

Лабораторная работа №4

Последовательность базовых заданий

1. Создайте скрипт с названием *lab4_ASP_SurnameN.m* в директории */Documents/Matlab*. Убедитесь, что в окне интерфейса *Current Folder* появился Ваш скрипт.
2. Создайте **первую секцию** с названием *Filters design*. Создайте массив частот, 2 переменные со значением порядка фильтра и частотой дискретизации:

```
freqArray = [31, 62, 125, 250, 500, 1000, 2000, 4000,  
            8000, 16000];  
order = 1024; % должен быть четным  
fS = 44100;
```

3. Используя функция *randi* выберите три случайных набора коэффициентов из банка фильтров и запишите результат в переменную *bTmp*.
4. Используя функцию *freqz* получите частотную характеристику. Пример использования:

```
[H(i, :), w()] = freqz(bTmp(i, :), :), 1, order);
```

5. Создайте анонимную функцию перевода в дБ по следующей формуле:

$$x_{db} = 20 \cdot \log_{10} x$$

6. Переведите в дБ модуль *H*, полученную в 4 пункте. Также преобразуйте вектор *w* разделив на π и помножив на $fS/2$.
7. Создайте **вторую секцию** скрипта с названием *Graph with standard functions*

Постройте график зависимости модуля частотной характеристики из п.6 от преобразованной частоты (из п.6) в графическом окне. Отформатируйте графики по следующим пунктам:

- первый график: непрерывная линия, черный цвет, без маркеров, толщина линии = 1;
- второй график: линия из точек, синий цвет, маркер *, толщина линии = 1;
- третий график: линия из точек и тире, красный цвет, маркер = квадрат, толщина линии = 1;
- вертикальная и горизонтальная сетка;
- подпись оси x: *f, kHz*, размер шрифта: 16;
- подпись оси y: *|H|, dB*, размер шрифта: 16;
- общая подпись к графикам: *Filters numbers: №, №, №*, размер шрифта: 16;

- добавьте легенду для графиков с соответствующими подписями
- пределы значений по оси x: 0 и 21000
- пределы значений по оси y: -60 и 10
- добавьте черточки-тики на оси: 2000, 8000, 16000
- добавьте подписи к тикам на оси x: 2 kHz, 8 kHz, 16 kHz.

Вопросы для самоконтроля. Сколько на самом деле графических объектов создал *Matlab* при построении графиков в п.7?

8. Создайте **третью секцию** скрипта с названием *Graph with changing object properties*. Повторите все действия из п.7, только при форматировании **нельзя** использовать стандартные функции. Форматирование необходимо осуществлять путем изменения свойств графических объектов. Измените свойство *Position* объекта класса *Figure*, чтобы данное графическое окно отображалось в левом нижнем углу экрана.

Вопросы для самоконтроля. Сколько графических объектов отображается в *Workspace*? Сколько на самом деле графических объектов создал *Matlab* при построении графиков в п.8?

9. Создайте **четвертую секцию** скрипта с названием *Graph with no formatting*. Постройте три графика из п.7 в одном графическом окне без форматирования. Измените свойство *Position* объекта класса *Figure*, чтобы данное графическое окно отображалось в правом нижнем углу экрана. В данном пункте Вам понадобятся только функции *figure*, *plot*. С помощью интерфейса *Matlab* отформатируйте график в соответствие с п.7. Сгенерируйте автоматически функцию для редактирования графика.

10. Создайте **пятую секцию** скрипта с названием *Graph with autofunc*. Вызовите функцию, созданную в п.9.

Вопросы для самоконтроля. Совпадают ли полностью графики, построенные в п.7, 8, 10? Если нет, объясните почему, вернитесь к данным пунктам и проверьте их выполнение.

11. Запустите скрипт целиком. Должно открываться четыре графических окна. Если количество графических окон меньше четырех, добавьте вызов функции *figure* в необходимых секциях скрипта.

12. Создайте **шестую секцию** с названием *Filtering of signals*. Прочитайте файл *song.mp3* с помощью функции *audioread* в массив *signal*. Добавьте к считанному аудиосигнала гармонический сигнал на частоте 17 кГц.

13. Используя функцию *pspectrum* получите спектр полученного сигнала. Пример использования:

```
[pxx, f] = pspectrum(signal, fs)
```

14. Используя функция `FiltrenigBanks` из лаб.3 отфильтруйте полученный сигнал так, что `gain` последнего фильтра был равен 0 (остальные 1). Постройте спектр до и после фильтрации на одном графике используя `subplot`, подпишите графики, оси любым из способов, рассмотренных выше.
15. Создайте **седьмую секцию** *Spectrogram*, используя функцию `spectrogram`, получите спектрограмму сигнала, а также временную и частотную сетку:

```
[s, f, t] = spectrogram(signal(:, 1), 2^10, [], freqArray, fs);  
[F, T] = meshgrid(f, t);
```
16. Постройте трехмерную спектрограмму используя функцию `mesh(...)`. Подпишите график, оси и сделайте логарифмическим масштаб по оси `x`.

Дополнительные задания

1. Создайте **новую секцию** *Stream sound spectrum*. В секции создайте системный объект для чтения:

```
fileReader = dsp.AudioFileReader('song.mp3');
```

Используя цикл `while` и условие `isDone(fileReader)` осуществите:

- чтение порции данных `audioData = fileReader();`
- вычисление спектра с использованием `pspectrum`, результат на каждой итерации сохраняются в матрицу `sigSpecMat`; (Прим: нужно считать спектр одного из каналов);
- используйте счетчик `cnt` для определения количества итераций;
- используйте `break` для **выхода** из цикла, если `cnt` **больше 500**.

После цикла создайте переменную `timeStep` следующим образом:

```
timeSteps = (1:cnt-1)*length(audioData)/fs;
```

2. Используя функции `meshgrid` и `mesh` постройте спектр в зависимости от времени `timeSteps`. Нанесите оси, название и пр. одним из выше рассмотренных способов. Установите предел по частоте в 2000.