CISC 3250
Systems Neuroscience

Matlab, part 3:
Functions

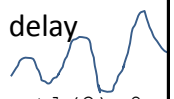
Professor Daniel Leeds
dleeds@fordham.edu
JMH 332

Similar goal, similar code

Goal: compute motion with 2 time-point delay

```
targ=5;
actl(1)=0;
actl(2)=0;
actl(3)=0;
for n=3:31,
 sens(n)=actl(n-2);
 mv=0.5*(targ-sens(n));
 actl(n+1)=actl(n)+mv;
end;
```

4 time-point delay



```
targ=5;
actl(1)=0;  actl(2)=0;
actl(3)=0;  actl(4)=0;
actl(5)=0;
for n=3:31  5:33,
 sens(n)=actl(n-4);
 mv=0.5*(targ-sens(n));
 actl(n+1)=actl(n)+mv;
end;
```

Sens(t) = Actual(t-4)
Sens(1) = Actual(1-4)=Actual(-3)
Actual(-3) <-> actl(1)

Similar goal, similar code

Goal: compute motion with 2 time-point delay

```
targ=5;
actl(1)=0;
actl(2)=0;
actl(3)=0;
for n=3:31,
 sens(n)=actl(n-2);
 mv=0.5*(targ-sens(n));
 actl(n+1)=actl(n)+mv;
end;
```

d time point delay

```
d=4; OR d=1;  OR d=10;
targ=5;
actl=zeros(d+29,1);
for n=(d+1):(d+29),
 sens(n)=actl(n-d);
 mv=0.5*(targ-sens(n));
 actl(n+1)=actl(n)+mv;
end;
```


Functions as reusable code

Function definition in modelMotion.m

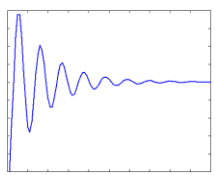
.mat saves data
.m saves Matlab commands

```
function actl=modelMotion(d)

targ=5;
actl=zeros(d+29,1);
for n=(d+2):(d+29),
 sens(n)=actl(n-d);
 mv=0.5*(targ-sens(n));
 actl(n+1)=actl(n)+mv;
end;
```

Use function

```
actl=modelMotion(2);
plot(actl)
```



Function syntax

- Define function in separate text file:
  `nameOfFunc.m`
- In file, define function:

```
function output=nameOfFunc(in1, …, inN)

Commands-to-perform
```

```
function actl=modelMotion(d)

targ=5;
actl=zeros(d+29,1);
for n=(d+1):(d+29),
 sens(n)=actl(n-d);
 …
 actl = …
```

5

Editing a new .m file

- In Matlab: edit funcName
- In Windows: Notepad
- In Mac: TextEdit (save as **Simple Text** file)
- In Linux: vim, emacs

- Save in the directory where you store .mat data files

6

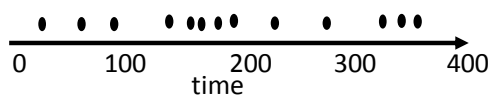
Simple example

- Write function `sumBig` that takes the vector `v` and the number `n`, and returns the sum of all elements in `V` greater than `n`
- Example: `sumBig([3,0,2,6,1,7],4)` returns `13`
- Example: `sumBig([20,-10,14,8,2],10)` returns `34`

7

More complex example

- Write function `computeRate` that takes the vector of spike outputs `S` and the window size `w`, and returns the spike count over windows of size `w`



- Example: `computeRate(S,400)` returns `[13]`
- Example: `computeRate(S,200)` returns [8, 5]
- Example: `computeRate(S,100)` returns [3, 5, 2, 3]

8

Simple example **ANSWER**

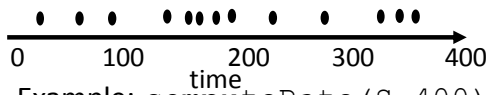
- Write function `sumBig` that takes the vector `v` and the number `n`, and returns the sum of all elements in `V` greater than `n`
- Example: `sumBig([3,0,2,6,1,7],4)` returns `13`

```
function finalSum=sumBig(V,n)
bigNums=find(V>n);
finalSum=sum(V(bigNums));
```

9

More complex example **Answer**

- Write function `computeRate` that takes the vector of spike outputs `S` and the window size `w`, and returns the spike count over windows of size `w`



- Example: `computeRate(S,400)` returns `[13]`

```
function countVec=computeRate(S,w)
countVec=zeros(1,length(S)/w);
for t=1:length(rateVec),
  countVec(t)=sum(S((t-1)*w+1:t*w));
end;
```

10