Exercises Unit 3. M-file Programming

Working period: Fifth and sixth weeks
Due date: 7 April 2013

Submit only one file **`my_name_E3.pdf`** with the MATLAB code that solves the following exercises. This file must include the results and your comments as well. Please, use the exercises template available in the virtual campus. Should there be any questions about the exercises, please feel free to contact the teacher for clarification

1. Scripts

Exercise 1. Input (keyboard) and output (command window). Write the MATLAB commands that implement the following game: The player has 7 chances to guess an integer number **`n`** between 1 and 100. For every trial **`x`**, MATLAB says to the player (a) if the unknown number **`n`** is greater than **`x`** or less than **`x`**, and (b) how many chances remain.

To do so, write the statements that at least perform the following actions:

1) Use the function **`input`** to ask the player's name and store it in the char variable **`name`**.

2) Then use function **`disp`** to call the name's player and tell him that he has 7 chances to guess an integer number between 1 and 100.

3) Generate a random integer number **`n`** between 1 and 100 (**`rand`**, **`fix`** or **`ceil`**).

4) Implement a loop **`while...end`** (to be executed no more than 7 times) with the following actions: ask the player the guessed number **`x`** (**`input`**), compare it to **`n`** (**`if...elseif...else...end`**), tell the player if **`n`** is greater than or less than **`x`** and tell the player how many chances still remain (**`disp`**, **`num2str`**).

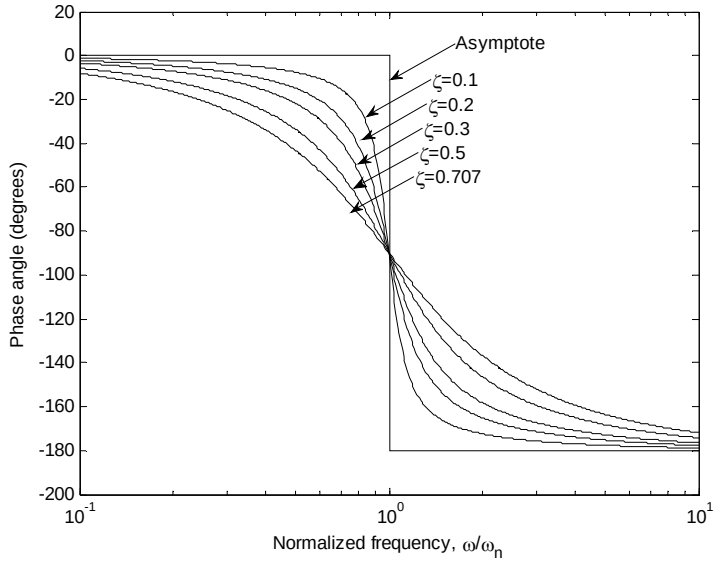
5) When the game is over, MATLAB must say to the player if he has won or not (**`disp`**). In this second case, MATLAB gives the correct number **`n`** (**`num2str`**).

Test the script (note that with a good strategy it is possible to win always): play with the game two times (win in one and fail in the other one). □

Exercise 2. Gain and phase normalized curves. Plot and label the normalized gain (in dB) and phase (in degrees) curves corresponding to the frequency response of the second order system:

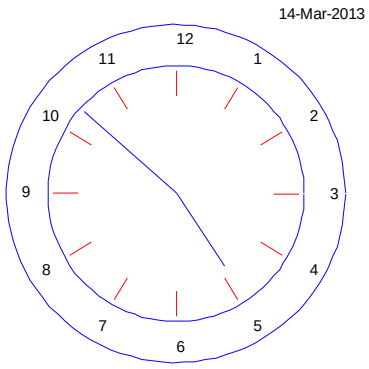
$$H(j\omega) = \frac{1}{(j\omega)^2 + 2\zeta(j\omega) + 1}$$

Next figure shows a suggestion for the phase plot (functions **`logspace`**, **`bode`** or **`freqs`**, **`abs`**, **`angle`**, **`log10`**, **`hold`**, **`for...end`**, **`text`**, **`annotation`**, **`ylabel`**, **`xlabel`**).



Optional (for a higher grade): Repeat for a first order system. □

Exercise 3. Date and hour. We want to present the current date and hour like in the figure below (note: it is not necessary to implement any type of movement or on-line refreshment. It is enough with a "photo" of the current time).



Write the statements to:

1) Represent the twelve hour lines. To do so, find the angles θ that correspond to each hour **`h=1:12`** and plot all lines (**`plot`**, **`hold on/off`**) by means of a loop (**`for...end`**). Label each line (**`text`**, **`int2str`**).

2) Call function **`date`** and show its result with function **`text`**.

3) Find the current hour and minutes with **`clock`**. Note: To transform 13^h, 14^h,... into 1^h, 2^h,... , you can divide the first ones by 12. Notice that the second ones are directly the division remainder (**`rem`**).

4) Plot the minute arrow. To do so, compute the angle corresponding to the minutes (**`plot`**).

5) Plot the hour arrow. To do so, compute the angle corresponding to the current hour and minutes (**`plot`**). □

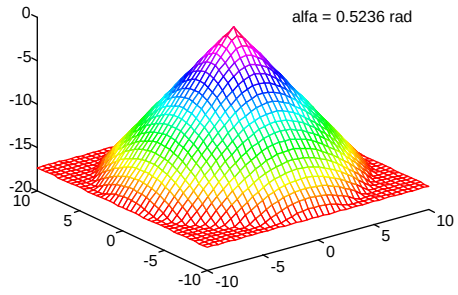
2. Functions

Exercise 4. Input and output arguments in functions. Create a function **`cones`** with a help message given by:

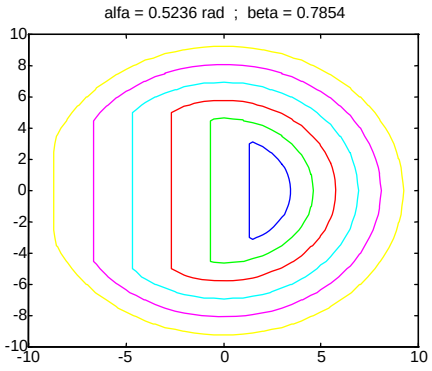
```
» help cones

CONES:  function that plots cones

z=cones(alpha)        Compute cones of angle alpha (degrees)
                      To plot them use mesh(z) or contour(z)

z=cones(alpha,beta)   Cones of angle alpha and cutting angle beta (degrees)
                      To plot them use mesh(z) or contour(z)

cones(alpha)            Direct 3D plot
cones(alpha,beta)

cones(alpha,'c')        Direct contour plot
cones(alpha,beta,'c')
```

Type the following two commands to check the correct execution of the function

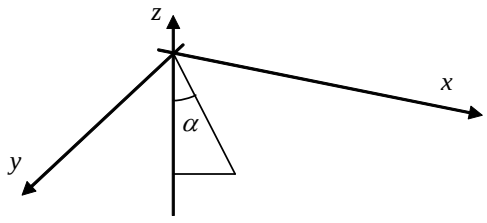
```
» cones(30)
```



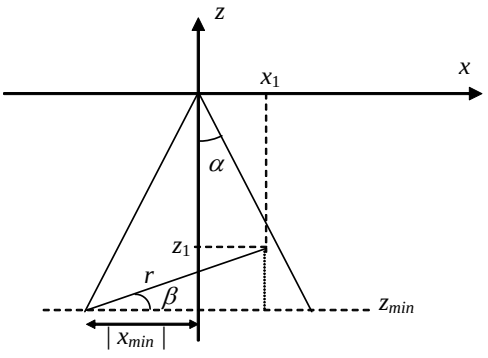

```
» cones(30,45,'c')
```



To obtain the cones, a possible way to do it is the following:



$$\left.\begin{array}{l}\operatorname{sen}\alpha = \sqrt{x^2+y^2}\\ \cos\alpha = z\end{array}\right\} \Rightarrow z = -\frac{\sqrt{x^2+y^2}}{\operatorname{tg}\alpha}$$



$$\left.\begin{array}{l} r\cos\beta = |x_{min}| + x_1\\ r\operatorname{sen}\beta = z_1 - z_{min}\end{array}\right\} \Rightarrow z_1 = \operatorname{tg}\beta\cdot\left(|x_{min}| + x_1\right) + z_{min}$$

Cone plot:

Write the statements that implement the following actions:

1) Generate a vector **`x=-10:0.5:10;`** (41 samples) and a vector **`y=x;`**.

2) Use **`meshgrid`** to obtain **`xx`** and **`yy`**. Use them to obtain the **z** values for the cone.

3) Check that the central column of **`xx`** (the 21st column) contains the central value of **`x`**. Idem for the central row of **`yy`**. In order to plot the "floor", assign the value **`z(1,21)`** to all values **z** which original value is less than **`z(1,21)`** (alternatively, you can assign the value **`z(21,1)`** to all **z**'s which original value is less than **`z(21,1)`**). (**`find`**, **`for...end`**)

4) Plot the cone with **`mesh`** and the contour plot with **`contour`**. Label the plot with a title (**`title`**) indicating the α value in radians (**`num2str`**).

Cone cut:

1) Write the statements that implement the cone cut.

2) Use **`nargin`** and **`nargout`** (or **`varargin`** and **`varargout`**) to control the function result for the different usages defined in the help message. □

Exercise 5. Polar plot in dB.

1) Plot the following expression in polar coordinates (**`linspace`**, **`abs`**, **`sin`**, **`polar`**):

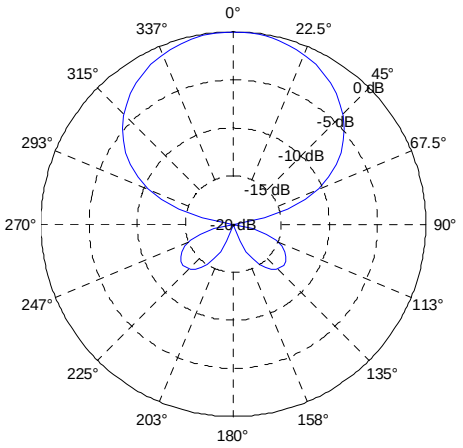
$$y = \left| \frac{\sin(2\theta)}{2\theta} \right| \quad \text{para} \quad -\pi \le \theta \le \pi$$

2) Plot $20\log_{10}(y)$ depending on θ (**`log10`**, **`plot`**).

3) Create a function with name **`polardB.m`** to generate polar plots of $20\log_{10}(y)$ exactly as the plot shown in the figure. Input arguments must be "**`theta`**", "**`y`**" y "**`min_db`**". The later one stands for the "central" dB value of the plot. The figure shows the result for **`min_db=-20`**.

4) Run the function for the case **`min_db=-40`**, as well.

(Functions: **`linspace`**, **`exp`** o **`cos`**/**`sin`**, **`plot`**, **`axis`**, **`text`**, [**`patch`**], **`hold`**)



3. Functions that use other functions as input arguments

Exercise 6. Solving equations.

1) We want to solve the following equation $\frac{\phi - \text{sen}\,\phi}{2} = \frac{A}{r^2}$ where A = 0.0472 and r = 2. To do so create a function containing the equation to be solved, $y = \frac{\phi - \text{sen}\,\phi}{2} - \frac{A}{r^2}$, with input argument ϕ and output argument y. Then, use **`fzero`** or **`fsolve`**.

2) Optional (for a higher grade): Consider the two curves defined by

$$p_2 - c_2 = \frac{1}{\alpha\left(1-\left(\frac{e^{x_2\beta-\alpha p_2}}{e^{x_2\beta-\alpha p_2}+e^{x_1\beta-\alpha p_1}}\right)\right)}$$

$$p_1 - c_1 = \frac{1}{\alpha\left(1-\left(\frac{e^{x_1\beta-\alpha p_1}}{e^{x_2\beta-\alpha p_2}+e^{x_1\beta-\alpha p_1}}\right)\right)}$$

where $c^T = \begin{pmatrix} c_1 & c_2 \end{pmatrix} = \begin{pmatrix} 2.8724 & 2.8839 \end{pmatrix}$, $x^T = \begin{pmatrix} x_1 & x_2 \end{pmatrix} = \begin{pmatrix} 20 & 19 \end{pmatrix}$, $\alpha = 0.172188$ and $\beta = 0.4$. In both cases, the plot coordinates are (p_1, p_2). Compute the intersection point between the two curves and check the result plotting them with **`ezplot`**. □

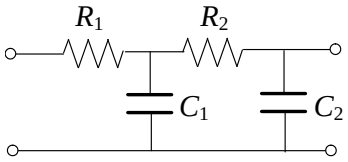
Exercise 7. Chaotic attractor. The state equations for a chaotic attractor are given by the Lorentz equations:

$$\begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \end{pmatrix} = \begin{bmatrix} -\beta & 0 & y_2 \\ 0 & -\sigma & \sigma \\ -y_2 & \rho & -1 \end{bmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

1) Create a function containing the Lorentz equations. Take $\sigma = 10$, $\rho = 28$ and $\beta = 8/3$.

2) Use **`ode23`** with initial conditions $(0\ 0\ \varepsilon)^T$, (use the pre-specified variable **`eps`**), $t_{initial}$=0s and t_{final}=100s. Plot the result in 3D with **`comet3`**. □

Exercise 8. Filter parametric optimisation. Consider a low pass filter with ω_b = 1000rad/s. A possible realisation is shown in next figure where $C_1 = C_2 = 1nF$.



The filter transfer function is

$$H(s) = \frac{\frac{10^{18}}{R_1 R_2}}{s^2 + \left(\frac{10^9}{R_2} + \frac{10^9}{R_1}\right)s + \frac{10^{18}}{R_1 R_2}}$$

We want to find the values for R_1 and R_2 which make the filter frequency response as close as possible to the ideal frequency response. The performance is measured using the index $J = \sum e_i^2$ where e_i is the mismatch between real attenuation and ideal attenuation:

$$e_i = \left|A(\omega_i) - 1\right|, \text{ for } \omega_i < 1000 \quad \text{and} \quad e_i = \left|A(\omega_i)\right|, \text{ for } \omega_i \geq 1000$$

where $A(\omega)=|H(j\omega)|$ is the frequency response magnitude.

1) Create a function with input argument $[R_1 , R_2]$ and output argument J. Note**:** Compute the error values e_i at frequencies ω_i = 0, 100, 500, 800, 900, 1000, 1100, 1200 y 1500. (**`function`**, **`freqs`**, **`for...end`**, **`if...else...end`**, **`abs`**).

2) Numerical optimisation: Use **`fminsearch`** to compute the optimal values for R_1, R_2 and J.

3) Optional (for a higher grade): Graphical optimisation: Plot (3D) the performance index J as a function of R_1 and R_2 (**`meshgrid`**, **`mesh`**). Obtain the contour plot (**`contour`**, **`clabel`**), and find the pair (R_1,R_2) which minimises J.

4) Optional (for a higher grade): Plot the modulus of the ideal filter and compare it to the modulus of the obtained filter. □

4. Other applications

Exercise 9. Audio processing.

1) Record your voice saying any sentence in a wav audio file with any program you wish.

2) Load them to MATLAB with **`wavread`** and listen to them with **`sound`**.

3) Compute and plot the spectrum. To find it use any method of the ones presented in Unit 3 (periodogram, Blackman-Tukey, Welch, Barlett,…). Identify your pitch frequency (i.e., the frequency of the first main peak).

4) Optional (for a higher grade): Perform any type of signal processing (filtering, scrambling,…) and listen to the result. □

Exercise 10. PLL dynamics.

A phase lock loop (PLL) presents a dynamic behavior described by means the following differential equation, $\ddot{\varphi}_c + \dot{\varphi}_c + \sin\varphi_c = \sin\varphi_r$ where φ_r is the reference phase and φ_c is the controlled phase. Let us consider $\varphi_r = 0$.

1) Solve the equation for different combinations of the initial conditions $\varphi_c(0)$, $\dot{\varphi}_c(0)$ ranging between -2π and 2π and plot all the trajectories in a plane φ_c, $\dot{\varphi}_c$ (function **`ode23`**, **`linspace`**, **`for…end`**)

2) Use the figure menu bar (Insert → Line) to add the lines that define the cycle slipping effect.