Tema 1

În cadrul primei teme vom analiza stabilitatea şi răspunsul în timp al mişcării verticale a unui elicopter:

0.5p 1. vom vedea cum putem determina dacă sistemul este stabil;

0.5p 2. vom vedea legătura dintre răspunsul în timp şi convoluţia cu funcţia pondere;

0.5p 3. vom împărţi acest răspuns pe componente (liber/forţat sau tranzitoriu/permanent);

0.5p 4. vom vedea care este efectul unui pol sau al unui zerou suplimentar în modelul mişcării verticale.

Pentru început, găsiţi-vă ID-urile în catalogul electronic de pe Moodle şi apelaţi:

```
>>P_tan = date_indiv_SS(ID);
```

pentru a vă obţine procesul personalizat cu care veţi lucra în temă. Descărcaţi, de asemenea, funcţia `ss_ci.p` pe care o vom folosi în această temă pentru a obţine răspunsul sistemului în condiţii iniţiale nenule.

1. Vom începe prin analiza de stabilitate.

0.1p a) Formaţi într-o variabilă numită `H` matricea Hurwitz a lui `P_tan`. Aveţi grijă să folosiţi chiar valorile numerice din proces, deoarece numere afişate în linia de comandă sunt trunchiate la ultimele câteva zecimale. Puteţi accesa polinoamele procesului prin `P_tan.num{1}` pentru numărător şi `P_tan.den{1}` pentru numitor.

0.2p b) Cu matricea `H` formată, memoraţi în variabilele `det1`, `det2` şi `det3` minorii principali ai lui `H`, corespunzători submatricelor de dimensiuni 1×1, 2×2 şi 3×3, respectiv. Calculul determinanţilor se poate face prin funcţia `det()`. Remarcaţi faptul că toate aceaste numere sunt pozitive, deci procesul este stabil.

O altă metodă de a analiza stabilitatea lui `P_tan` este de a verifica poziţionarea în planul complex a polilor acestuia.

0.2p c) Memoraţi într-un vector linie, numit `numitor`, numitorul lui `P_tan`. Polii funcţiei de transfer sunt chiar rădăcinile lui `numitor`. Memoraţi într-un vector coloană denumit `poli` rădăcinile acestuia cu ajutorul funcţiei `roots()`. Observaţi că toţi polii au partea reală negativă, confirmând din nou stabilitatea procesului.

2. Având stabilitatea garantată, putem să analizăm răspunsul în timp al procesului. Începem prin a declara un vector de timp:

```
>>t = (0:0.01:180)';
```

şi a calcula răspunsul sistemului la impuls.

0.1p a) Apelaţi `impulse()` pentru sistemul `P_tan` pe intervalul `t` şi memoraţi ieşirea într-un vector coloană numit `h_pondere`. Dacă figuraţi grafic acest semnal, veţi vedea că el tinde asimptotic la 0, ceea ce este o condiţie necesară, dar nu suficientă, ca sistemul să fie stabil.

Un alt răspuns uzual din domeniul timp este cel indicial, la intrare treaptă. El se poate calcula apleând `step()` pentru sistemul `P_tan` pe intervalul `t`.

0.1p b) Memoraţi rezultatul apelării într-un vector coloană numit `rasp_trp`.

Un mod alternativ de a calcula răspunsul indicial este cu ajutorul convoluţiei şi al funcţiei pondere.

0.1p c) Declaraţi o treaptă unitară de dimensiunea lui `t` într-un vector coloană numit `trp`, cel mai simplu fiind să folosiţi expresia `double(t>=0)`. Cu semnalul format, calculaţi convoluţia continuă dintre acesta şi `h_pondere`.

Aveţi grijă, deoarece funcţia `conv()` efectuează convoluţia în discret iar, pentru operaţia din timp continuu, rezultatul acestei funcţii trebuie să se înmulţească cu pasul de eşantionare al vectorului de timp, în cazul nostru `0.01`. Vom selecta doar prima parte a rezultatului, de lungimea lui `t`, deoarece suportul convoluţiei este suma suporturilor intrărilor, iar pe noi ne interesează doar perioada de timp cât avem la intrare semnalul treaptă unitară.

0.1p d) Memoraţi această primă parte într-un vector coloană numit `rasp_conv`.

0.1p e) Calculaţi într-o variabilă numită `norm_dif` norma infinit a diferenţei dintre `rasp_trp` şi `rasp_conv`. Valoarea foarte mică obţinută confirmă faptul că cele două metode de calcul dau aceleaşi rezultate. Puteţi să vă convingeţi suplimentar figurând grafic cele două semnale. Acestea vor fi suprapuse.

3. Pentru o analiză completă a răspunsului în timp, vor trebui investigate inclusiv cele două maniere de descompunere a răspunsului total: în răspuns permanent/tranzitoriu şi liber/forţat. Înainte de a începe, va trebui să ne generăm răspunsul total.

0.1p a) Apelaţi instrucţiunea `lsim()`, căreia îi vom da drept argumente (în ordinea indicată) sistemul `ss_ci(P_tan)`, treapta unitară `trp` declarată la exerciţiul anterior, vectorul de timp `t` declarat tot la exerciţiul 2 şi un vector de condiţii iniţiale, `x0 = [1 1 1]`. Memoraţi răspunsul total într-un vector coloană numit `rasp_tot`.

Precizare: Mereu când lucrăm în condiţii iniţiale nenule, procesul trebuie apelat cu funcţia `ss_ci()`. Denumirea vine de la `state-space` şi vom intra în mai multe detalii pe al doilea semestru.

Teorema Valorii Finale ne spune că răspunsul total al sistemului stabil $P_{tan}(s)$ va tinde asimptotic la $P_{tan}(0)$.

0.1p b) Calculaţi răspunsul permanent într-un vector coloană numit `rasp_perm` prin înmulţirea lui `trp` cu valoarea lui `P_tan` evaluat în 0, cu ajutorul funcţiei `evalfr()`.

0.1p c) Memoraţi în vectorul coloană numit `rasp_tran` diferenţa dintre `rasp_tot` şi `rasp_perm`.

Figuraţi pe câte un grafic `rasp_perm` şi `rasp_tran`. Deoarece `P_tan` este stabil, primul este mărginit iar al doilea tinde asimptotic la 0.

Răspunsul liber se poate calcula prin funcţia `initial()`.

0.1p d) Memoraţi într-un vector coloană numit `rasp_libr` ieşirea funcţiei atunci cand este apelată cu (în ordinea indicată) sistemul `ss_ci(P_tan)`, condiţiile iniţiale `x0` şi vectorul de timp `t`.

0.1p e) Calculaţi în vectorul coloană numit `rasp_fort` diferenţa dintre `rasp_tot` şi `rasp_libr`.

Comparaţi grafic `rasp_fort` şi `rasp_trp` de la exerciţiul anterior. Observaţi că sunt suprapuse. De asemenea, graficul lui `rasp_libr` tinde asimptotic la 0 deoarece `P_tan` este stabil.

4. În final, dorim să analizăm performanţele de regim tranzitoriu ale sistemului la intrare treaptă unitară.

0.1p a) Cu ajutorul funcţiei `stepinfo()`, vom determina valorile următoarelor variabile: `tc1` – timpul de creştere (rise time), `tt1` – timpul tranzitoriu (settling time), `tv1` – timpul de vârf (peak time) şi `sr1` – suprareglajul (overshoot), pentru sistemul `P_tan`.

Figuraţi răspunsul indicial al lui `P_tan`. Observaţi oscilaţiile importante din graficul răspunsului. Pentru a amortiza această mişcare nedorita în înclinarea verticală elicopterului, se folosesşte în practică un sistem auxiliar $P_{aux}(s) = \frac{1}{10s+1}$ ce acţionează pe comanda trimisă pe elice pentru a netezi mişcarea în plan veritcal.

0.2p b) Declaraţi o variabilă de tip `tf` numită `P_aux` ce are funcţia de transfer indicată, reapelaţi `stepinfo()` pentru produsul `P_tan * P_aux` şi calculaţi următorul set de variabile: `tc2` – timpul de creştere (rise time), `tt2` – timpul tranzitoriu (settling time), `tv2` – timpul de vârf (peak time) şi `sr2` – suprareglajul (overshoot).

Am văzut efectul unui pol suplimentar în modelul elicopterului. Acum vom investiga efectul unui zerou suplimentar. Funcţia `tf()` nu suporta direct modele improprii, dar se poate declara variabila complexă s.

0.2p c) Reapleaţi funcţia `stepinfo()` pentru produsul `P_tan * (tf('s') + 1)` şi calculaţi următorul set de variabile: `tc3` – timpul de creştere (rise time), `tt3` – timpul tranzitoriu (settling time), `tv3` – timpul de vârf (peak time) şi `sr3` – suprareglajul (overshoot).

Figuraţi pe acelaşi grafic răspunsul indicial al celor trei sisteme cu care am lucrat la exerciţiul curent. Observaţi efectul fiecărui element suplimentar comparativ cu răspunsul sistemului original.

Adaugati următoarea instrucţiune:

```
>>save('tema_ID.mat', 'H', 'det1', 'det2', ...
'det3', 'poli', 'h_pondere', 'rasp_trp', 'rasp_conv', ...
'norm_dif', 'rasp_tot', 'rasp_perm', 'rasp_tran', ...
'rasp_libr', 'rasp_fort', 'tc1', 'tt1', 'tv1', 'sr1', ...
'tc2', 'tt2', 'tv2', 'sr2', 'tc3', 'tt3', 'tv3', 'sr3');
```

la finalul script-ului MATLAB folosit pentru a rezolva tema, denumiţi-l `nume_prenume_grupa_tema1.m` şi încărcaţi fişierul `.m` pe Moodle până la data de **25.11.2021/23:59**, în secţiunea `Tema 1 - evaluare finala`.

Script-ul încărcat trebuie sa fie **rulabil fără erori** şi să genereze **un singur fişier `.mat` cu denumirea cerută**. Spre exemplu, Teo Rotaru de la 321AA cu ID-ul 734 din catalogul de pe Moodle va trebui să încarce `Rotaru_Teo_321AA_tema1.m`, care să genereze la rulare **doar** fişierul denumit `tema_734.mat`.

Opţional, puteţi încărca fişierul `.m` până la data de **18.11.2021/23:59**, în secţiunea `Tema 1 - evaluare pe parcurs`, pentru a primi un punctaj consultativ obţinut prin rularea soluţiilor prin checker. În acest fel, veţi mai avea aproximativ o săptămână pentru a ajusta eventualele nereguli.