Introduction to Matlab

Plots

Pouyan R. Fard and Dario Cuevas

7. Januar 2016

Plot command

- plot(x,y), where x and y are vectors of the same size. For example:
  - `x=1:0.1:10; y=sin(x); plot(x,y)`
  - `x1=-pi:0.1:pi; plot(x1, 2*cos(x1))`
  - `x2=1:10; plot(x2,x2.^2,'red')` % or blue, black, b, r, g, ...
  - `x3=0:0.1:pi/2; plot(x3,arctan(x3),'b*')` % color + marker
  - `x4=-10:10; plot(x4, heaviside(x4), '-.');`
- To see a list of markers and colors: `help plot`
- You can put more than one function in a plot:
  `plot(x,sin(x),'g',x,cos(x),'red)`

Useful commands

The following commands can be executed after a plot:

title	Sets title for the plot figure	title('Average population activity')
xlabel/ylabel	Sets the label to each axis	xlabel('time')
legend	Creates a floating legend	legend('First subject','Second subject')
axis	Changes the range of the plot	axis([0 100 0 10])
axis equal/square/ tight	Changes the aspect ratio of the plot	-
grid on/off	Turn the grid on or off	-
LineWidth	Changes the width of the plot line	plot(x,y, 'linewidth',3)

The are many more properties. For more advanced shaping, see the `get` and `set` commands.

Multiple plots in one command

To draw multiple plots in one set of axes, with one command, use: `plot(x1,y1,x2,y2,...)`, where (x1,y1) are one set of data, (x2,y2) another set, etc. Examples:

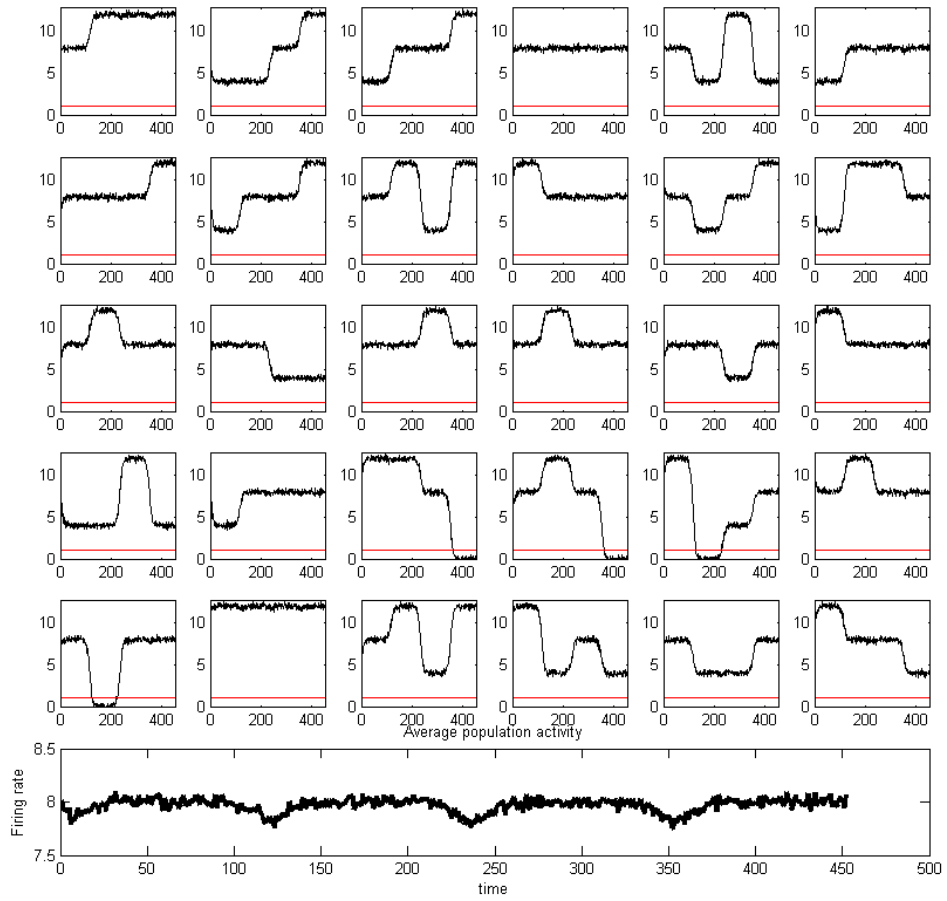
- `x=1:0.1:5; y1=exp(x); y2=log(x); plot(x,y1,x,y2);`
- ```
  x1 = -pi:0.1:pi; y1 = cos(x1);
  x2 = -pi/2:0.05:pi/2; y2 = sin(x2);
  plot(x1,y1,'-.ored',x2,y2,'-*green')
  ```

Alternatively, use the `hold on` and `hold off` commands:

- ```
  x=1:0.1:5; y1=exp(x); y2=log(x);
  plot(x,y1); hold on; plot(x,y2); hold off
  ```
- ```
  x1 = -pi:0.1:pi; y1 = cos(x1);
  x2 = -pi/2:0.05:pi/2; y2 = sin(x2);
  plot(x1,y1,'-.ored'); hold on;
  plot(x2,y2,'-*green'); hold off
  ```
  ```

Subplots

To draw many plots in one Figure window, use the subplot(rows,cols,plotnr) command. This will create one figure with many plots inside, in a 2D arrange. Example with `subplot(6,6,x)`:



Subplot code

```
figure(1)
subplot(2,1,1)
x = linspace(-2*pi,2*pi,5);
plot(x, cos(x))
title('Cosine with 5 points')
subplot(2,1,2)
x = linspace(-2*pi, 2*pi,10);
plot(x, cos(x))
title('Cosine with 10 points')

% Another one
figure(2)
for i=1:4
    x = linspace(0,2*pi,5*i);
    subplot(2,2,i)
    plot(x,sin(x));
    title(sprintf('Sine with %d points',i*5));
end
```

Exercise

Import the data file FiringRates.mat. Each row is a different neuron; each column a time point.

1. Plot the firing rate for each neuron in a separate subplot, such that all are plotted on their own „square“. You can decide the exact arrangement of squares; however, your code should work for any number of neurons, not just the 30 in the .mat file.
2. The lines should be black
3. Every subplot should have the same range. They should all go from zero to the maximum in the data file on the Y axis.
4. Add axis labels to the first subplot. They should be 'time' and 'firing rate'.
5. You should include a red line at height y=1 (see the picture). Use the command `line` for this.
6. Calculate the average population activity (of all 30 neurons) and plot it at the bottom of the figure, as shown in the picture; use a thick, black line. Give this plot a fitting title and axis labels 'time' and 'Population firing rate'.
7. Getting into the advanced stuff. Using `get` and `set`, remove the axis ticks on all the subplots except the first one.