

# Занятие №2.

Обучение для выполнения НИР 5 курса,  
подготовка к выполнению курсовой  
работы

# План занятия:

-Разбор домашнего задания прошлого занятия;

% не забыть пожуричь обленившихся и похвалить отличившихся;

-Перемножение матриц, операторы;

-Пример.

# Домашнее задание прошлого занятия:

1. Определите показания второго датчика в момент времени, когда первый достигает своего максимального по амплитуде значение.
2. Поменяйте на графике цвет показаний второго датчика с зеленого на красный.
3. Центрирование относительно нуля.

Найдите среднее арифметическое значение для показаний каждого из датчиков.  
Вычтите из показаний датчиков его среднее значение.

Подсказка: используйте одну из функций, приведенных на предыдущем слайде.

4. В семинаре присутствует ошибка при определении максимальной перегрузки, показанной датчиком. В чем она заключается и как её устранить?

Подсказка: используйте одну из функций, приведенных на предыдущем слайде.

5. Отфильтруйте сигналы с датчиков при помощи осреднения:

Показания датчика для каждого момент времени = сумма показаний в предыдущие 5 моментов (замеров) и 5 последующих / 11.

(для первых 5 показаний и последних пяти показаний такую функцию можно не применять).

Подсказка: может потребоваться цикл for

Вывести результаты на график.

# Домашнее задание прошлого

## ЗАНЯТИЯ:

1. Определите показания второго датчика в момент времени, когда первый достигает максимальной амплитуды.

Решение: необходимый код приведен ниже. В зависимости от того, как Вы находили ind1 ответ может быть разным. Правильный ответ -1,0717 (см п.4)

```
>> Dat2(ind1)
```

```
ans =
```

```
-1.0717
```

```
'
```

```
>> Dat2(ind1)
```

```
ans =
```

```
0.1195
```

2. Поменяйте на графике цвет показаний второго датчика с зеленого на красный.

Решение: необходимый код приведен ниже. Нужно было написать 'r'

```
plot(T, Dat1, 'b', T, Dat2, 'r');
```

# Домашнее задание прошлого

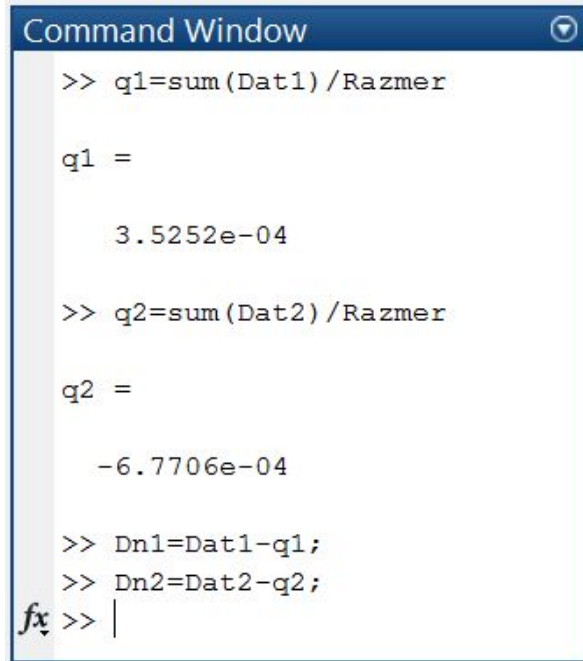
## ЗАНЯТИЯ:

3. Центрирование относительно нуля.

Найдите среднее арифметическое значение для показаний каждого из датчиков.

Вычтите из показаний датчиков его среднее значение.

Решение:



```
Command Window
>> q1=sum(Dat1)/Razmer
q1 =
    3.5252e-04
>> q2=sum(Dat2)/Razmer
q2 =
   -6.7706e-04
>> Dn1=Dat1-q1;
>> Dn2=Dat2-q2;
fx >> |
```

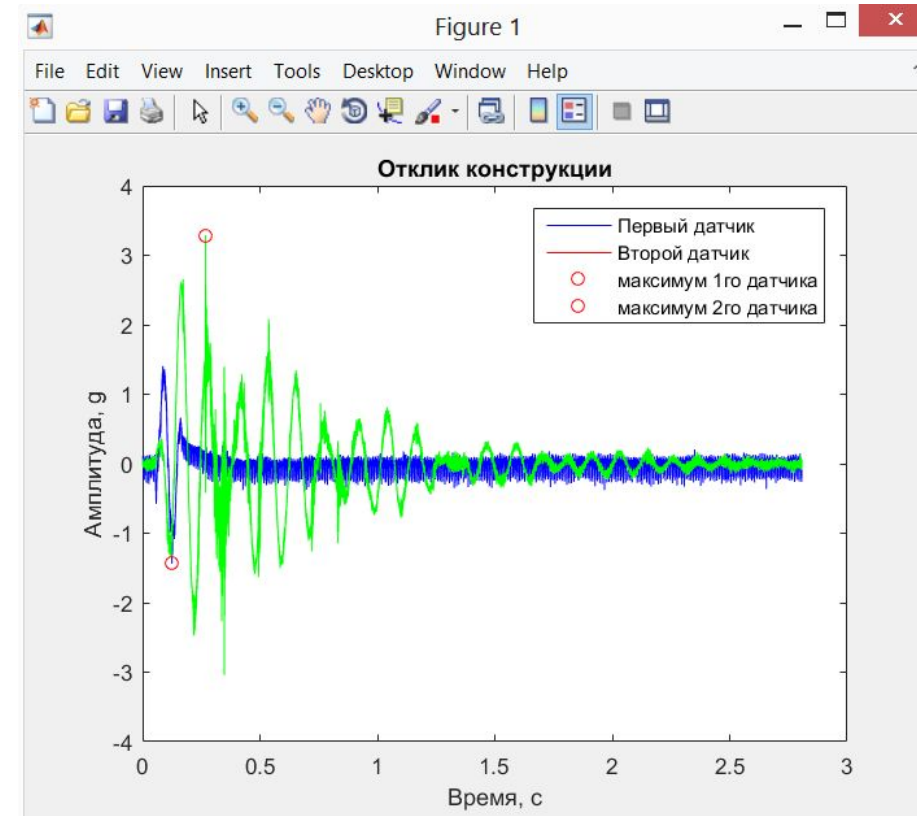
# Домашнее задание:

4. В семинаре присутствует ошибка при определении максимальной перегрузки, показанной датчиком. В чем она заключается и как её устранить?

Было

```
Editor - D:\P.K. Хамидуллин\МАТЛАБ для студентов\Seminar_1.m*
Reducirovanie_VR_failov.m x Seminar_1.m* x Untitled.m x +
5 - Dat1 = DATA(:, 2); %т.е. в Dat1 запишем все строки (":") первого столбца
6 - Dat2 = DATA(:, 3); %т.е. в Dat2 запишем все строки (":") первого столбца
7   % Размер матрицы
8 - Razmer = length(Time);
9   % Длительность записи
10 - Dlit = Time(end)-Time(1);
11  % Поиск максимального значения
12 - [D1, ind1]=max(Dat1); % ищем максимум показаний первого датчика
13 - [D2, ind2]=max(Dat2); % ищем максимум показаний второго датчика
14
```

Необходимо поставить модуль при определении амплитуды перегрузки



Должно

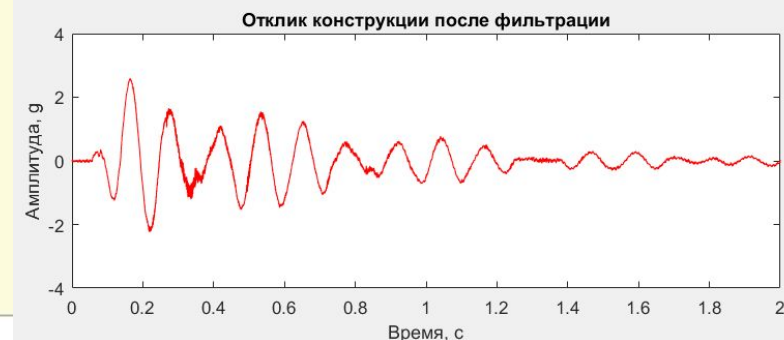
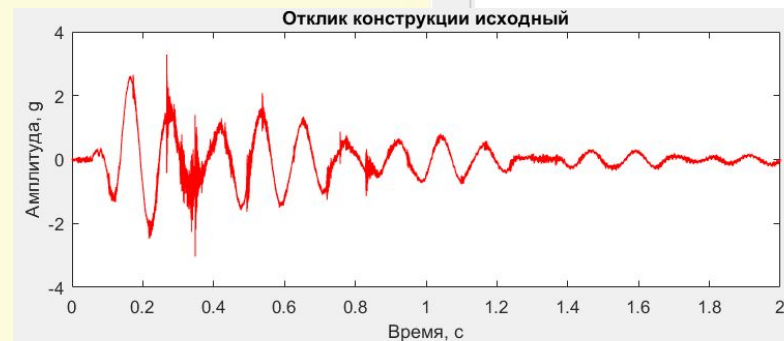
```
Быть: Seminar_1.m*
Seminar_1.m* x Untitled.m x +
Variables - Dat2s
4 - Time = DATA(:, 1); %т.е. в Time запишем все строки (":") первого столбца
5 - Dat1 = DATA(:, 2); %т.е. в Dat1 запишем все строки (":") второго столбца
6 - Dat2 = DATA(:, 3); %т.е. в Dat2 запишем все строки (":") третьего столбца
7   % Размер матрицы
8 - Razmer = length(Time);
9   % Длительность записи
10 - Dlit = Time(end)-Time(1);
11  % Поиск максимального значения
12 - [D1, ind1]=max(abs(Dat1)); % ищем максимум показаний первого датчика
13 - [D2, ind2]=max(abs(Dat2)); % ищем максимум показаний второго датчика
```

# Домашнее задание прошлого занятия:

## 5. Отфильтруйте сигналы с датчиков при помощи осреднения:

Решение: необходимый код приведен ниже.

```
Editor - Seminar_1.m* Variables - Dat2s
Seminar_1.m* x Untitled.m x 3M22.m x Untitled* x Untitled2* x Untitled3* x Untitled4* x +
48 %% Домашняя работа
49 % фильтрация при помощи осреднения
50 n=5;
51 % Осредненные значения запишем в матрицы Dat1s и Dat2s.
52 % До начала осреднения эти матрицы будут равны исходным
53 Dat1s=Dat1;
54 Dat2s=Dat2;
55 % Заменяем элементы матрицы осредненными значениями при помощи цикла for
56 for i=(n+1):(Razmer-n-1); % выбираем каждый n-ый элемент от первого до последнего элемента
57 Dat1s(i)=sum(Dat1((i-n):(i+n)))/(2*n+1); % для первого датчика
58 Dat2s(i)=sum(Dat2((i-n):(i+n)))/(2*n+1); % для второго датчика
59 end
60 % Строим графики для сравнения в новом окне
61 figure(3)
62 subplot(2,1,1)
63 plot(T, Dat2, 'r');
64 title('Отклик конструкции исходный'); % заголовок
65 ylabel('Амплитуда, g'); % подпись оси абсцисс
66 xlabel('Время, с'); % подпись оси ординат
67 axis([0,2,-4,4]) % границы графика от 0с до 2с и от -4 до 4g
68 subplot(2,1,2)
69 plot(T, Dat2s, 'r')
70 title('Отклик конструкции после фильтрации'); % заголовок
71 ylabel('Амплитуда, g'); % подпись оси абсцисс
72 xlabel('Время, с'); % подпись оси ординат
73 axis([0,2,-4,4]) % границы графика от 0с до 2с и от -4 до 4g
```



# Домашнее задание прошлого занятия:

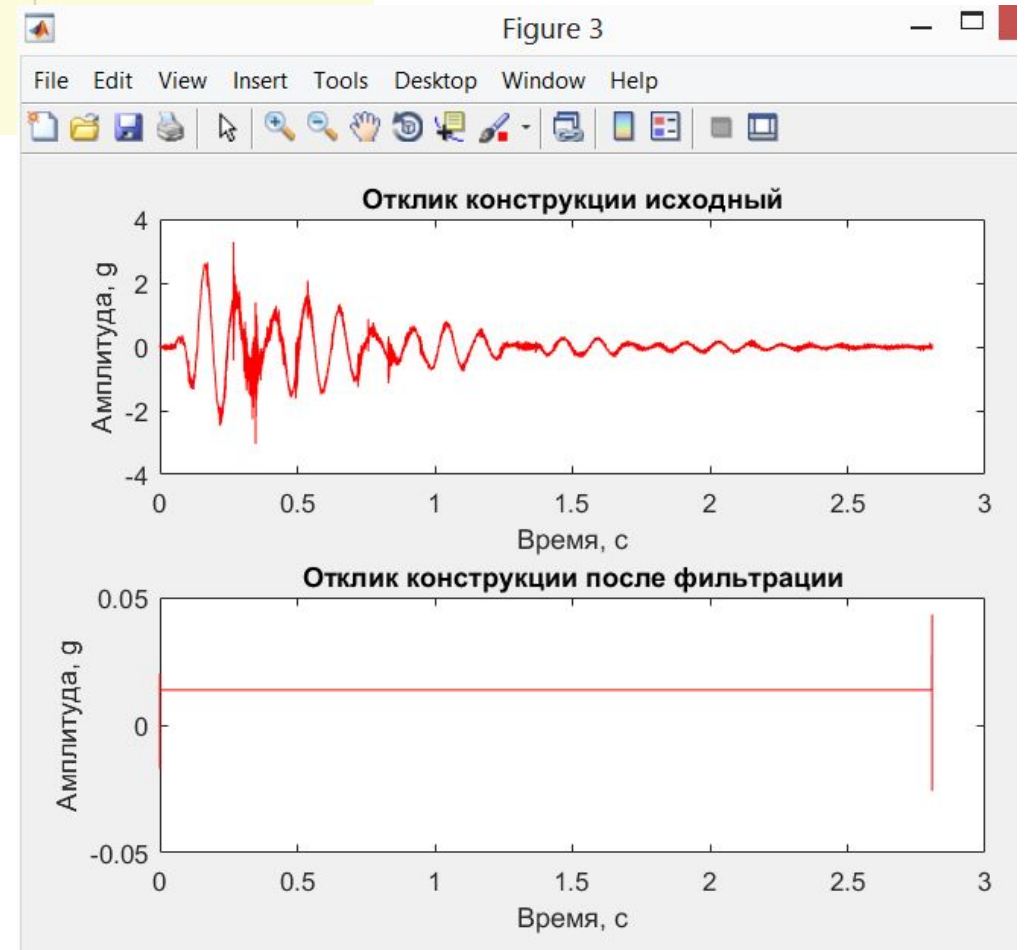
5. Отфильтруйте сигналы с датчиков при помощи осреднения:  
Разберем для чего использовался цикл for. Уберем его и построим график заново.

```
% Заменяем элементы матрицы осредненными значениями при помощи цикла for
%for
i=(n+1):(Razmer-n-1); % выбираем каждый n-ный элемент от первого до последнего элемента
Dat1s(i)=sum(Dat1((i-n):(i+n)))/(2*n+1); % для первого датчика
Dat2s(i)=sum(Dat2((i-n):(i+n)))/(2*n+1); % для второго датчика
%end
% Строим графики для сравнения в новом окне
figure(3)
```

Это фиаско! Грусть, тоска и отчаяние. Но в чем же причина?

Без цикла for было посчитано лишь 1 значение функций Dat1s и Dat2s для  $i=n+1$  и присвоено для всех значений функций от  $(n+1)$  до  $(Razmer-n-1)$ .

А зачем мы вообще записали  $i$  не от 1 до end?

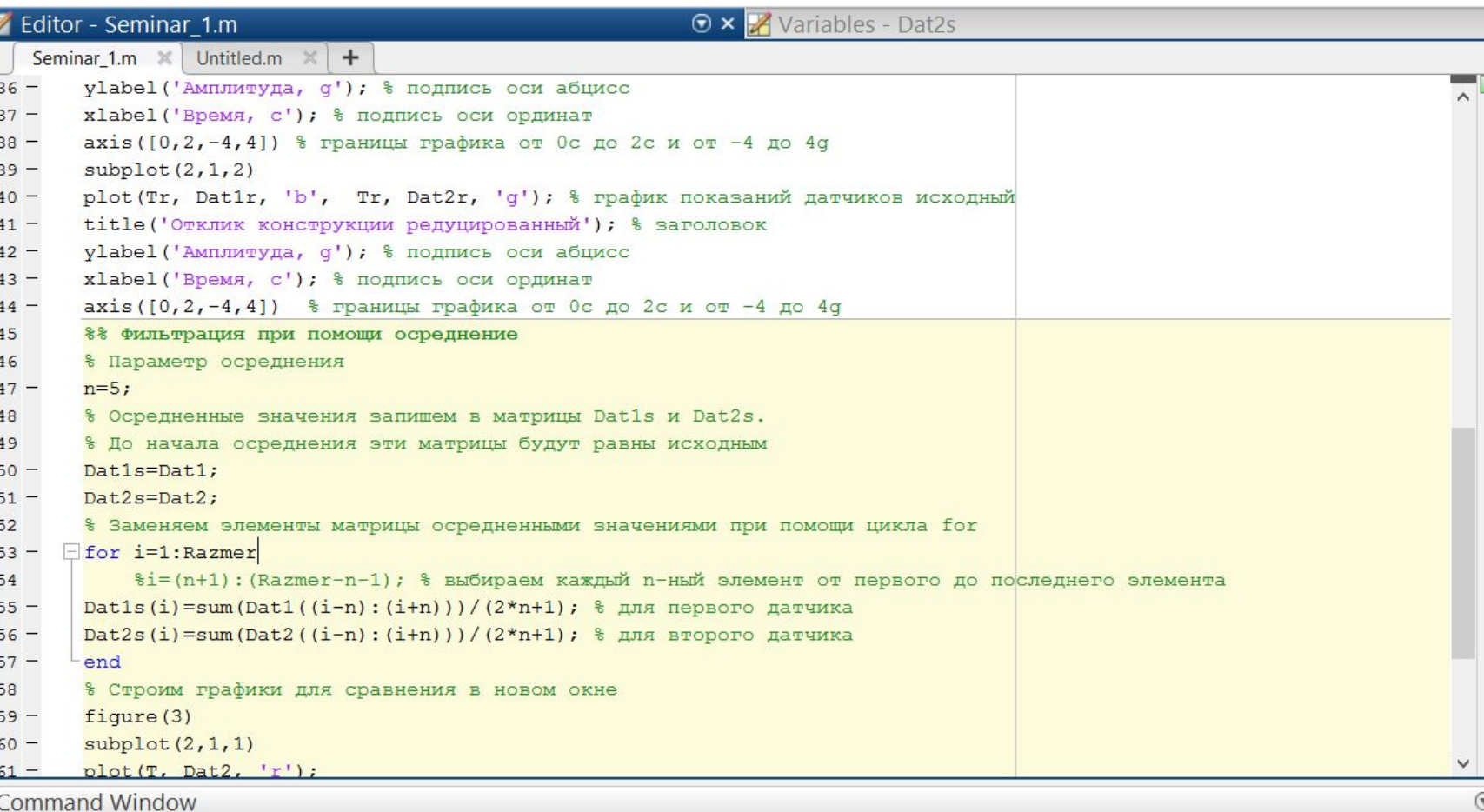




# Домашнее задание прошлого занятия:

## 5. Отфильтруйте сигналы с датчиков при помощи осреднения:

Вернем цикл for и поменяем пределы определения переменной  $i$



```
36 - ylabel('Амплитуда, g'); % подпись оси абсцисс
37 - xlabel('Время, с'); % подпись оси ординат
38 - axis([0,2,-4,4]) % границы графика от 0с до 2с и от -4 до 4g
39 - subplot(2,1,2)
40 - plot(Tr, Dat1r, 'b', Tr, Dat2r, 'g'); % график показаний датчиков исходный
41 - title('Отклик конструкции редуцированный'); % заголовок
42 - ylabel('Амплитуда, g'); % подпись оси абсцисс
43 - xlabel('Время, с'); % подпись оси ординат
44 - axis([0,2,-4,4]) % границы графика от 0с до 2с и от -4 до 4g
45 - %% Фильтрация при помощи осреднение
46 - % Параметр осреднения
47 - n=5;
48 - % Осредненные значения запишем в матрицы Dat1s и Dat2s.
49 - % До начала осреднения эти матрицы будут равны исходным
50 - Dat1s=Dat1;
51 - Dat2s=Dat2;
52 - % Заменяем элементы матрицы осредненными значениями при помощи цикла for
53 - for i=1:Razmer
54 -     %i=(n+1):(Razmer-n-1); % выбираем каждый n-ный элемент от первого до последнего элемента
55 -     Dat1s(i)=sum(Dat1((i-n):(i+n)))/(2*n+1); % для первого датчика
56 -     Dat2s(i)=sum(Dat2((i-n):(i+n)))/(2*n+1); % для второго датчика
57 - end
58 - % Строим графики для сравнения в новом окне
59 - figure(3)
60 - subplot(2,1,1)
61 - plot(T, Dat2, 'r');
```

Command Window  
Subscript indices must either be real positive integers or logicals.

ОШИБКА  
!!!

Да потому что нельзя, например, при  $i=1$  находить  $Dat1(i-n)$ , т.е.  $Dat1(-4)$ .  
Не бывает матриц с отрицательными номерами

# Перемножение элементов матриц:

Есть две матрицы A и B.

Для матричного умножения нужно ввести

$A*B$

Для скалярного умножения нужно ввести

$A.*B$

```
A =
```

```
1 2 3
4 5 6
7 8 9
```

```
>> B=[1 2 3; 4 5 6; 7 8 9]
```

```
B =
```

```
1 2 3
4 5 6
7 8 9
```

```
>> A*B
```

```
ans =
```

```
30 36 42
66 81 96
102 126 150
```

```
>> A.*B
```

```
ans =
```

```
1 4 9
16 25 36
49 64 81
```

# Оператор цикла `for`

```
for index=values
```

```
    команды
```

```
end
```

`index` - счетчик цикла. `values` могут задаваться в следующем виде:

- ▶  $a : b$ , где  $a$  - начальное значение,  $b$  - конечное значение. Шаг по умолчанию будет равен 1.
- ▶  $a : s : b$ , где  $a$  - начальное значение,  $b$  - конечное значение,  $s$  - шаг.
- ▶ `valArray` - массив значений.

# Условный оператор

```
if условие
    команды
elseif условие
    команды
else
    команды
end
```

Условие может содержать

- ▶ операторы отношения (например,  $<$  или  $==$ )
- ▶ логические операторы (например,  $\&\&$  или  $\sim$ ).

# Оператор цикла `while`

**while** условие

    команды

**end**

Команды из тела цикла выполняются, пока выполнено заданное условие.

Условие может содержать

- ▶ операторы отношения (например, `<` или `==`)
- ▶ логические операторы (например, `&&` или `~`).

# Операторы `break` и `continue`

- ▶ Оператор `break` используется для прерывания выполнения циклов `for` и `while`. Если `break` выполнен в одном из вложенных циклов, то прекратится выполнение только этого цикла (остальные продолжат работу).
- ▶ Оператор `continue` используется для прерывания текущей итерации циклов `for` и `while`. При выполнении оператора `continue` в теле цикла текущая итерация этого цикла завершается, и начинается следующая, если это возможно.

# ПРИМЕР! ПРИМЕР! ПРИМЕР! ПРИМЕР!

Взять текстовый файл с тремя переменными (время, показания двух датчиков) и

- Оценить частоту колебаний основной гармонике по 5 периодам колебаний второго датчика;
- Оценить декремент колебаний по 5 периодам затухания амплитуды колебаний по второму датчику.

# Оценить частоту колебаний основной гармоники по 5 периодам колебаний второго датчика

Определим первые пять периодов колебаний по пересечению нуля графика амплитуды.

Поиск значений датчика равные нулю:

```
Command Window
>> find(Dat2s==0)

ans =

Empty matrix: 0-by-1
```

У нас ничего не получилось. Таких значений не оказалось, хотя на графике было видно, что пересечений нуля множество.

```
74 %% Семинар 2
75 - for i=1:(Razmer-1); % выбираем каждый n-ный элемент от первого до предпоследнего элемента
76 -     Q(i)=Dat2s(i)*Dat2s(i+1);
77 - end
78 - ind0=find(Q<=0);
```

Получилось! Мы обнаружили больше сотни пересечений нуля. Условием пересечения нуля являлось неположительность результата перемножения двух последовательных показаний датчика.

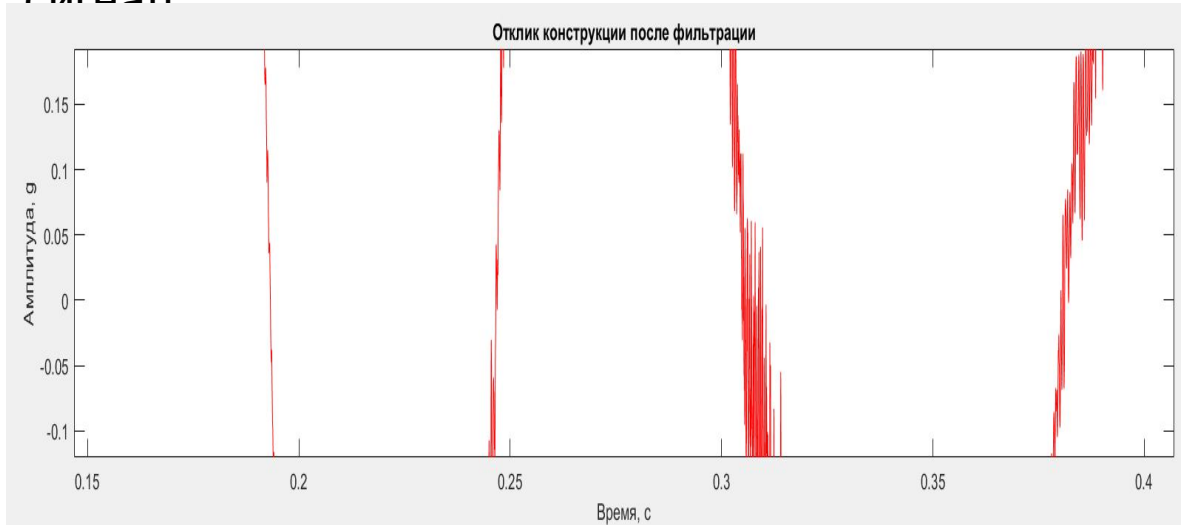
Но почему же их так много (более 1000)? Увеличим график амплитуды от времени и всё поймем:



Но почему же их так много? Увеличим график амплитуды от времени и всё поймем:

В домашнем задании мы недостаточно хорошо отфильтровали

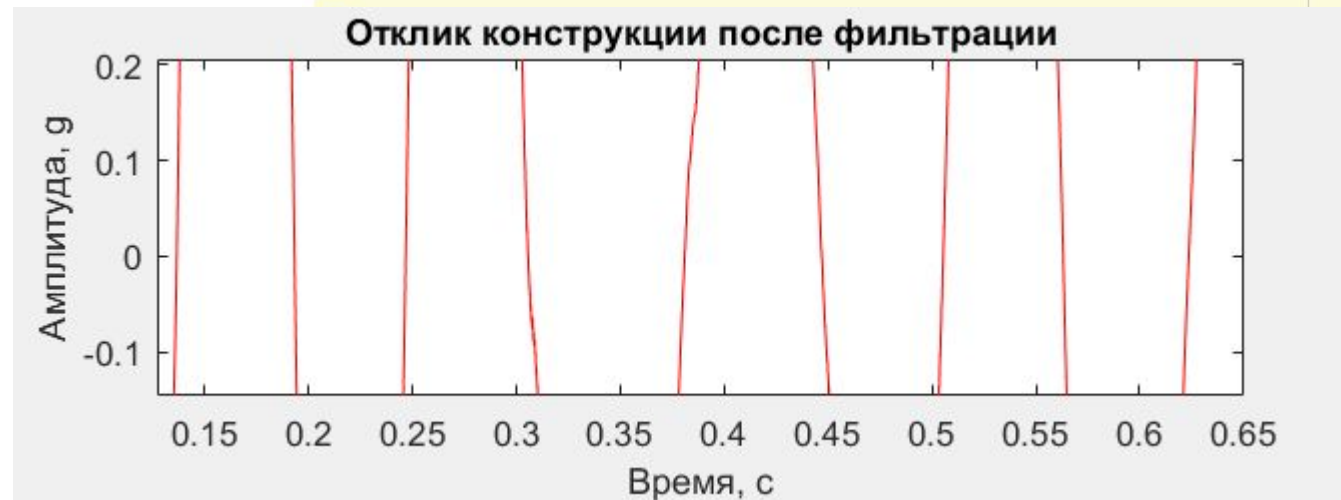
сигнал



Поставим  $n=100$  и перестроим

график

```
%% Домашняя работа
% Фильтрация при помощи осреднения
n=100;
% Осредненные значения запишем в матрицы Dat1s и Dat2s.
% До начала осреднения эти матрицы будут равны исходным
Dat1s=Dat1;
Dat2s=Dat2;
% Заменяем элементы матрицы осредненными значениями при помощи цикла for
for i=(n+1):(Razmer-n-1); % выбираем каждый n-ный элемент от первого до последнего элемента
    Dat1s(i)=sum(Dat1((i-n):(i+n)))/(2*n+1); % для первого датчика
    Dat2s(i)=sum(Dat2((i-n):(i+n)))/(2*n+1); % для второго датчика
end
```



Но почему же их так много? Увеличим график амплитуды от времени и всё поймем:

Давайте для наглядности отметим на графике точки близкие к пересечению

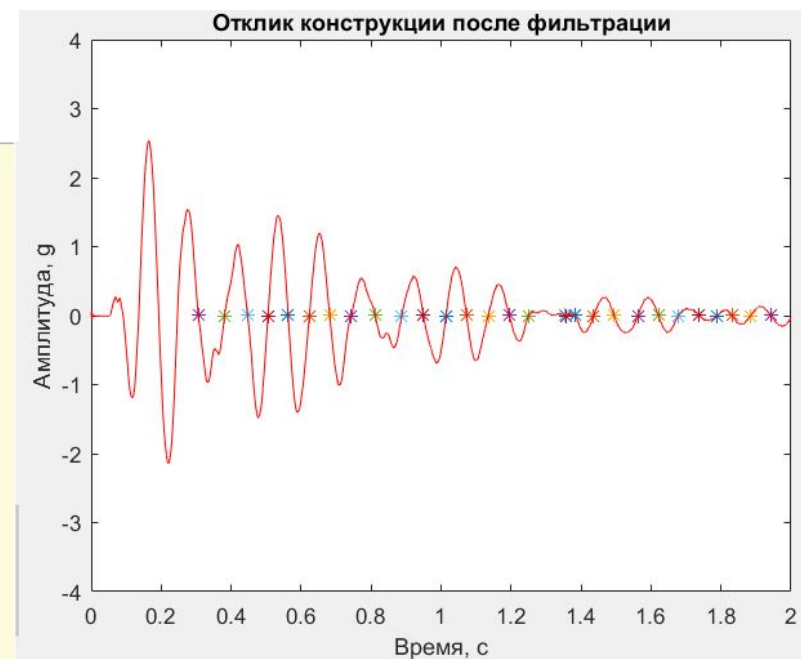
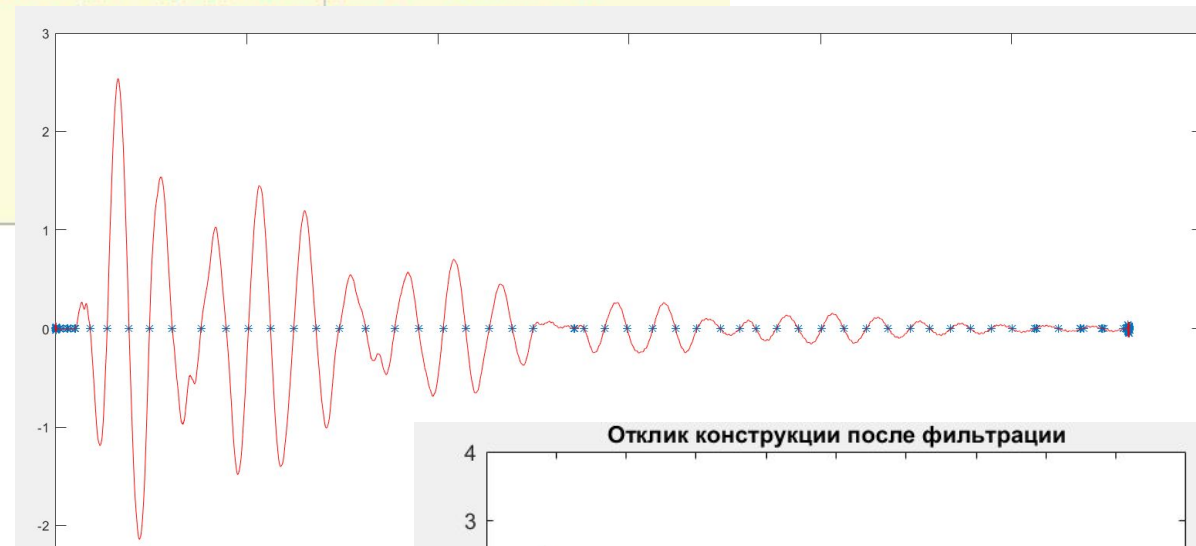
нуля

```
75 %% Семинар 2
76 for i=1:(Razmer-1); % выбираем каждый n-ный элемент от первого до предпоследнего элемента
77 Q(i)=Dat2s(i)*Dat2s(i+1); %
78 end
79 ind0=find(Q<=0); % индексы моментов пересечения нуля.
80 figure(4)
81 plot(T(ind0), Dat2s(ind0),'*', T, Dat2s, 'r');
```

Ой! Их слишком много. Давайте выберем только те, что идут

После максимального значения амплитуды. Напомню, что индекс максимума второго датчика ind2 мы находили на прошлом семинаре. Используем это...

```
75 %% Семинар 2
76 for i=1:(Razmer-1); % выбираем каждый n-ный элемент от первого до предпоследнего элемента
77 Q(i)=Dat2s(i)*Dat2s(i+1); %
78 end
79 ind0=find(Q<=0); % индексы моментов пересечения нуля.
80 figure(4) % новое окно
81 for i=1:length(ind0) % для всех значений ind0
82     if ind0(i)>ind2 % если оно правее максимума второго датчика...
83         plot(T(ind0(i)), Dat2s(ind0(i)), '*'); hold on % ...то печатаем. Не забывать hold on!!!
84     end
85 end
86 plot(T, Dat2s, 'r') % добавим график показаний датчика
87 title('Отклик конструкции после фильтрации'); % заголовок
88 ylabel('Амплитуда, g'); % подпись оси абсцисс
89 xlabel('Время, с'); % подпись оси ординат
90 axis([0,2,-4,4]) % границы графика от 0с до 2с и от -4 до 4g
```



Определим основную частоту колебаний.

```
91 - Chislo_Period=5; % количество рассматриваемых периодов
92 - I=find(ind0>ind2) %выбираем только те индексы, что правее максимума второго датчика
93 - dlit=T(ind0(I(2*Chislo_Period+1)))-T(ind0(I(1))); % длительность колебаний за Chislo_Period
94 - Period=dlit/Chislo_Period;
95 - Chastota=1/Period;
```

Workspace	
Name ^	Value
ans	0.9498
Chastota	7.7690
Chislo_Period	5
D1	1.4371
D2	3.2878

Разумеется, это значение не точно, так как на результаты влияли другие гармоники, но для первичной оценки сгодится.

Ну а теперь найдем декремент

колебаний

```
96 %%
97 figure(5)
98 for i=1:2:(length(I)-1) %выбираем все нечетные значения
99     %(т.к. четные являются второй полуволной)
100     [MAX(i),b(i)]=max(Dat2s(ind0(I(i)):ind0(I(i+2)))); %ищем максимум между нулями по Амплитуде
101     % причем MAX - значение, b - его порядковый номер в "локальной" матрице
102     % значений
103     w(i)=ind0(I(i))+b(i); % порядковый номер точек максимума в "глобальной" матрице значений
104     plot(T(w(i)), MAX(i),'*'); hold on % строим график максимумов
105 end
106 plot(T, Dat2s, 'r') % исходный график амплитуды
107 kolPer=input('введите количество рассматриваемых периодов '); % количество рассматриваемых периодов
108 dekrement=log(MAX(1)/MAX(2*kolPer+1))/kolPer
```

# Домашнее задание:

1. На слайде 18 видно, что поиск пиков показаний датчика велся не с первого большого пика. Можете ли это исправить?
2. Постройте график изменения декремента от количества рассматриваемых периодов  $k$ .