Range: [-1.4000 -0.4000]
PlusMinus: [-0.4000 0.6000]
Percentage: [-40.0000 60]
AutoSimplify: 'basic'
```

Exemplu 5. Creaţi un parametru real, incert, cu proprietatea AutoSimplify având valoarea FULL.

```
e = ureal('e',10,'plusminus',[-2 3],'mode','perce','autosimplify','full')
get(e)
Name: 'e'
NominalValue: 10
Mode: 'Percentage'
Range: [8 13]
PlusMinus: [-2 3]
Percentage: [-20 30]
AutoSimplify: 'full'
```

Exemplu 6. Creaţi un parametru real, incert, folositi *usample* pentru a genera 1000 de eşantioane (rezultând într-o matrice 1 - pe - 1 – pe - 1000), remodelaţi matricea, şi trasaţi o histogramă , cu 20 de dreptunghiuri (în intervalul de la 2 la 4).

```
h = ureal('h',3);
hsample = usample(h,1000);
hist(reshape(hsample,[1000 1]),20);
```

Sisteme cu dinamică nemodelată - *udyn*, reprezintă sisteme multivariabile, neliniare complet necunoscute, care variază în timp.

Toate operaţiile algebrice, cum ar fi adunarea, scăderea, înmulţirea funcţionează în mod corespunzător şi operaţia de substituţie (*usubs*) este permisă. Cu toate acestea, toate instrumentele de analiză (de exemplu, *robuststab*) nu pot folosi aceste tipuri de elemente incerte.

Exemplu 7. Creaţi un element de dimensiunea 2-pe-3 folosind funcţia *udyn*.

```
m = udyn('m',[2 3])
get(m)
```

Matrici cu parametrii incerţi

Exemplu 8. Creaţi 2 parametrii incerţi şi apoi o matrice (3-pe-2) folosind acei parametrii incerţi.

```
a = ureal('a',3);
b = ureal('b',10,'pe',20);
M = [-a 1/b;b a+1/b;1 3]
```

Modificaţi elementul de pe linia 3 coloana 2:

```
M(3,2) = ureal('c',3,'perc',40)
```

Obs 3. Proprietăţile matricii pot fi accesate folosind funcţie *umat*.

```
get(M)
M.NominalValue
M.Uncertainty
M.Uncertainty.a
M.Uncertainty.a.Range
M
```

Obs 4. Elementele incerte pot fi înlocuite folosind funcţia *usubs*.

Exemplu 9. Înlocuiţi toate valorile paremetrului incert „a” cu valoarea 4.

```
M2 = usubs(M,'a',4)
```

Se poate înlocui toate gama de variaţie a parametrului real incert „b” cu gama de variaţie a parametrului incert „a” *M.Uncertainty.a.*

```
M3 = usubs(M,'b', M.Uncertainty.a)
```

Sisteme cu parametrii incerţi reprezentate în spaţiul stărilor

Sistemele cu parametrii incerţi (USS) sunt sisteme liniare reprezentate în spaţiul stărilor folosind matrici cu parametrii incerţi – obiecte *ss*.

Vizualizarea proprietăţilor unui sistem cu parametrii incerţi se realizează folosind funcţia *get*: get(sys).

Crearea modelelor în spaţiul stărilor cu parametrii incerţi

Modelele în spaţiul stărilor se creează pornind de la matricile din spaţiul stărilor folosind funcţia *ss*.

Exemplu 10. Creaţi 3 parametrii reali incerţi după care creaţi matricile A, B, C, D. Creaţi modelul în spaţiul stărilor folosind funcţia *ss*.

```
p1 = ureal('p1',10,'pe',50);
p2 = ureal('p2',3,'plusm',[-.5 1.2]);
p3 = ureal('p3',0);
A = [-p1 p2;0 -p1];
B = [-p2;p2+p3];
C = [1 0;1 1-p3];
D = [0;0];
sys = ss(A,B,C,D)
```

Răspunsul sistemului nominal la o referintă de tip treaptă se realizează folosind funcţia *step*.

```
step(sys.NominalValue)
```

Obs 5. Comanda *usample* genereaza numărul impus de modele (alese arbitrar) din familia iniţială de modele incerte.

```
manysys = usample(sys,20);
size(manysys)
step(manysys)
```

Interpretarea incertitudinilor în domeniul discret

Pentru crearea unui model în spatiul stărilor cu parametrii incerţi, în domeniul discret se foloseste tot funcţia *ss* sau *uss*, specificând şi timpul de eşantionare.

```
sys = ss(A,B,C,D, 0.42)
```

Includerea timpului mort în sistemele cu parametrii incerţi

În implementarea curentă folosind funcţia *uss* timpul mort se omite.

Exemplu 11.

```
sys = rss(3,2,1);
sys.inputdelay = 1.3;
usys = uss(sys)
```

Pentru a păstra efectul timpului mort se foloseşte aproximarea Pade folosind funcţia *pade*.

pade(T,N) – returnează aproximarea Pade de ordinul N a timpului mort e^{-sT}.

Exemplu 12.

Fie sistemul de ordinul 2 descris de: `sys = tf(1,[1 1])*tf(1,[0.05 1]);`
Timpul mort considerat: `sys.inputdelay = 0.3;`
Gain cu domeniu de variaţie între 4 şi 6: `gain = ureal('gain',5);`
Sistemul cu parametrii incerţi: `usys = gain*pade(sys,4)`
Raspunsul sistemului la referinţă de tip treaptă: `step(usys,gain*sys,4,5)`

Determinarea răspunsului frecvenţial al modelelor cu parametrii incerţi

Răspunsul frecvenţial al modelelor cu parametrii incerţi se determină folosind funcţia *ufrd*.

Exemplu 13.

```
p1 = ureal('p1',10,'pe',50);
p2 = ureal('p2',3,'plusm',[-.5 1.2]);
p3 = ureal('p3',0);
A = [-p1 p2;0 -p1];
B = [-p2;p2+p3];
C = [1 0;1 1-p3];
D = [0;0];
sys = ss(A,B,C,D)

sysg = ufrd(sys,logspace(-2,2,100)) -
```

Pentru vizualizarea proprietăţilor se foloseşte funcţia *get*: get(sysg).
Pentru vizualizarea diagramei Bode a sistemului nominal se foloseşte funcţia *bode*: bode(sysg.nom).

Modificaţi valoarea nominală a parametrului incert “p1” cu 14 şi refaceţi diagram Bode a noului sistem nominal.

```
sysg.unc.p1.nom = 14
```

Exemplu 14. Modelarea sistemului de suspensie activă descris în Fig.1:

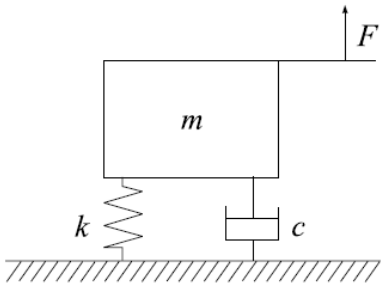

Fig.1 Sistemul de suspensie activă

Dinamica sistemului este descrisă de ecuaţia diferenţială: $m\ddot{x} + c\dot{x} + kx = u$.
unde: x – reprezintă deplasarea blocului de masă - m - din poziţia de echilibru, c – reprezintă constanta de amortizare, k – reprezintă constanta arcului iar u=F – reprezintă forţa care acţionează asupra blocului de masă m.

În Fig.2 este prezentată diagrama bloc a sistemului de suspensie activă.

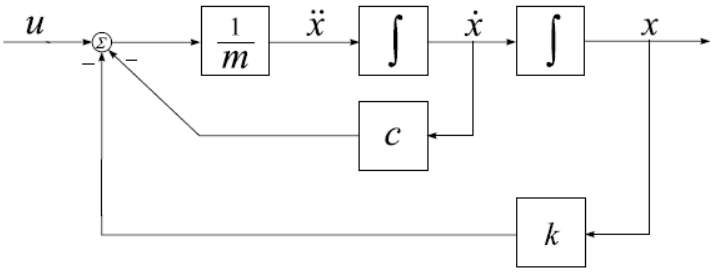

Fig.2 Diagrama bloc a sistemului de suspensie activă

Într-un sistem real, parametrii fizici m, c, k nu sunt cunoscuţi cu exactitate, dar se cunoaşte gama în care variază.
Astfel

$$m = \bar{m}(1 + p_m\delta_m)$$
$$c = \bar{c}(1 + p_c\delta_c)$$
$$k = \bar{k}(1 + p_k\delta_k)$$

unde $\bar{m} = 3, \bar{c} = 1$ şi $\bar{k} = 2$ reprezintă valorile nominale ale parametrilor m, c, k iar p_m, p_c, p_k şi $\delta_k, \delta_c, \delta_m$ reprezintă perturbaţiile posibile (relative) ale acestor parametrii. Se consideră , $p_m = 0.4, p_c = 0.2$ si $p_k = 0.3$ şi $-1 \leq \delta_k, \delta_c, \delta_m \leq 1$

Aceşti parametrii reprezintă incertitudini de 40% în masa blocului, de 20% a coeficientului de amortizare şi de 30% a coeficientului de rigiditate a arcului.

Diagrama bloc a sistemului de suspensie activă luând în considerare incertitudinile parametrilor este prezentată în Fig.3.

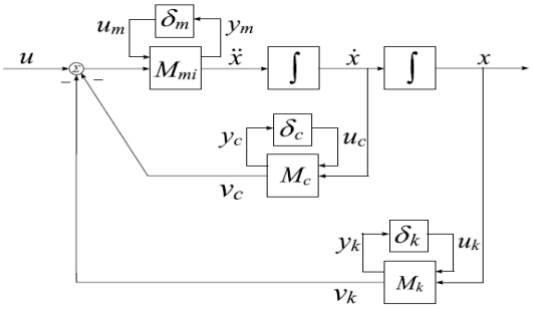

Fig.3 Diagrama bloc a sistemului de suspensie activă luând în considerare incertitudinile parametrilor

Se notează: $x_1 = x, x_2 = \dot{x} = \dot{x}_1, y = x_1 \ \ astfel\ incat\ \dot{x}_2 = \ddot{x} = \dot{\bar{x}}_1$

Astfel se obţine modelul sistemului în spaţiul stărilor care ţine cont incertitudinea parametrilor descris în Fig.4:

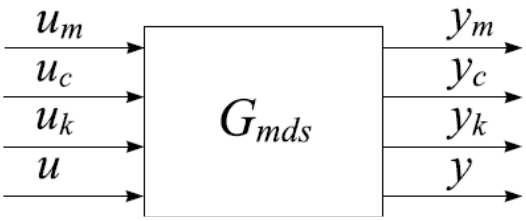

Fig.4

Se observă ca sistemul va avea 4 intrări (u_m, u_c, u_k, u), 4 ieşiri (y_m, y_c, y_k, y) şi 2 stări (x_1, x_2).

$$A = \begin{pmatrix} 0 & 1 \\ -\frac{\bar{k}}{\bar{m}} & -\frac{\bar{c}}{\bar{m}} \end{pmatrix} \quad B = \begin{pmatrix} 0 & 0 & 0 & 0 \\ -p_m & -\frac{p_c}{\bar{m}} & -\frac{p_k}{\bar{m}} & \frac{1}{\bar{m}} \end{pmatrix}$$

$$C = \begin{pmatrix} -\frac{\bar{k}}{\bar{m}} & -\frac{\bar{c}}{\bar{m}} \\ 0 & \bar{c} \\ \bar{k} & 0 \\ 1 & 0 \end{pmatrix} \quad D = \begin{pmatrix} -p_m & -\frac{p_c}{\bar{m}} & -\frac{p_k}{\bar{m}} & \frac{1}{\bar{m}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

m = 3;
c = 1;
k = 2;
pm = 0.4;
pc = 0.2;
pk = 0.3;

Se definesc matricile A, B, C, D.

Bibliografie:

1. D.-W. Gu, P. Hr. Petkov and M. M. Konstantinov, *Robust Control Design with Matlab*, Springer-Verlag London Limited, 2005
2. Robust Control Toolbox User Guide, www.mathworks.com