

Calculation of Response using FFT in MATLAB

РАСЧЕТ РЕАКЦИИ С ПРИМЕНЕНИЕМ ПРЕОБРАЗОВАНИЯ ФУРЬЕ В СРЕДЕ МАТЛАБ

Цель работы: Знакомство с правилами перевода сигналов в частотную область и обратно в среде МАТЛАБ.

Задача работы: Вычислить реакцию сигнала во временной области используя частотные характеристики передаточных функций.

Приборы и принадлежности: Персональный компьютер, интегрированная среда МАТЛАБ.

ВВЕДЕНИЕ

Основными формами представления конечномерных линейных непрерывных стационарных детерминированных операторов преобразования входных переменных в переменные выхода являются:

- дифференциальные уравнения,
- передаточные функции,
- временные и частотные характеристики [1].

Временная область (time domain) является важной для анализа и синтеза систем управления. Сигналы в ней представлены функциями изменения амплитуды с течением времени. В этой области определяются прямые показатели качества, такие как

- установившееся значение процесса,
- время регулирования,
- перерегулирование,

только во временной области можно измерить длительность фронта и спада сигнала.

В частотной области (frequency domain) сигнал представлен совокупностью синусоидальных функций различной частоты, амплитуды и фазы. Преимуществами анализа и синтеза систем в частотной области являются относительная простота

- построения передаточных функций;
- оценки устойчивости системы;
- анализа характеристик шума;
- корректировки параметров системы,

а также то, что частотный анализ может выполняться и для нелинейных систем.

Прямые и обратные преобразования Фурье – наиболее распространенный метод для перевода сигнала из временной области (time domain) в частотную область (frequency domain) и обратно. Основная идея прямого преобразования заключается в том, что любую функцию можно представить в виде бесконечной суммы синусоид. Для точного перевода сигнала в

частотную область необходим бесконечный период, а для перевода спектра сигнала во временную область необходим бесконечный частотный диапазон. На практике такие требования невыполнимы, однако, применение ограниченного периода и диапазона с наибольшей энергией сигнала позволяет получить вполне приемлемые результаты преобразования.

В этой работе в среде МАТЛАБ рассчитывается реакция периодического импульсного сигнала проходящего через нормированный фильтр и канал, представленный частотной характеристикой измеренной в ограниченном диапазоне с периодом, не кратным частоте входного сигнала. При согласовании параметров выполняется масштабирование частотных характеристик с использованием функций интерполяции и экстраполяции.

ОБЩИЕ СВЕДЕНИЯ

Разрешение (шаг квантования по амплитуде) измерения аналогового непрерывного физического параметра преобразуемого в дискретную величину цифровым вычислительным устройством обратнопропорционально разрядности Аналого Цифрового Преобразователя (АЦП). Период квантования по времени дискретного сигнала ограничен скоростью преобразования и временем считывания. Шаг квантования аналитической или моделируемой функции зависит от разрядности компьютера и типа переменной. Шаг вычисления амплитуды, как правило, значительно меньше минимального приращения по амплитуде АЦП. Квантование сигнала содержит ошибку по амплитуде как показано на Рисунок 1.

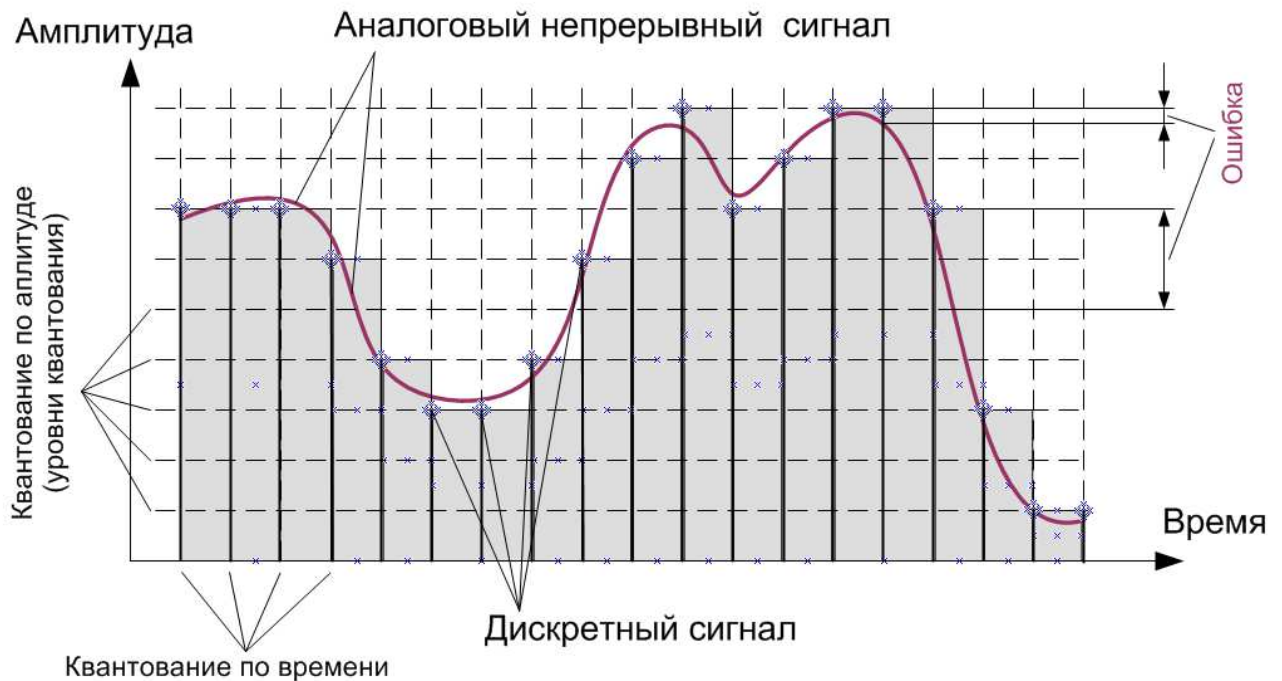


Рисунок 1. Аналоговый непрерывный сигнал и его дискретный образ. Дискретный сигнал получается в результате преобразования аналогового сигнала или вычисления аналогового сигнала цифровым устройством. При использовании АЦП амплитуда дискретного сигнала равна коду уровня, ближайшего к аналоговому сигналу в момент измерения. До очередного преобразования сигнала используется его значение, полученное на предыдущем шаге.

По теореме Котельникова, аналоговый периодический сигнал, имеющий ограниченный по частоте спектр $0 \dots f_{\max}$, может быть восстановлен без искажений по своим дискретным отсчётам, взятым с частотой дискретизации $f_{\text{дискр}} \geq 2 \cdot f_{\max}$. Другими словами, при частоте дискретизации 1000 Гц можно восстановить без искажений аналоговый сигнал, спектр которого не превышает 500 Гц.

Следует отметить, что дискретизация функции по времени приводит к периодизации ее спектра, а дискретизация спектра по частоте приводит к периодизации функции.

Основные функции МАТЛАБ используемые для вычисления реакции с переводом сигнала в частотную область и обратно приведены в Таблица 1.

Таблица 1. Основные функции МАТЛАБ используемые в Приложении для расчета реакции.

| Функция | Описание, примеры |
|---|---|
| Преобразование Фурье | |
| fft | Переводит сигнал в частотную область, строит спектр сигнала по алгоритму Быстрое Преобразование Фурье |
| fftshift | Сдвигает элементы массива на половину длины массива (см. Рисунок 5 и Рисунок 6). fftshift([1 2 3 4 5 6 7 8 9 10]) “=” 6 7 8 9 10 1 2 3 4 5 fftshift([1 2 3 4 5 6 7 8 9 10 11]) “=” 7 8 9 10 11 1 2 3 4 5 6 |
| ifft | Обратное преобразование Фурье. Строит сигнал во временной области по его спектральной (частотной) характеристике. |
| ifftshift | Сдвигает элементы массива на половину длины массива (см. Рисунок 5 и Рисунок 6). ifftshift([1 2 3 4 5 6 7 8 9 10]) “=” 6 7 8 9 10 1 2 3 4 5 ifftshift([1 2 3 4 5 6 7 8 9 10 11]) “=” 6 7 8 9 10 11 1 2 3 4 5 |
| Аппроксимация, интерполяция, экстраполяция | |
| polyfit | Нахождение коэффициентов степенного полинома аппроксимирующей функции |
| polyval | Построение степенного полинома по его коэффициентам |
| interp1 | Интерполяция, экстраполяция |

| Комплексные числа | |
|--------------------------|--|
| abs | Абсолютное значение, модуль комплексного числа abs(-5) “=” 5; abs(3+4i) “=” 5 |
| real | Действительная часть комплексного числа real(3+4i) “=” 3 |
| imag | Мнимая часть комплексного числа imag(3+4i) “=” 4 |
| angle phase | Фазовый угол комплексного числа в радианах angle(0+4i) “=” 1.5708; angle(2+2i) “=” 0.7854; phase(0+4i) “=” 1.5708; phase(2+2i) “=” 0.7854; |
| abs()*exp(+j.*angle()); | Вычисление комплексного числа по его модулю и фазе |
| real() + j*imag() | Построение комплексного числа по его действительной и мнимой частям |

Функция `fft()` переводит сигнал в частотную область по алгоритму Быстрого Преобразования Фурье (Fast Fourier Transform) - строит спектр периодического сигнала.

По спектру сигнала можно найти основную частоту, например, случайной последовательности, как это показано в Help МАТЛАБ и на .

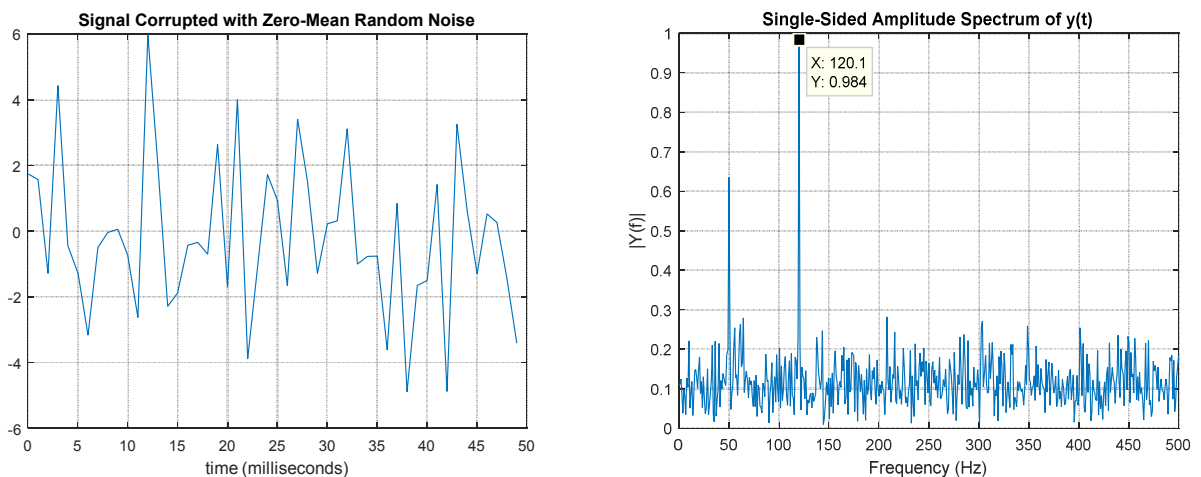


Рисунок 2. Шум во временной области (слева) и его спектр (справа). Пример взят из Help МАТЛАБ. По виду сигнала (слева) нелегко определить его основную частоту, однако спектральная характеристика (справа) полученная Быстрым Преобразованием Фурье выделяет основную частоту равную 120.1 Гц.

ifft(x) выполняет обратное преобразование Фурье. Переводит сигнал из частотного пространства во временное пространство - восстанавливает сигнал по его спектральной характеристике.

Частотный диапазон Фурье спектров включает отрицательные, нулевую и положительные частоты. Амплитудный спектр Фурье-образа действительной функции всегда является четной функцией, а фазовый спектр - нечетной. Неучет части Фурье-образа, соответствующей отрицательным частотам, может привести к большим погрешностям.

ПРИМЕРЫ ПОЛУЧЕНИЯ ПРОВЕРЕННЫХ РЕЗУЛЬТАТОВ И ВАРИАНТЫ ДЛЯ САМОКОНТРОЛЯ

В этой работе рассматривается расчет реакции канала передачи на импульсное воздействие, основные вычисления которого выполняются в частотной области. Для этого

- исходный импульсный сигнал из временной области переводится в частотную область с использованием преобразования Фурье,
- перемножением спектра сигнала, частотных характеристик фильтра и сигнальной линии находится выходной сигнал тракта в частотной области,
- выходной сигнал (реакция на импульсное воздействие) из частотной области переводится во временную область обратным преобразованием Фурье.

ЗАДАНИЕ

Необходимо определить изменение формы двоичного цифрового сигнала проходящего через фильтр и сигнальную линию. Сигнал задан амплитудой и скоростью передачи. Фильтр второго порядка, нормированный относительно частоты передачи данных, задан постоянными времени. Передаточная функция сигнальной линии представлена измеренной частотной характеристикой в комплексной форме.

Среда, используемая для вычисления и отображения данных – MATLAB R2015a.

В качестве примера исходных данных взяты следующие отношения, опубликованные на сайте <http://www.StatEye.org/> для версии метода StatEye 3.0 GUI [1, 3, 3].

Скорость передачи данных $\text{bps} = 10,3125$ Гбит/с. Постоянные времени нормированного фильтра второго порядка совпадают, их обратная величина составляет $\frac{3}{4}$ частоты передачи данных. Сигнальная линия представлена частотной характеристикой. Измерение характеристики выполнено на частотах $\text{channel.f} = 0,006495:0,0012475:20$ ГГц. Заданное число точек дискретизации преобразования Фурье: $\text{points} = 2^{13}$.

На Рисунок 3 показаны передача данных, последовательность и результаты обработки данных, которые рассмотрены в этой работе. Переход из временной области в частотную область и обратно выполняется при помощи алгоритма Быстрого Преобразования Фурье (БПФ, FFT).

Общее задание необходимо выполнять в последовательности, как показано ниже. Для каждого задания ниже необходимо написать скрипт для среды МАТЛАБ и сравнить его со скриптом Приложения.

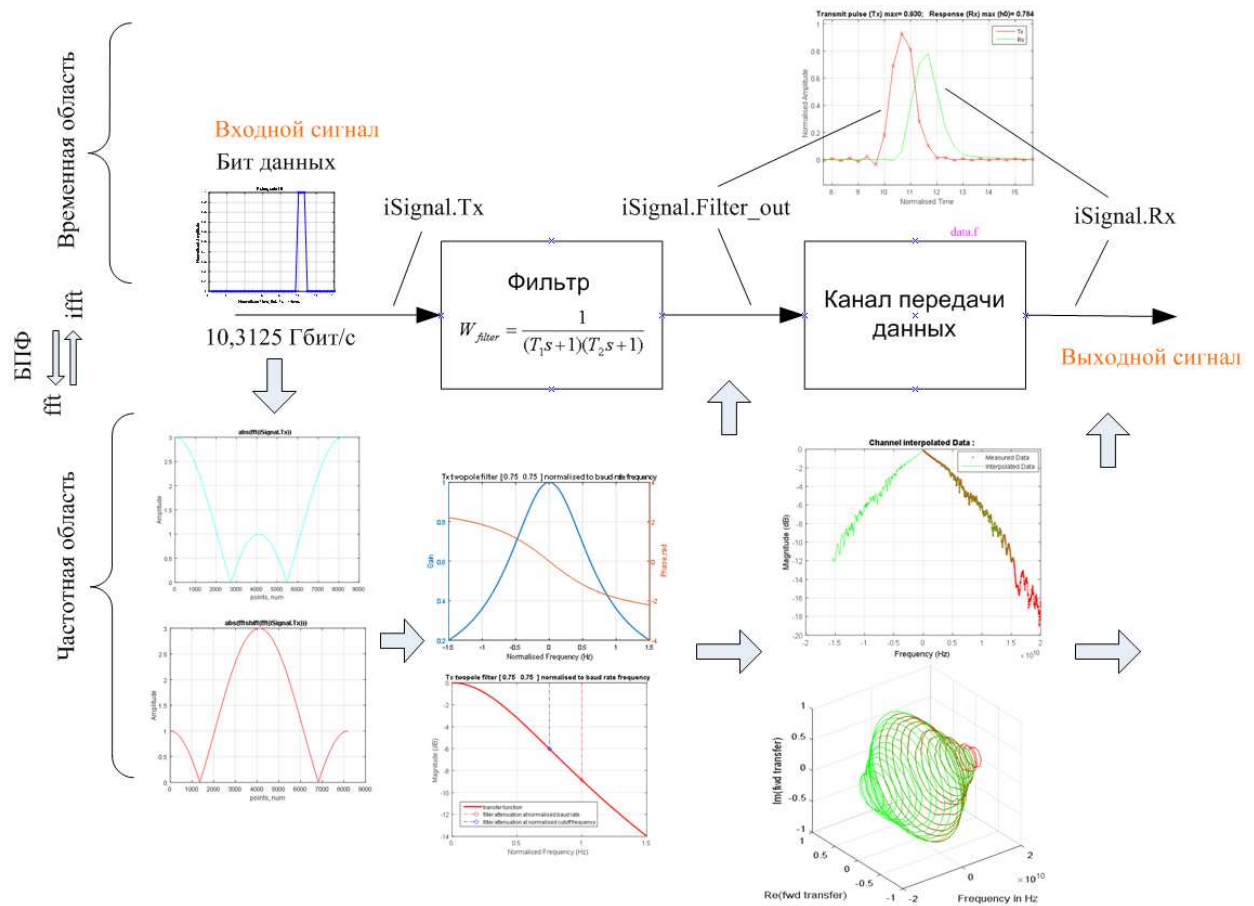


Рисунок 3. Канал передачи данных. Входной сигнал `iSignal.Tx`, выходной сигнал фильтра `iSignal.Filter_out`, выход сигнальной линии `iSignal.Rx`. Представленные на диаграмме характеристики рассмотрены ниже.

Последовательность выполнения задания:

1. Вычисление максимальной частоты передачи данных.

Скорость передачи данных в два раза выше частоты, на которой происходит передача данных. Максимальная частота измеренной сигнальной линии $\max(\text{channel.f}) = 20$ ГГц. На этой частоте можно передавать данные со скоростью 40 Гбит/с (как $2 \cdot \max(\text{channel.f})$).

Максимальная скорость передачи данных, которая не превышает максимальную скорость передачи по сигнальной линии 40 Гбит/с и кратная скорости передачи $\text{bps} = 10,3125$ Гбит/с, равна $f_{\max} = 30,9375$ Гбит/с, кратность $N = 3$ ($N = f_{\max}/\text{bps}$). Далее, f_{\max} используется как предельная частота для расчета реакции сигнала с применением преобразования Фурье.

2. Построение импульсного сигнала и перевод сигнала в частотную область.

Дискретность по времени для построения входного сигнала (бита данных) во временной области $T_s = 1/f_{max}$; $T_s = 3,232e-11$ с. Нормированная по отношению к длительности сигнала, временная шкала `time` состоит из 2^{13} точек (points), шкала включает следующий массив точек `time = bps/Ts .* (1:points)`. Дискретный единичный сигнал при скорости передачи $bps = 10,3125$ Гбит/с и квантовании с периодом $T_s = 1/f_{max}$ состоит из трех точек в диапазоне от 10 до 11 единиц нормализованного времени. В соответствии со вторым свойством преобразования Фурье амплитудный спектр не зависит от сдвига сигнала по времени. Т.о. сигнал единичной амплитуды можно построить и в любом другом месте временной шкалы, но лучше отступить с краев, чтобы полностью видеть предысторию и переходный процесс выходного сигнала (реакции). Импульсный сигнал (бит данных), построенный с использованием следующих команд МАТЛАБ, показан на Рисунок 4.

```
iSignal.Tx(1:size(time,2)) = 0;  
t0 = max(find(time<=10));  
t1 = max(find(time<11));  
iSignal.Tx(t0:t1) = 1.0;
```

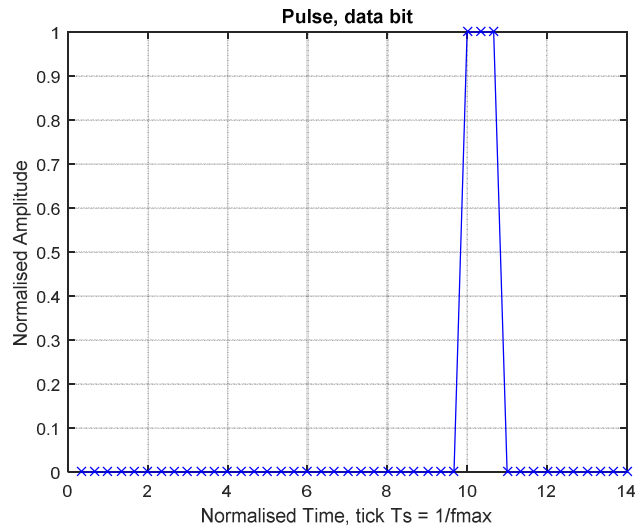


Рисунок 4. Входной импульсный сигнал `iSignal.Tx`, бит данных.

Перевод сигнала `iSignal.Tx` в частотную область выполняют следующие БПФ функции.

```
iSignal.shiftedPSD = fft(iSignal.Tx);  
iSignal.PSD = fftshift(iSignal.shiftedPSD);
```

Функция преобразования Фурье `fft` строит симметричный спектр сигнала в областях положительных и отрицательных частот, максимальная частота которого находится в центре спектра (см. Рисунок 5). Функция `fftshift` восстанавливает спектр, сдвигая в центр нулевую частоту сигнала как показано на Рисунок 6.

Разрешение частоты спектра равно $f_s = f_{max}/points$; Частоты спектра лежат в диапазоне от $-f_{max}/2$ до $f_{max}/2 - f_s$ и равны $f = -f_{max}/2 : f_s : f_{max}/2 - f_s$;

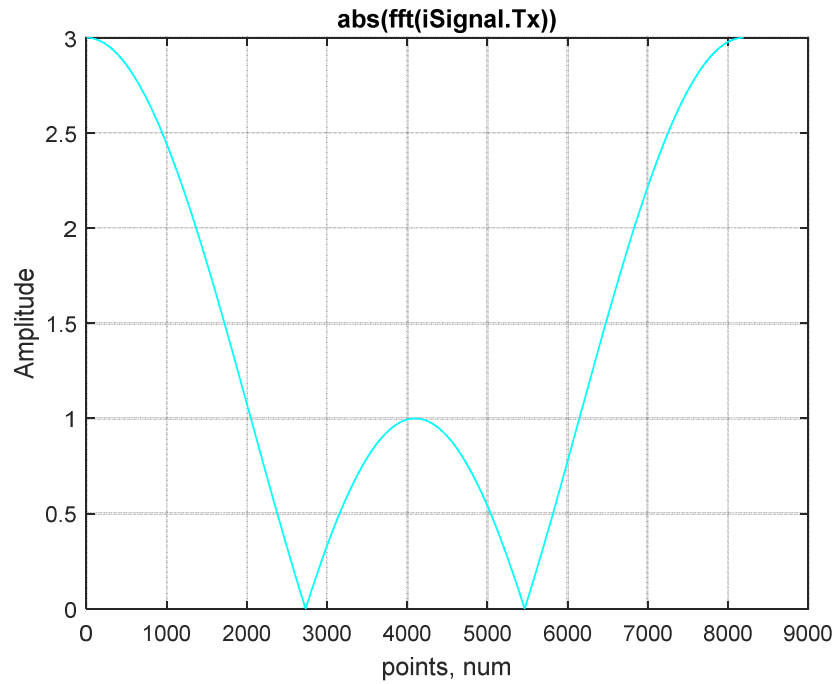


Рисунок 5. Амплитудная характеристика сдвинутого спектра сигнала `iSignal.Tx` полученного с использованием БПФ.

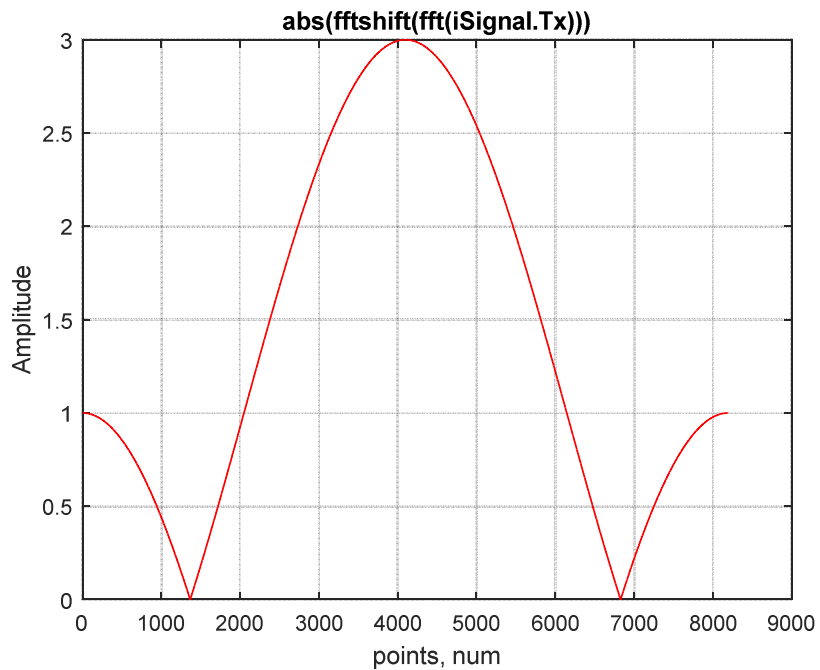


Рисунок 6. Амплитудная характеристика восстановленного спектра сигнала `iSignal.Tx` показанного на **Рисунок 5**. Представлено 2^{13} отсчетов. Средний отсчет в точке 4097 соответствует нулевой частоте. В левой части (от 1 до 4096 точки) располагаются отрицательные частоты, в правой части (от 4098 до 8192 точки) – область положительных частот.

3. Построение передаточной функции нормализованного фильтра нижних частот.

В этом примере передаточная функция фильтра второго порядка имеет вид

$$W_{filter} = \frac{1}{(T_1s + 1)(T_2s + 1)},$$

где T_1 и T_2 – постоянные времени фильтра. Значения частот $1/T_1$ равны и $1/T_2$ заданы относительно частоты, на которой передаются данные: $1/T_1 = 1/T_2 = 0,75 \cdot \text{bps}$ ($\text{bps} = 10,3125$ Гбит/с).

Полоса частот нормализованного фильтра

$$f_nrm = fmax/\text{bps}/\text{points} \cdot (-\text{points}/2:\text{points}/2-1).$$

Оператор

$$s = f_nrm \cdot j;$$

Амплитудно-фазовая характеристика нормализованного фильтра для положительных и отрицательных частот нормализованных относительно частоты передачи сигнала показана на **Рисунок 7**. Логарифмическая амплитудно-частотная характеристика фильтра показана на **Рисунок 8**.

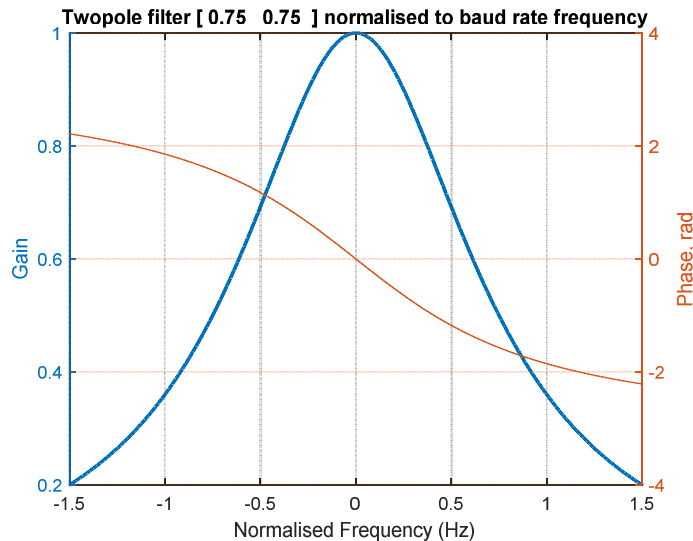


Рисунок 7. Амплитудно-фазовая характеристика нормализованного фильтра

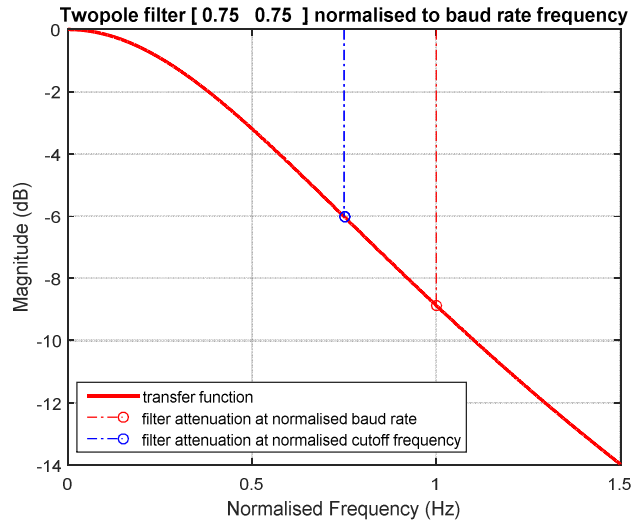


Рисунок 8. Логарифмическая амплитудно-фазовая частотная характеристика нормализованного фильтра. Синяя штриховая линия показывает положение частоты фильтра со значением 0,75 от частоты, на которой идет передача данных. На этой частоте ($1/T1 = 1/T2$) коэффициент передачи фильтра второго порядка равен -6 децибел. Красная штриховая линия показывает единичную частоту, на которой идет передача данных.

4. Перевод результатов измерения сигнальной линии к виду передаточной функции.

Измеренная амплитудно-фазовая характеристика сигнальной линии включает 1599 отсчетов в полосе до 20 ГГц с фиксированным шагом 12,475 МГц. Она содержит следующие значения частот: $channel.f = 0,006495:0,0012475:20$ ГГц. Изначально, сигнальная линия была представлена характеристикой четырехполюсника. Эта характеристика была преобразована и в примере используется в виде одномерной комплексной функции.

Частоты характеристики сигнальной линии, полученные в результате измерения, не совпадают с частотами спектра входного сигнала кратными частоте передачи данных. Кроме того, спектр сигнальной линии содержит только положительные частоты и не содержит частот в области нуля. Спектр входного сигнала содержит положительные, нулевую и отрицательные частоты.

Для преобразования характеристики сигнальной линии в передаточную функцию – характеристику, частоты которой совпадают с частотами спектра входного сигнала, выполнены следующие шаги.

- 4.1. Вычисление амплитуды характеристики линии на нулевой частоте путем ее экстраполяции. Для этого по десяти точкам амплитудной характеристики, ближайших к нулевой частоте, найдены коэффициенты линейного полинома, аппроксимирующего амплитудную характеристику:

```
[a] = polyfit(channel.f(1:10), channel.abs(1:10), 1);
```

Найденный второй коэффициент полинома равен амплитуде характеристики на нулевой частоте: $channel.dc = a(2)$;

- 4.2. Фазовая характеристика на нулевой частоте принята равной нулю.

```
channel.dcPhase = 0.00;
```

4.3. Пересчет амплитудной `channel.abs` и фазовой `channel.phase` характеристик сигнальной линии со значениями на нулевой частоте выполняется на частоты спектра входного сигнала ($f = -f_{max}/2:f_{max}/points:f_{max}/2-f_{max}/points$) с экстраполяцией характеристик в область нулевой и отрицательных частот:

```

ichannel.abs = interp1([0 channel.f], [channel.dc channel.abs],
abs(f), 'linear', 'extrap');
ichannel.phase = interp1([0 channel.f], [channel.dcPhase
unwrap(channel.phase)], abs(f), 'linear', 'extrap');
ichannel.s = ichannel.abs .* exp(+j.*ichannel.phase);
ichannel.tf = real(ichannel.s) + j*imag(ichannel.s) .* sign(f);

```

Полученная передаточная функция - амплитудно-фазовая частотная характеристика канала в области низких частот показана на **Рисунок 9**. Амплитудно-частотные характеристики измеренной сигнальной линии и расчетной передаточной функции в полных частотных диапазонах показаны на **Рисунок 10**. Эти же характеристики в фазовом пространстве показаны на **Рисунок 11**.

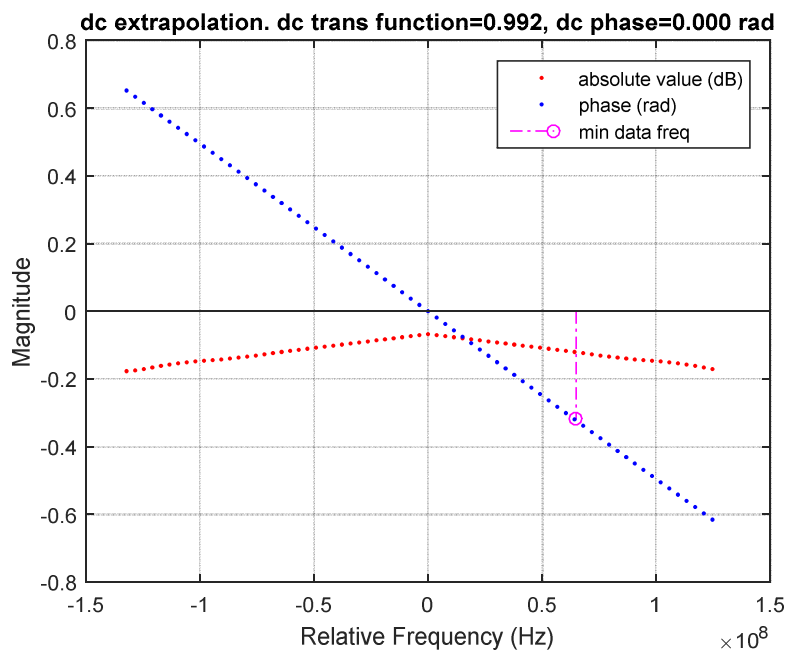


Рисунок 9. Передаточная функция сигнальной линии в области низких частот. Красными и синими точками обозначены дискретные амплитудная и фазовая характеристики соответственно. Амплитудная характеристика показана в децибелах, фазовая - в радианах. Розовой линией отмечена самая низкая частота измеренной характеристики сигнальной линии. Коэффициент передачи на нулевой частоте равен 0,992.

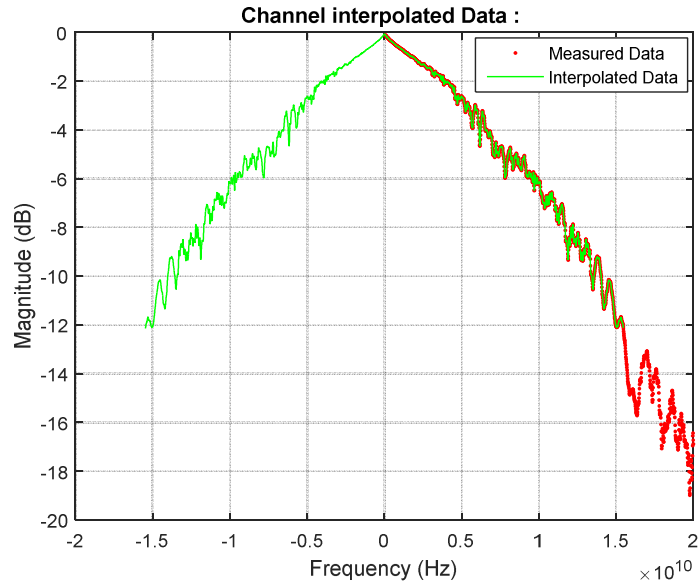


Рисунок 10. Амплитудно-частотные характеристики сигнальной линии. Синими точками обозначены комплексные данные измеренной линии. Расчетная симметричная зависимость усиления сигнальной линии на частотах спектра входного сигнала выделена красным. В области нулевых частот эта характеристика показана на **Рисунок 9**.

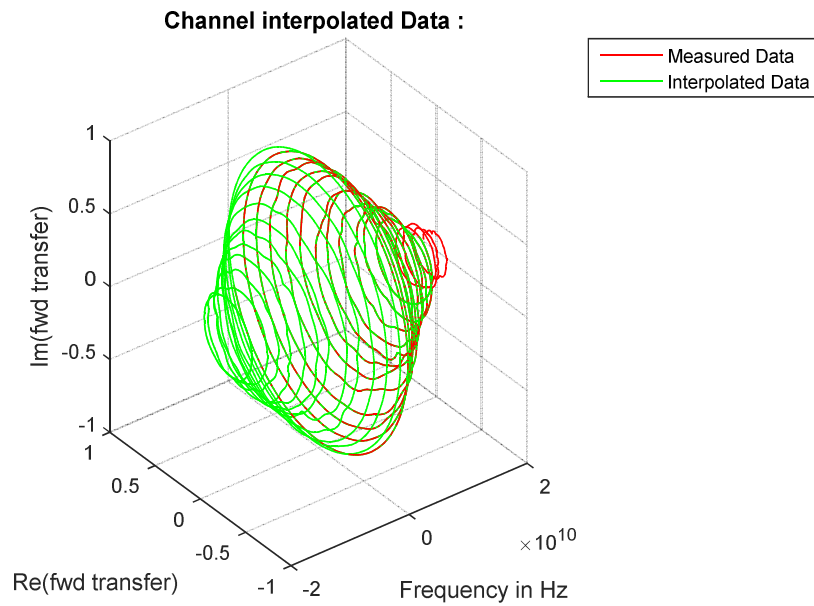


Рисунок 11. Амплитудно-фазовые частотные характеристики измеренной линии передачи данных и ее нормированного спектра.

5. Вычисление реакции сигнала.

Реакция (отклик на входное воздействие) в частотной области получается перемножением спектра сигнала на произведение передаточных функций элементов, которые связывают реакцию с входным сигналом. В нашем случае сигнал проходит через фильтр и сигнальную линию, поэтому реакция в частотной области равна произведению в комплексном виде спектров сигнала, фильтра и сигнальной линии.

Для перевода сигнала из частотной области во временную область используется обратное преобразование Фурье `ifft`.

Выходной сигнал фильтра во временной области `iSignal.Filter_out` вычисляется как

```
TransFunction.PSD = iSignal.PSD .* Filter.PSD_Tx;  
TransFunction.shiftedPSD = ifftshift(TransFunction.PSD);  
iSignal.Filter_out = real(ifft(TransFunction.shiftedPSD));
```

Выходной сигнал линии `iSignal.Rx` равен произведению спектра входного сигнала на передаточные функции фильтра и сигнальной линии с последующим переводом полученного сигнала из частотной области во временную область.

```
TransFunction.PSD = TransFunction.PSD .* ichannel.tf;  
TransFunction.shiftedPSD = ifftshift(TransFunction.PSD);  
iSignal.Rx = real(ifft(TransFunction.shiftedPSD));
```

Реакция фильтра на входной идеальный импульс и реакция канала показаны на **Рисунок 12**.

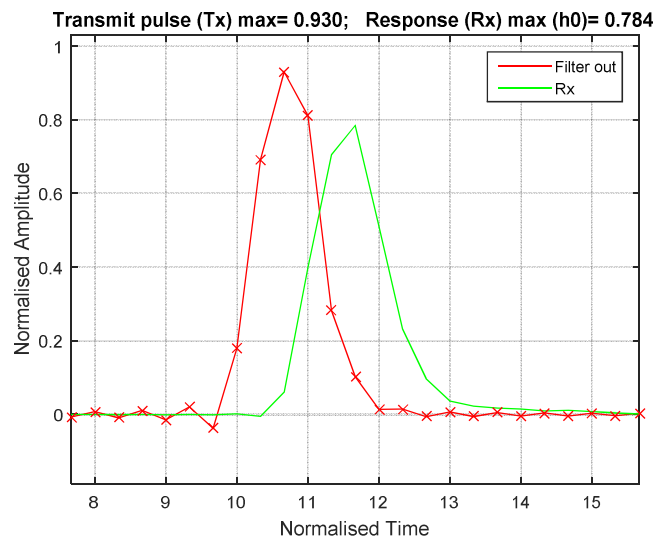


Рисунок 12. Выходной сигнал фильтра (красный график) и выходной сигнал линии передачи данных (зеленый график). Входной сигнал фильтра – единичный импульс показан на Рисунок 4. Входом сигнальной линии является выходной сигнал фильтра.

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие функции МАТЛАБ используются для прямого и обратного преобразования Фурье?
2. Как вычислить период дискретизации по времени периодического сигнала?
3. При каком отношении между частотой дискретизации и полосой периодического сигнала можно восстановить аналоговый сигнал без искажения?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. А.А.Алексеев, Д.Х.Имаев, Н.Н.Кузьмин, В.Б.Яковлев. Теория управления.: Учеб./СПб.: Изд-во СПбГЭТУ “ЛЭТИ”, 1999. – 435 с.
2. IEEE802.3ap. 10.3125Gbps NRZ Simulation Results Using “StatEye” and “Signal to Interference Model” on Cascaded Channel Components. Shannon Sawyer and Charles Moore / Agilent Technologies. January 24, 2005
http://www.ieee802.org/3/ap/public/jan05/sawyer_01_0105.pdf
3. What is StatEye. IEEE 803.3ap Task Force. September 16, 2004
http://www.ieee802.org/3/ap/public/signal_adhoc/ghiasi_01_0904.pdf
4. [Stat Eye / IBM Agreement. Steve Anderson. Xilinx, Inc.](http://www.ieee802.org/3/ap/public/nov04/anderson_01_1104.pdf)
http://www.ieee802.org/3/ap/public/nov04/anderson_01_1104.pdf
5. Dr. Bob Davidov. Компьютерные технологии управления в технических системах
<http://portalnp.ru/author/bobdavidov>

ПРИЛОЖЕНИЕ

СКРИПТ: Пример вычисления реакции на импульсное воздействие в МАТЛАБ.

```
clear all

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Ini data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
bps = 1.03125e+10;
FilterParam = [0.75 0.75];
points      = 2^13;
load('channel');

N = floor(max(channel.f)*2/bps);
fmax = N*bps;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Signal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% normalise all the scales for the bit rate
time      = bps/fmax .* (1:points);
iSignal.Tx(1:size(time,2)) = 0;
t0 = max(find(time<=10));
```

```

t1 = max(find(time<11));
iSignal.Tx(t0:t1) = 1.0;

figure
plot(time(1:t1+10), iSignal.Tx(1:t1+10),'b');
hold on
plot(time(1:t1+10), iSignal.Tx(1:t1+10),'xb');
grid on
xlabel('Normalised Time, tick Ts = 1/fmax');
ylabel('Normalised Amplitude');
title(['Pulse, data bit']);

iSignal.shiftedPSD = fft(iSignal.Tx);

figure
plot(abs(iSignal.shiftedPSD),'c');
grid on
xlabel('points, num');
ylabel('Amplitude');
title(['abs(fft(iSignal.Tx))']);

iSignal.PSD = fftshift(iSignal.shiftedPSD);

figure
plot(abs(iSignal.PSD),'r');
grid on
xlabel('points, num');
ylabel('Amplitude');
title(['abs(fftshift(fft(iSignal.Tx)))']);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Filter
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f_nrm = fmax/bps/points.*(-points/2:points/2-1);
s = f_nrm .* j;

Filter_PSD = 1 ./ (1 + s/FilterParam(1)) ./ (1 + s/FilterParam(2));

figure
[AX,H1,H2] = plotyy(f_nrm, abs(Filter_PSD), f_nrm, phase(Filter_PSD));
hold(AX(1));
hold(AX(2));
set(H1,'LineWidth',2);
grid(AX(2),'on');
xlabel('Normalised Frequency (Hz)');
set(get(AX(1),'Ylabel'),'String','Gain');
set(get(AX(2),'Ylabel'),'String','Phase, rad');
title(['Twopole filter [ sprintf(' %3.2f ', FilterParam) ] normalised to baud rate frequency']);

figure
plot_handles_Filter = plot(f_nrm(points/2 + 1:points), 20*log10(abs(Filter_PSD(points/2 + 1:points))), 'r', 'linewidth', 2);
hold on
stem_handles_br = stem(1, 20*log10(abs(Filter_PSD(max(find(f_nrm < 1))))), '-.ro');
hold on
stem_handles_c = stem(FilterParam, [20*log10(abs(Filter_PSD(max(find(f_nrm < FilterParam(1))))))
20*log10(abs(Filter_PSD(max(find(f_nrm < FilterParam(2))))))], '-.bo');

```

```

grid
legend_handles = [plot_handles_Filter, stem_handles_br(1), stem_handles_c(1)];
legend(legend_handles, 'transfer function', 'filter attenuation at normalised baud rate', 'filter attenuation at normalised cutoff
frequency', 3);
xlabel('Normalised Frequency (Hz)');
ylabel('Magnitude (dB)');
title(['Twopole filter [' sprintf(' %3.2f ', FilterParam) ] normalised to baud rate frequency']);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Channel
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% create negative frequencies, convert data to complex value, taking care about negative frequency
channel.abs = abs(channel.s);
channel.phase = angle(channel.s);

%channel.s = channel.abs .* exp(+j.*channel.phase);

[a] = polyfit(channel.f(1:10), channel.abs(1:10), 1);
channel.dc = a(2);
channel.dcPhase = 0.00;

fs = fmax/points; % frequency step
f = -fmax/2:fs:fmax/2-fs; % frequency matrix

% create new data structure with linearly interpolated data
ichannel.abs = interp1([0 channel.f], [channel.dc channel.abs], abs(f), 'linear', 'extrap');
ichannel.phase = interp1([0 channel.f], [channel.dcPhase unwrap(channel.phase)], abs(f), 'linear', 'extrap');

% correct for negative frequencies
ichannel.s = ichannel.abs .* exp(+j.*ichannel.phase);
ichannel.tf = real(ichannel.s) + j*imag(ichannel.s) .* sign(f);

figure
disp_points = 2*round(channel.f(1)/fs);
stem_handles_br = stem(channel.f(1), angle(ichannel.tf(max(find(f < channel.f(1))))), '-.mo');
hold on
plot_abs = plot(f(points/2-disp_points:points/2+disp_points), 20*log10(abs(ichannel.tf(points/2-
disp_points:points/2+disp_points))), '.r', 'linewidth', 3);
hold on
plot_phase = plot(f(points/2-disp_points:points/2+disp_points), angle(ichannel.tf(points/2-disp_points:points/2+disp_points)),
'.b', 'linewidth', 3);
grid
legend_handles = [plot_abs, plot_phase, stem_handles_br(1)];
legend(legend_handles, 'absolute value (dB)', 'phase (rad)', 'min data freq', 3);
xlabel('Relative Frequency (Hz)');
ylabel('Magnitude');
title(sprintf('dc extrapolation. dc trans function=%4.3f, dc phase=%4.3f rad', abs(ichannel.tf(points/2+1)),
angle(ichannel.tf(points/2+1))));

figure
plot(channel.f, 20*log10(channel.abs), '.r', 'linewidth', 3);
hold on
plot(f, 20*log10(ichannel.abs), 'g');
grid on

```



```

legend('Measured Data', 'Interpolated Data', 3);
xlabel('Frequency (Hz)');
ylabel('Magnitude (dB)');
title(['Channel interpolated Data : ']);

figure
plot3(channel.f, real(channel.s), imag(channel.s), 'r');
hold on
plot3(f, real(ichannel.tf), imag(ichannel.tf), 'g');
grid on
legend('Measured Data', 'Interpolated Data');
xlabel('Frequency in Hz');
ylabel('Re(fwd transfer)');
zlabel('Im(fwd transfer)');
title(['Channel interpolated Data : ']);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Response
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% filter Output
TransFunction.PSD = iSignal.PSD .* Filter_PSD;
TransFunction.shiftedPSD = ifftshift(TransFunction.PSD);
iSignal.Filter_out = real(ifft(TransFunction.shiftedPSD));

% pass through channel
TransFunction.PSD = TransFunction.PSD .* ichannel.tf;
TransFunction.shiftedPSD = ifftshift(TransFunction.PSD);
iSignal.Rx = real(ifft(TransFunction.shiftedPSD));

figure
plot(time, iSignal.Filter_out, 'r');
hold on
[max_Tx, time_maxTx] = max(iSignal.Filter_out);
[min_Tx, time_minTx] = min(iSignal.Filter_out);
[max_Rx, time_maxRx] = max(iSignal.Rx);
dtime_p5= round((time_maxRx - time_maxTx)*time(1) -1);
plot(time - dtime_p5, iSignal.Rx, 'g');
hold on
plot(time, iSignal.Filter_out, 'rx');
axis([(time_maxTx*time(1) - 3) (time_maxTx*time(1) + 5) (min_Tx-0.15) (max_Tx+0.1)])
grid on
legend('Filter out', 'Rx', 2);
xlabel('Normalised Time');
ylabel('Normalised Amplitude');
title(sprintf('Transmit pulse (Tx) max= %4.3f; Response (Rx) max (h0)= %4.3f', max(iSignal.Filter_out), max(iSignal.Rx)));

```