

Intelligens Rendszerek

2018 tavasz

A laboratóriumi gyakorlatok témái

- Mobil robotok pályatervezése - földi robotok
- Mobil robotok pályatervezése - robotrepülők
- Genetikus algoritmusok
- Neurális hálózatok
- Globális helyzetmeghatározó rendszerek

Követelményrendszer

- A két zárthelyi átlaga alapján:
- Elért eredmény Érdemjegy
- 89%-100% jeles (5)
- 76%-88<% jó (4)
- 63%-75<% közepes (3)
- 51%-62<% elégséges (2)
- 0%-50<% elégtelen (1)

- 1.óra: Bevezetés és alapok
- 2.óra: Földi robotok pályatervezése 1.
- 3.óra: Földi robotok pályatervezése 2.
- 4.óra: Robotrepülőgépek
- 5.óra: Neurális hálózatok 1.
- 6.óra: Neurális hálózatok 2.
- 7.óra: Zh konzultáció, gyakorlaton 1. zh
- 8.óra: Genetikus algoritmusok 1.
- 9.óra: Genetikus algoritmusok 2.
- 10.óra: Műholdas navigációs rendszerek
- 11.óra: Zh konzultáció, gyakorlaton 2. zh (május 8-10)
- 12.óra: gyakorlaton egyetlen, meg nem írt zh pótlása (május 15-18)

1. Gyakorlat

Órai munka

- <http://mobil.nik.uni-obuda.hu>
- Mindenkinek saját Matlab jár! 😊 Tessék letölteni, telepíteni, használni!

Command Window

- Parancsok
- Több parancs -> .m file
- Sor végi ; jel
 - Ha nincs kint, végrehajtja és az eredményt kiírja
 - Ha kint van , az eredményt nem írja ki, de végrehajtja
- Komment %
- Indexelés 1-től
- Help help 😊

Műveletek

Tömbműveletek

+ összeadás

- kivonás

.* elemek szerinti
szorzás

./ elemek szerinti osztás

.^ elemek szerinti
hatványozás

Mátrixműveletek

+ összeadás

- kivonás

* szorzás

/ osztás

^ hatványozás

Relációs és logikai műveletek

<	kisebb
>	nagyobb
<=	kisebb vagy egyenlő
>=	nagyobb vagy egyenlő
==	egyenlő
≠	nem egyenlő (!!!!!!!)
&	ÉS
	VAGY
~	NEGÁCIÓ
xor	KIZÁRÓ VAGY

Munkaterület változók

```
>> clear           %munkaterület változóinak törlése  
>> clc            %parancs ablak törlése
```

Szöveg megjelenítése

```
>> disp('Szöveg')
```

Vektorok, tömbök

```
>> a=[1 2 3 4]
```

```
a =
```

```
1 2 3 4
```

```
>> b=[1 2 3 4]'
```

```
b =
```

```
1
```

```
2
```

```
3
```

```
4
```

Vektorok, tömbök

```
>> a=[1 2; 3 4]
```

```
a =
```

```
1 2  
3 4
```

```
>> b=[1 2; 3 4]'
```

```
b =
```

```
1 3  
2 4
```

Vektorok, tömbök

```
>> zeros(2)
```

```
ans =
```

```
0 0  
0 0
```

```
>> ones(2)
```

```
ans =
```

```
1 1  
1 1
```

```
>> eye(2)
```

```
ans =
```

```
1 0  
0 1
```

```
%egységmátrix
```

```
>> rand(2)
```

```
ans =
```

```
0.8147 0.1270  
0.9058 0.9134
```

Vektorok, tömbök

`a(:)` %minden elem – 1D

`a(:,:)` %minden elem – 2D

`a(1,1)` %1,1 elem

`a(:,1)` %1. oszlop (*minden x, 1. y*)

`a(1,:)` %1. sor

size - tömb dimenziói

length - vektor hossza

Sorozatok

```
>> a=[20:30]
```

```
a =
```

```
20 21 22 23 24 25 26 27 28 29 30
```

```
>> a(1:3)
```

```
ans =
```

```
20 21 22
```

```
>> b=[0:0.1:0.5]
```

```
b =
```

```
0 0.1000 0.2000 0.3000 0.4000 0.5000
```

Vezérlési szerkezetek

Az elágazás szintaxisa

if feltétel

utasítások

elseif feltétel

utasítások

else

utasítások

end

A for ciklus szintaxisa:

for i = 1:N

%for start:step:end

utasítások;

end

A while ciklus szintaxisa:

while feltétel

utasítások;

end

Feladat

- $x=[1,2,3;4,5,6;7,8,9]$
- Keressük meg a min és max értéket, adjuk meg ennek index értékét!

Feladat

- $x=[1,2,3;4,5,6;7,8,9]$
- Keressük meg a min és max értéket, adjuk meg ennek index értékét!
 - $[idx,val]=\min(x(:))$
 - $[idx,val]=\max(x(:))$
 - Idx: sorfolytonos index-szel tér vissza, val: érték

Feladat

- Tessék kiírni a következő sorozatot:
- x első eleme, x utolsó eleme, x második eleme, x utolsó előtti eleme, stb...
- Addig, amíg a két ciklusváltozó helyet nem cserél a tömbben.
- For ciklussal érdemes, sorra és oszlopra is tessék hivatkozni ($x(i,j)$) 😊

Feladat

```
map=zeros(50,50);
```

```
robot=[25,25];
```

Tessék a robot pozícióját a térképen megadni (1-es értékkel)

(értékkadás `map(25,25)` koordinátájára, a `robot` változót felhasználva)

Feladat

```
map=zeros(50,50);
```

```
robot=[25,25];
```

Tessék a robot pozícióját a térképen megadni (1-es értékkel)

(értékkadás `map(25,25)` koordinátájára, a `robot` változót felhasználva)

```
map(robot(1),robot(2))=1;
```


Megjelenítés:

`imagesc(map)`

- Opcionális érték skála:

`colorbar;`

Feladat

- Tessék a pálya szélét bekeretezni
- Az akadály értéke legyen 2!

Feladat

- Ha kész, tessék a robotot léptetni!
- 10 pozíciót fel, majd 10-et jobbra!
- Automatikusan (for cikluson belül) tessék a rajzolást is frissíteni
- Léptetés lassítása:
 - `pause(0.1); %100ms`

```
clear all;
close all;
map=zeros(50,50);
robot=[25,25];
map(robot(1),robot(2))=1;
imagesc(map)

for i=1:10
    map(robot(1),robot(2))=0;
    robot(1)=robot(1)-1;
    map(robot(1),robot(2))=1;
    imagesc(map);
    pause(0.1);
end
for j=1:10
    map(robot(1),robot(2))=0;
    robot(2)=robot(2)+1;
    map(robot(1),robot(2))=1;
    imagesc(map);
    pause(0.1);
end
```

Feladat

- A robot környezetét (a környező 8 cellát) vegyük körbe fallal (2-es érték!)

Feladat

- A robot környezetét (a környező 8 cellát) vegyük körbe fallal (2-es érték!)

```
for (k=-1:1)
    for (l=-1:1)
        if ~and(k==0 , l==0)
            map(robot(1)+k,robot(2)+l)=2;
        end
    end
end
imagesc(map);
```

Feladat

- Számítsa ki, hogy milyen messzire került a robot a kiindulási pozíciójától!
- Tipp:
 $n = \text{norm}(v)$ returns the 2-norm or Euclidean norm of vector v .