



Моделирование РТУиС

Лекция 3

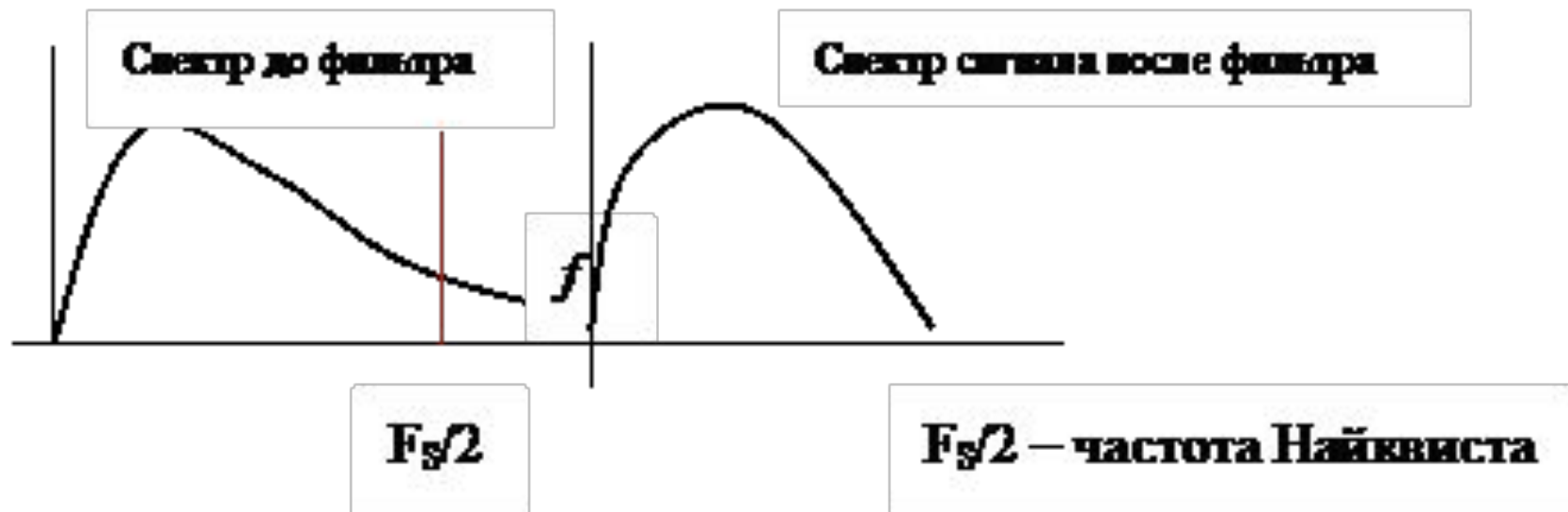
Вывод данных моделирования

Теорема Найквиста—Шеннона—Котельникова

Теорема гласит (упрощенно), что для сигнала, представленного последовательностью дискретных отсчетов, точное восстановление возможно, только если частота дискретизации более чем в 2 раза выше максимальной частоты в спектре сигнала.

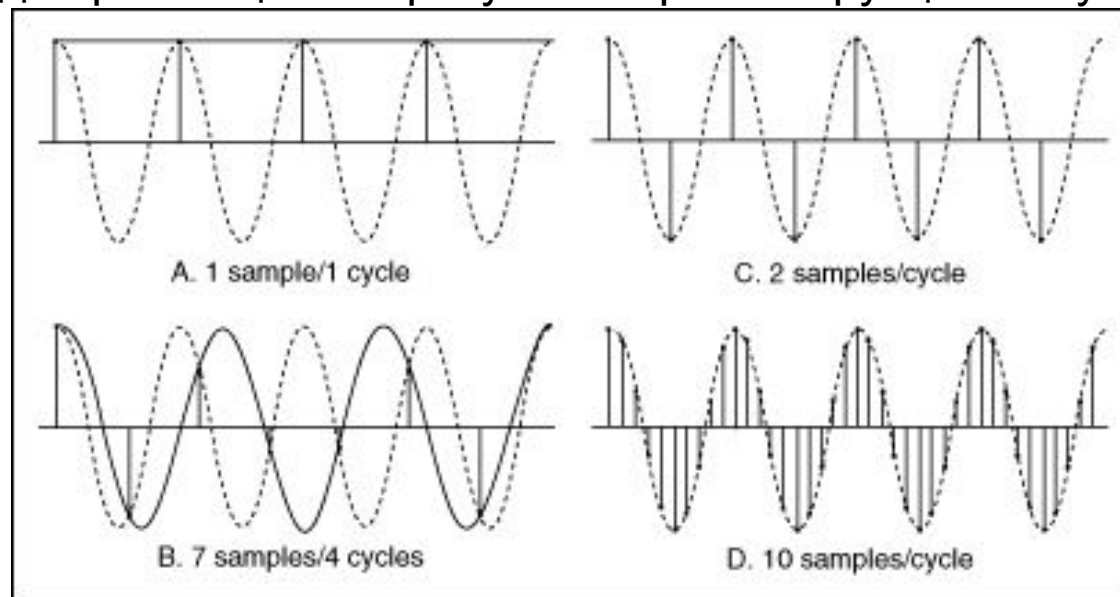
Из сказанного следует, что восстановить без искажений можно только сигнал, спектр которого ограничен некоторой частотой F_{\max} (частота Найквиста).

Однако, теоретически все реальные сигналы имеют бесконечные спектры. Спектры реальных сигналов, хотя и не бесконечны, но могут быть очень широкими. Для того чтобы при дискретизации избежать искажений, вызванных этим обстоятельством, сигнал вначале пропускают через фильтр, подавляющий в нем частоты, превышающие заданное значение F_{\max}' , и лишь затем производят дискретизацию:



Итак, согласно теореме, рассматривающей идеальные условия, частота дискретизации, с которой следует брать отсчеты, составляет не менее $2F_{\max}$. Однако, с учетом реальных свойств сигналов и устройств преобразования, восстановленный сигнал имеет произвольные амплитуду и фазу (в определенных пределах). Статистически достоверное восстановление исходного аналогового сигнала имеет место при частоте выборки не менее $5F$.

На картинке ниже: в случае А. частота сэмплирования f_s равна частоте синусоидальной волны f . Поэтому, здесь отсчет берется один раз за цикл. В результате реконструкции получается прямая



В случае В: $f_s = 7/4f$, или 7 отсчетов на 4 цикла волны. В этом случае восстановленный сигнал будет отличаться от исходного — 3 цикла вместо четырех.

В случае С: увеличение частоты сэмплирования до $f_s = 2f$ позволит получить сигнал той же частоты, что и исходный. Однако, если сдвинуть сигнал по фазе, мы увидим, что амплитуды исходного и восстановленного сигнала могут отличаться. Только увеличение частоты сэмплирования до значений, значительно больших f , например, $f_s = 10f$ (10 отсчетов за цикл), позволит точно восстановить сигнал (случай D).

История вопроса об основополагающей теореме взятия отсчетов

Хотя в западной литературе теорема часто называется теоремой Найквиста со ссылкой на работу «Certain topics in telegraph transmission theory» 1928 года, в этой работе речь идёт лишь о требуемой полосе линии связи для передачи импульсного сигнала (частота следования должна быть меньше удвоенной полосы). Таким образом, в контексте теоремы отсчётов справедливо говорить лишь о частоте Найквиста. Примерно в это же время Карл Кюпфмюллер получил тот же результат. О возможности полной реконструкции исходного сигнала по дискретным отсчётам в этих работах речь не идёт. Теорема была предложена и доказана Владимиром Котельниковым в 1933 году в работе «О пропускной способности эфира и проволоки в электросвязи», в которой, в частности, была сформулирована одна из теорем следующим образом: *«Любую функцию $f(t)$, состоящую из частот от 0 до f_c можно непрерывно передавать с любой точностью при помощи чисел, следующих друг за другом через $1/2f_c$ секунд»*. Независимо от него эту теорему в 1949 году (через 16 лет) доказал Клод Шеннон, поэтому в западной литературе эту теорему часто называют теоремой Шеннона. В 1999 году Международный научный фонд Эдуарда Рейна (Германия) признал приоритет Котельникова, наградив его премией в номинации «за фундаментальные исследования» за впервые математически точно сформулированную и доказанную в аспекте коммуникационных технологий теорему отсчётов. Исторические изыскания показывают, однако, что теорема отсчётов как в части утверждения возможности реконструкции аналогового сигнала по дискретным отсчётам, так и в части способа реконструкции, рассматривалась в математическом плане многими учёными и ранее. В частности, первая часть была сформулирована ещё в 1897 году Борелем.

Из теоремы Котельникова следует, что при дискретизации аналогового сигнала потерь информации не будет только в том случае, если (спектральная плотность) наивысшая частота полезного сигнала равна половине или меньше частоты дискретизации (в англоязычной литературе под обозначением половины частоты дискретизации употребляют термин частота Найквиста). В противном случае при восстановлении аналогового сигнала будет иметь место наложение спектральных «хвостов» (подмена частот, маскировка частот, [алиасинг](#)), и форма восстановленного сигнала будет искажена. Если спектр сигнала не имеет составляющих выше частоты Найквиста, то сигнал может быть (теоретически) продискретизирован и затем восстановлен без искажений. Фактически «оцифровка» сигнала (превращение аналогового сигнала в цифровой) сопряжена с квантованием отсчётов — каждый отсчёт записывается в виде цифрового кода конечной разрядности, в результате чего к отсчётам добавляются ошибки квантования (округления), при определённых условиях рассматриваемые как «шум квантования».

Реальные сигналы конечной длительности всегда имеют бесконечно широкий спектр, более или менее быстро убывающий с ростом частоты. Поэтому дискретизация сигналов всегда приводит к потерям информации (искажению формы сигнала при дискретизации—восстановлении), как бы ни была высока частота дискретизации. При выбранной частоте дискретизации искажение можно уменьшить, если обеспечить подавление спектральных составляющих аналогового сигнала (до дискретизации), лежащих выше частоты Найквиста, для чего требуется противоположенный фильтр очень высокого порядка, чтобы избежать наложения «хвостов». Практическая реализация такого фильтра весьма сложна, так как амплитудно-частотные характеристики фильтров имеют не прямоугольную, а гладкую форму, и образуется некоторая переходная полоса частот между полосой пропускания и полосой подавления. Поэтому частоту дискретизации выбирают с запасом, к примеру, в аудио компакт-дисках используется частота дискретизации 44100 Герц, в то время как высшей частотой в спектре звуковых сигналов, которую может услышать человек, считается частота 20000 Гц. Запас по частоте Найквиста в $44100 / 2 - 20000 = 2050$ Гц позволяет избежать подмены частот при использовании реализуемого фильтра невысокого порядка.

При доказательстве теоремы Котельникова взяты ограничения на спектр частот $0 < \omega < \omega_1$, $\omega = 2\pi f$.

Такая трактовка рассматривает идеальный случай, когда сигнал начался бесконечно давно и никогда не закончится, а также не имеет во временной характеристике точек разрыва. Если сигнал имеет разрывы любого рода в функции зависимости его от времени, то его спектральная мощность нигде не обращается в ноль. Именно это подразумевает понятие «спектр, ограниченный сверху конечной частотой f_c ».

Разумеется, реальные сигналы (например, звук на цифровом носителе) не обладают такими свойствами, так как они конечны по времени и обычно имеют разрывы во временной характеристике. Соответственно, ширина их спектра бесконечна. В таком случае полное восстановление сигнала невозможно, и из теоремы Котельникова вытекают следствия:

- любой аналоговый сигнал может быть восстановлен с какой угодно точностью по своим дискретным отсчётам, взятым с частотой $f > 2f_c$, где f_c — максимальная частота, которая ограничена спектром реального сигнала;
- если максимальная частота в сигнале равна или превышает половину частоты дискретизации (наложение спектра), то способа восстановить сигнал из дискретного в аналоговый без искажений не существует.

Говоря шире, теорема Котельникова позволяет представить непрерывный сигнал $x(t)$ в виде интерполяционного ряда

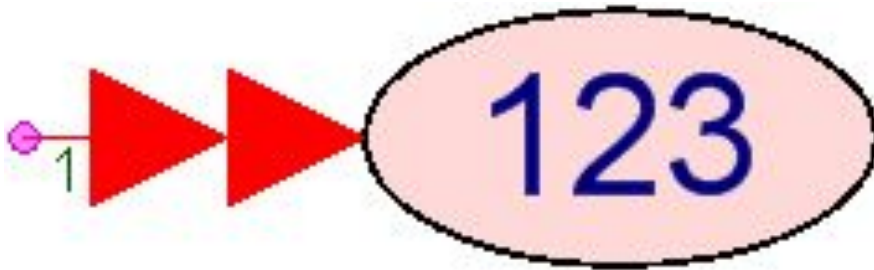
$$x(t) = \sum_{k=-\infty}^{\infty} x(k\Delta) \operatorname{sinc} \left[\frac{\pi}{\Delta} (t - k\Delta) \right],$$

Где $\operatorname{sinc}(x) = \sin(x)/x$ функция синк. Интервал дискретизации удовлетворяет ограничениям $0 < \Delta < 1/2f_c$. Мгновенные значения данного ряда есть дискретные отсчёты сигнала $x(k\Delta)$.

Элементы приемника данных (контроля моделируемой схемы и вывода данных из точек анализа)

- [BeamMeasurement Part](#)
- [BER FER Part](#)
- [BER IS Part](#)
- [CCDF Part](#)
- [Constellation Part](#)
- [DataPort Part](#)
- [DigitalDemod Part](#)
- [Dynamic3D Part](#)
- [EVM Part](#)
- [EyeDiagram Part](#)
- [FlexDCA Sink Part](#)
- [RMSE Part](#)
- [SINAD Part](#)
- [Sink Part](#)
- [SpectrumAnalyzer Part](#)
- [WriteBaseBandStudioFile Part](#)
- [WriteFile Part](#)

Приемник данных (универсальный вывод данных)



Может применяться в качестве осциллографа и/или спектроанализатора.

Port	Name	Description	Signal Type	Optional
1	input	Input Signal	multiple anytype	NO

Порт	Имя	Описание	Тип сигнала	Опции
1	Вход	Входной сигнал	Множество типов	НЕТ

Description: Data Sink

Domain: Timed

C++ Code Generation Support: NO

SystemVueEngine Support: YES (ContinuousMode=YES не будет вызывать динамические графики)

Associated Parts: [Sink Part](#)

При записи данных в файлы ни ASCII-файл, ни двоичный файл не предназначены для использования с другими продуктами Keysight (такими как программное обеспечение VSA 89601A). ASCII-файл используется исключительно для регистрации данных моделирования. Двоичный файл используется для временного хранения большого объема данных моделирования для последующей обработки скриптом MATLAB (см. SignalDownloader_E4438C в качестве примера использования скрипта MATLAB для последующей обработки двоичного файла).

Name	Description	Default	Units	Type	Range
StartStopOption	Режим сбора данных: Авто, Отсчеты, Время	Авто		Enumeration	
SampleStart	Номер отсчета для начала сбора данных	0		Integer	[0:∞)
SampleStop	Номер отсчета остановки сбора данных	Num_Samples - 1		Integer	[SampleStart:∞)
TimeStart	Время начала сбора данных	Start_Time	s	Float	[0:∞)
TimeStop	Время окончания сбора данных	Stop_Time	s	Float	[TimeStart:∞)
SinkTarget	Место, куда записываются данные моделирования: набор данных, файл, оба варианта	Набор данных		Enumeration	
ContinuousMode	Запуск моделирования в непрерывном режиме: НЕТ, ДА	НЕТ		Enumeration	
WindowSize	Количество выборок для сбора и отображения в динамически обновляемом графике	500		Integer	[1:∞)
DisableDataCollect	Отключить сбор данных: НЕТ, ДА	НЕТ		Enumeration	
Graph	График для отображения данных в нем (если он пуст, то график не создается)			Text	
Table	Таблица для отображения данных в ней (если она пуста, то таблица не создается)			Text	
ToSingleFile	Запись данных со всех входов в один и тот же файл: НЕТ, ДА	ДА		Enumeration	
DataFileName	Имя файла данных	SinkData		Filename	
DataFileType	Формат файла данных: ASCII, Binary, SignalStudio, N5106A, CSV	ASCII		Enumeration	
InsertFileHeader	Включить вставку заголовка файла перед содержимым csv-файла: НЕТ, ДА	НЕТ		Enumeration	
SkipFrequency	Не писать частотную характеристику (для огибающей комплексного сигнала) в файл: НЕТ, ДА	ДА		Enumeration	
NormalizeWaveform	Нормализовать величину входного сигнала до диапазона [-1,1] перед записью в файл: НЕТ, ДА	ДА		Enumeration	
ClippingLevel	Уровень отсечки по напряжению	1.0	V	Float	(0:∞)
Persistence	Хранить значения в наборе данных при сохранении рабочей области: НЕТ, ДА	Data_Persistence		Enumeration	
Overlap	Процент перекрытия данных в динамическом графике	0		Float	[0:100)
EmptyMatrixExpansion	Показать сигнальную выборку в диапазоне задержки как нулевую или пустую: Zero_Value, Empty_Matrix	Zero_Value		Enumeration	

Анализаторы спектра. Общие сведения

- **По диапазону частот** — низкочастотные, радиодиапазона (широкополосные) и оптического диапазона.
- **По принципу действия** — параллельного типа (многоканальные) и последовательного типа (сканирующие).
- **По способу обработки измерительной информации и представлению результатов** — аналоговые и цифровые.
- **По характеру анализа** — скалярные, дающие информацию только об амплитудах гармонических составляющих спектра, и векторные, предоставляющие также информацию о фазовых соотношениях.

Основные свойства анализа

Анализатор спектра позволяет определить амплитуду и частоту спектральных компонентов, входящих в состав анализируемого процесса. Важнейшей его характеристикой является разрешающая способность: наименьший интервал Δf по частоте между двумя спектральными линиями, которые ещё разделяются анализатором спектра. Анализатор спектра может дать истинный спектр только тогда, когда анализируемое колебание $x(t)$ существует только в пределах

не истинный сп

$$S_T(t_1, \omega) = \int_{t_1}^{t_1+T} x(t)e^{-i\omega t} dt,$$

изе длительностей процес $S(\omega) = \int_{-\infty}^{\infty} x(t)e^{-i\omega t} dt$, а даёт

а его оценку, зависящую от времени включения t_1 и времени анализа T . Так как спектр колебания может в общем случае изменяться во времени, то оценка даёт т. н. текущий

Низкочастотные анализаторы

НЧ анализаторы бывают параллельного и последовательного типа (чаще параллельного) и предназначены для работы в диапазонах частот от нескольких герц до десятков — сотен килогерц. Используются в акустике, например, при исследовании характеристик шума, при разработке и обслуживании аудиоаппаратуры и в других целях. Анализаторы, используемые для контроля качества питающей электросети, иначе называются **анализаторами гармоник**.

ПРИМЕРЫ: UPV, UPR, Ф4327, С4-34, СК4-83, ZET017, FLUKE_41B (анализатор гармоник), ZET 017.

Радиочастотные

анализаторы

Большинство радиочастотных анализаторов являются широкополосными, позволяют работать в полосе от нескольких килогерц до единиц — сотен гигагерц, как правило, это анализаторы последовательного типа. Применяются для анализа свойств радиосигналов, для исследования характеристик радиоустройств.

•**ПРИМЕРЫ:** FSL, FSV, FSU, СК4-84, С4-82, GSP-810

Анализаторы последовательного

типа

Анализаторы последовательного типа являются наиболее распространенным видом анализаторов для исследования радиосигналов, принцип их действия состоит в сканировании полосы частот с помощью перестраиваемого гетеродина. Составляющие спектра последовательно переносятся на промежуточную частоту. Перестройка частоты гетеродина эквивалентна перемещению спектра исследуемого сигнала. Селективный УПЧ последовательно выделяет составляющие спектра, и, благодаря синхронной развёртке осциллографического индикатора, отклики каждой спектральной составляющей последовательно воспроизводятся на его экране.

Анализаторы параллельного типа

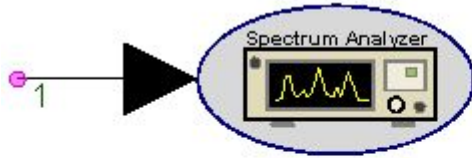
Анализаторы параллельного типа содержат набор идентичных узкополосных фильтров (высокодобротных резонаторов), каждый из которых настроен на определенную частоту (в области низкочастотных измерений фильтры могут иметь одинаковой не абсолютную полосу пропускания, а относительный частотный интервал, например, «третьоктавные фильтры»). При одновременном воздействии исследуемого сигнала на все фильтры каждый из них выделяет соответствующую его настройке составляющую спектра. Параллельный анализатор спектра имеет перед последовательным преимущество в скорости анализа, однако уступает ему в простоте.

Цифровые

анализаторы

Цифровые анализаторы могут быть реализованы двумя способами. В первом случае это обычный анализатор последовательного типа, в котором измерительная информация, полученная методом сканирования полосы частот с помощью гетеродина, оцифровывается с помощью АЦП и, далее, обрабатывается цифровым методом. Во втором случае реализуется цифровой эквивалент параллельного типа в виде ДПФ-анализатора, который вычисляет спектр с помощью алгоритмов дискретного преобразования Фурье (ДПФ). По сравнению с последовательными цифровые параллельные ДПФ-анализаторы обладают определёнными преимуществами: более высоким разрешением и скоростью работы, возможностью анализа импульсных и однократных сигналов. Они способны вычислять не только амплитудный, но и фазовый спектры, а также одновременно представлять сигналы во временной и частотной областях. К сожалению, параллельные ДПФ-анализаторы из-за ограниченных возможностей аналого-цифровых преобразователей (АЦП) работают только на относительно низких частотах.

Корпорация Tektronix создала цифровые анализаторы спектра реального времени. Они позволяют отслеживать в реальном масштабе времени быстрые изменения спектра, которые используются в некоторых видах современных коммуникационных систем. При этом, наряду с обычными спектрами, приборы предоставляют стрелит, спектрограммы, которые предоставляют собой множество спектров, предоставленных в



Анализатор спектра комплексной огибающей сигналов

Description: Spectrum Analyzer for Complex Envelope Signals

Domain: Timed

C++ Code Generation Support: NO

SystemVueEngine Support: YES (ContinuousMode=YES will NOT bring up dynamic graphs)

Associated Parts: [SpectrumAnalyzer Part](#)

Port	Name	Description	Signal Type	Optional
1	вход	Входной сигнал	огибающая	HET

Name	Description	Default	Units	Type	Runtime Tunable	Range
Mode	Режим работы: TimeGate, ResBW	TimeGate		Enumeration	NO	
Start	Время старта сбора данных	Start_Time	s	Float	NO	[0:∞)
SegmentTime	Время сегмента	Stop_Time - Start_Time + Time_Spacing	s	Float	NO	[16*TStep:∞)†
ResBW	Разрешение по частоте	Freq_Resolution	Hz	Float	NO	(0:∞)
NumSegments	Число обрабатываемых сегментов (массивов) данных	1		Integer	NO	[1:∞)
Overlap	Перекрывание сегментов в %	0.0		Float	NO	[0:100)
Window	Виды оконных весовых функций при обработке выводимых данных: Uniform (прямоугольная), Hanning (Хэннинга), Gaussian Top (Гаусса с плоской вершиной), Flat Top (плосковершинное), Blackman Harris (Блэкманна-Харриса), Hamming (Хэмминга)	Uniform		Enumeration	NO	
FStart	Начальная частота	0.0	Hz	Float	NO	[0:∞)
FStop	Конечная частота	100.0e9	Hz	Float	NO	(FStart:∞)
SpectrumType	Тип выхода: Комплексное напряжение, Мощность/Фаза	Power/Phase		Enumeration	NO	
RefR	Выходное сопротивление	50.0	ohm	Float	NO	(0:∞)
Persistence	Сохранить спектральные данные в рабочей области проекта: НЕТ, ДА	Data_Persistence		Enumeration	NO	
ContinuousMode	Запуск моделирования в непрерывном режиме: НЕТ, ДА	Нет		Enumeration	NO	

Модель анализатора спектра может быть использована для измерения спектра действительной составляющей сигнала в пределах основной полосы или огибающей комплексного сигнала. В следующих примечаниях TStep используется для обозначения временного шага моделирования, а f_c - для обозначения частоты сигнала ($f_c = 0$ для реального сигнала основной полосы и $f_c > 0$ для комплексной огибающей сигнала).

Примечания к анализатору комплексной огибающей сигнала

Анализатор спектра имеет два режима работы: Time Gate и ResBW. Режим работы выбирается в параметре Mode. В обоих режимах работы модель собирает данные, начиная с момента запуска и до тех пор, пока не будут собраны сегменты NumSegments, которые перекрываются на Overlap% перекрытия. Каждый сегмент выделяется заданным окном, вычисляется спектр каждого сегмента и, наконец, спектры усредняются. Единственное различие между этими двумя режимами работы заключается в том, что в режиме TimeGate продолжительность каждого сегмента равна SegmentTime, а в режиме ResBW длительность каждого сегмента равна $NENBW/ResBW$, где NENBW - нормализованная эквивалентная полоса пропускания шума используемого окна.

Спектр сигнала основной полосы частот простирается от 0 Гц до $1/(2 \times TStep)$ Гц. Спектр радиочастотного сигнала простирается от $f_c - 1/(2 \times TStep)$ Гц до $f_c + 1/(2 \times TStep)$ Гц. Когда $f_c < 1/(2 \times TStep)$, спектр расширяется до отрицательных частот. Спектральное содержание на этих отрицательных частотах комплексно сопряжено, отзеркалено и добавлено к содержимому спектра ближайшей положительной частоты. Таким образом, отрицательные частотные гармоники отображаются на положительной частотной оси, как это происходит в реальном измерительном приборе анализатора спектра. Этот процесс может привести к ошибке в отображаемой частоте для зеркальных гармоник. При этом абсолютная ошибка меньше $\Delta f/2$, где Δf равно $1/SegmentTime$ (в режиме TimeGate) и $ResBW/NENBW$ (в режиме ResBW).

Примечания к анализатору комплексной огибающей сигнала

При обработке и усреднении нескольких сегментов усреднение фаз (для тонов, представляющих шум) приводит к тому, что усредненная фаза сходится к 0. Когда такие тоны существуют на отрицательных частотах ($f_c < 1/(2 \times T_{Step})$) и через процесс сопряжения и зеркального отображения они добавляются к эквивалентным тонам положительной частоты, которые также представляют шум, результирующая мощность шума может быть на целых 3 дБ выше, чем ожидалось. Это связано с тем, что при добавлении двух тонов одинаковой частоты, амплитуды и фазы они всегда когерентно складываются, увеличивая суммарную мощность на 6 дБ. Напротив, при добавлении двух тонов одинаковой частоты и амплитуды, но со случайными фазами, они (в среднем) приводят к тону с большей мощностью на 3 дБ. Параметры FStart и FStop используются для управления частотным диапазоном, в котором будет рассчитываться спектр. Если FStart и Fstop находятся за пределами указанных выше пределов, они сбрасываются на нижний и верхний пределы соответственно. Параметр SpectrumType используется для выбора типа вычисляемого спектра. Вычисленный спектр всегда является односторонним спектром. Если SpectrumType-это комплексное напряжение, то вычисленный спектр-это спектр комплексной амплитуды сигнала. Результат сохраняется в наборе данных в комплексной переменной с именем <InstanceName>, где <InstanceName> - имя экземпляра модели. Если SpectrumType-Power/Phase, то вычисляются два спектра: спектр мощности и фазовый спектр. Результаты сохраняются в наборе данных в двух реальных переменных с именами <InstanceName>_Power и <InstanceName>_Phase, где <InstanceName> - имя экземпляра модели.

Оконная обработка

Оконные окна необходимы при оценке спектра на основе преобразования (БПФ). Без оконного отображения расчетный спектр может содержать ошибки измерений или маскировку слабых спектральных деталей сигнала паразитными артефактами. Каждый раз, когда к сигналу в спектральной области применяется окно, происходит утечка мощности, то есть мощность от одной спектральной составляющей просачивается в соседние. Утечка от сильных спектральных компонентов (за счет боковых лепестков оконной функции) может привести к скрытию/маскировке близлежащих более слабых спектральных компонентов. Даже сильные спектральные компоненты могут быть затронуты утечкой. Например, две сильные спектральные компоненты, близкие друг к другу, могут проявиться как одна из-за утечки. Выбор правильного окна для спектрального измерения очень важен. Выбор окна зависит от того, что измеряется и каковы компромиссы между частотным разрешением (способностью различать близкие друг к другу спектральные компоненты сопоставимой интенсивности или мощности) и динамическим диапазоном. Окна могут быть охарактеризованы по их нормированным эквивалентам пропускной способности подавления шумов (NENBW). Эквивалентная полоса пропускания шума (ENBW) сравнивает окно с идеальным прямоугольным фильтром. Это эквивалентная ширина прямоугольного фильтра, пропускающего такое же количество белого шума, как и окно. Нормализованный ENBW (NENBW) - это ENBW, умноженный на длительность оконного сигнала. В общем случае для одной и той же длины обрабатываемого сигнала, чем выше NENBW окна, тем выше его динамический диапазон (меньше искажения) и тем хуже его частотное разрешение. NENBW окон, доступных в спектроанализаторе, приведен в таблице ниже:

Window	NENBW	Typical Uses
Uniform	1	Analyzing transients or periodic signals
Hanning	1.5	Good general purpose window
Gaussian Top	2.215	Minimize leakage
Flat Top	3.819	For accurate amplitude measurements
Blackman Harris	2.021	Minimize leakage
Hamming	1.363	For closely spaced frequencies

Рекомендации по применению типов окон

- Не используйте окно (установите окно на Равномерное) при анализе переходных процессов.
- Для периодических сигналов, спектральные компоненты которых имеют сравнимую амплитуду и когда обрабатываемый сегмент сигнала включает точное целое число, кратное периодам, наилучшие результаты получаются, если не используется окно (установить окно на Uniform). Любые переходные процессы запуска должны быть исключены.
- Для периодических сигналов, спектральные компоненты которых имеют значительно различающиеся амплитуды и/или когда обрабатываемый сегмент сигнала не включает точное целое число, кратное периодам, использование окна может улучшить обнаружение более слабых спектральных компонентов. Чем выше NENBW, тем более вероятно, что будут обнаружены более слабые спектральные компоненты. Однако это меняет частотное разрешение, и поэтому, если спектральные компоненты находятся очень близко друг к другу, более слабая из них может остаться неразрешенной. Для улучшения частотного разрешения при сохранении хорошего динамического диапазона используйте окно, но обрабатывайте более длинный сегмент сигнала.
- Для аперiodических сигналов, таких как модулированные сигналы (QPSK, QAM, GSM, EDGE, CDMA, OFDM), настоятельно рекомендуется использовать окно. Окно ослабит сигнал на обоих концах обрабатываемого сегмента сигнала до нуля. Это делает сигнал квазипериодическим и уменьшает ошибки при измерениях.

Аналитические выражения для весовых функций (окон)

$$\text{Uniform } w(k) = \begin{cases} 1.0 & \text{if } 0 \leq k \leq N \\ 0.0 & \text{otherwise} \end{cases}$$

$$\text{Hanning } w(k) = \begin{cases} 0.5 - 0.5 \cdot \cos\left(\frac{2\pi k}{N}\right) & \text{if } 0 \leq k \leq N \\ 0.0 & \text{otherwise} \end{cases}$$

$$\text{Gaussian Top } w(k) = \begin{cases} 0.323 - 0.471 \cdot \cos\left(\frac{2\pi k}{N}\right) + 0.176 \cdot \cos\left(\frac{4\pi k}{N}\right) - 0.0285 \cdot \cos\left(\frac{6\pi k}{N}\right) + 0.00126 \cdot \cos\left(\frac{8\pi k}{N}\right) & \text{if } 0 \leq k \leq N \\ 0.0 & \text{otherwise} \end{cases}$$

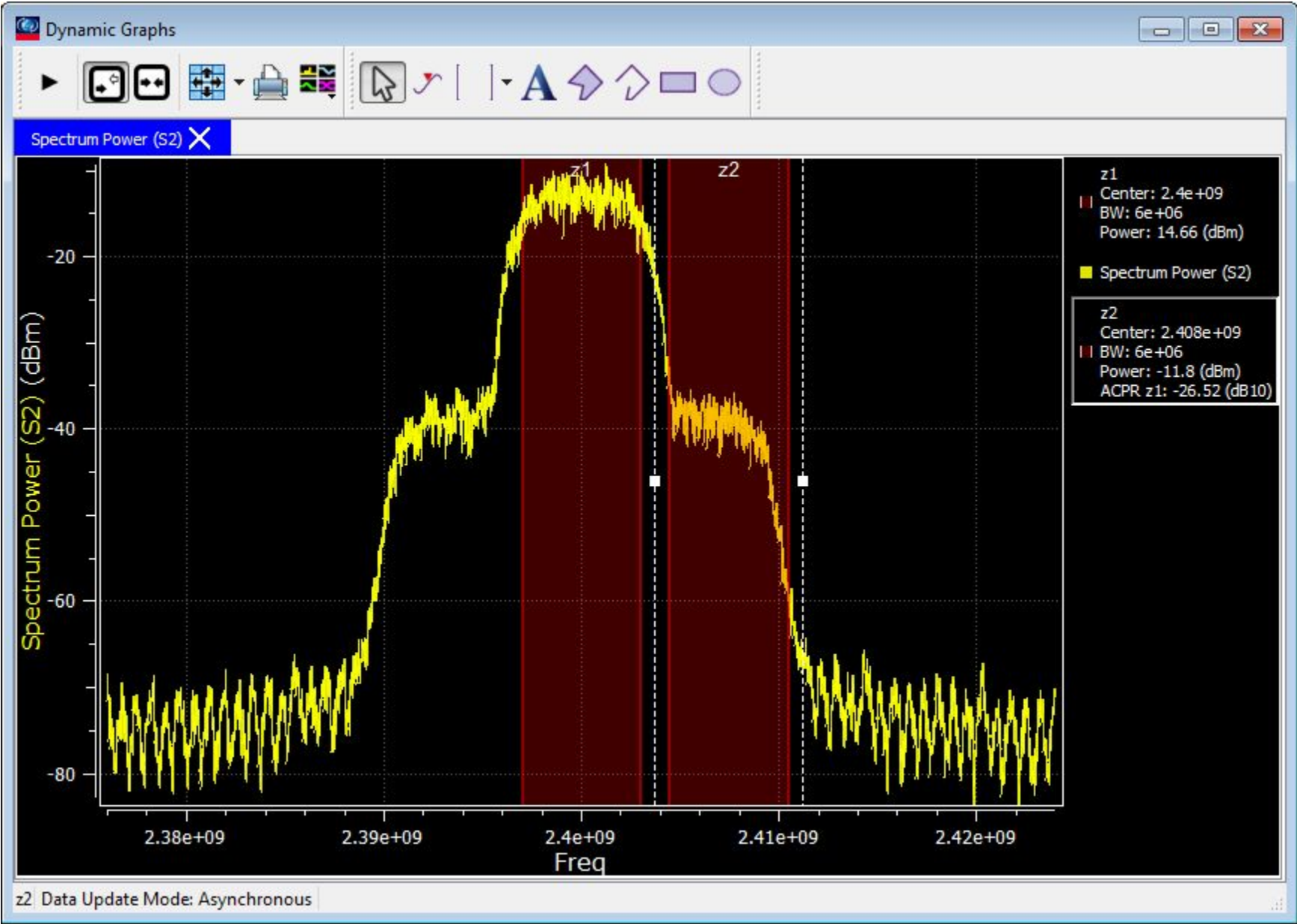
$$\text{Flat Top } w(k) = \begin{cases} 0.213 - 0.414 \cdot \cos\left(\frac{2\pi k}{N}\right) + 0.279 \cdot \cos\left(\frac{4\pi k}{N}\right) - 0.0862 \cdot \cos\left(\frac{6\pi k}{N}\right) + 0.00749 \cdot \cos\left(\frac{8\pi k}{N}\right) & \text{if } 0 \leq k \leq N \\ 0.0 & \text{otherwise} \end{cases}$$

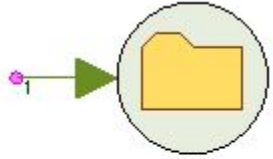
$$\text{Blackman Harris } w(k) = \begin{cases} 0.355768 - 0.487396 \cdot \cos\left(\frac{2\pi k}{N}\right) + 0.144232 \cdot \cos\left(\frac{4\pi k}{N}\right) - 0.012604 \cdot \cos\left(\frac{6\pi k}{N}\right) & \text{if } 0 \leq k \leq N \\ 0.0 & \text{otherwise} \end{cases}$$

$$\text{Hamming } w(k) = \begin{cases} 0.54 - 0.46 \cdot \cos\left(\frac{2\pi k}{N}\right) & \text{if } 0 \leq k \leq N \\ 0.0 & \text{otherwise} \end{cases}$$

Когда сигнал временной области обрабатывается окном, отличным от прямоугольного, его мощность уменьшается. Анализатор SpectrumAnalyzerEnv нормализует свой выходной спектр таким образом, чтобы мощность, содержащаяся в нем, была такой же, как мощность входного сигнала временной области, независимо от того, какое окно используется.

Параметр Persistence можно использовать для уменьшения размера файла рабочей области, не сохраняя переменные, генерируемые этой моделью при закрытии рабочей области. Спектральные данные могут содержать тысячи частотных точек, что может значительно увеличить размер рабочей области, особенно если в рабочей области имеется несколько моделей спектроанализатора. Если параметр Persistence установлен в значение NO, то при закрытии рабочей области переменные, сгенерированные этой моделью, удаляются из соответствующих наборов данных. Используйте этот параметр с осторожностью, например если время моделирования для генерации спектральных данных велико, вы, вероятно, захотите сохранить данные в рабочей области независимо от того, насколько увеличится ее размер. По умолчанию этот параметр имеет значение Data_Persistence, которое является переменной, управляемой флажком Data Persistence на вкладке Data Flow Analysis Options. Параметр ContinuousMode включает/отключает динамический график, который будет создан. Динамический график всегда будет отображать спектр мощности сигнала, даже если параметр SpectrumType установлен на Complex Voltage. Фазовый спектр недоступен в виде динамического графика.





Запись данных моделирования в файл

Description: N6030 Formatted File Writer
Domain: Timed
C++ Code Generation Support: NO
SystemVueEngine Support: YES
Associated Parts: [WriteFile Part](#)

Port	Name	Description	Signal Type	Optional
1	input	IQ data	complex	NO

Name	Description	Default	Units	Type	Range
StartStopOption	Sink collection mode: Auto, Samples, Time	Auto		Enumeration	
SampleStart	Sample number to start data collection	0		Integer	[0:∞)
SampleStop	Sample number to stop data collection	Num_Samples - 1		Integer	[SampleStart:∞)
TimeStart	Time to start data collection	Start_Time	s	Float	[0:∞)
TimeStop	Time to stop data collection	Stop_Time	s	Float	[TimeStart:∞)
FileI	I input file name	file_i.bin		Filename	
FileQ	Q input file name	file_q.bin		Filename	
NormalizeData	Normalize data to avoid clipping: NO, YES	YES		Enumeration	
FullScaleFactor	Full scale factor for peak value	1.0		Float	[0.000030517578125:1]

Опции компонента записи в файл

Данная модель записывает реальную и мнимую части комплексных данных в файлы *FileI* и *FileQ* соответственно с использованием формата N6030. Данные записываются в виде 16-битных целых чисел со знаком.

IQ данные представляются в виде комплексного значения со значением I в действительной части и значением Q в мнимой части.

Параметр Start Stop Option указывает режим сбора данных:

- если установлено значение Samples, то сбор данных будет начинаться с отсчета Sample Start и останавливаться на SampleStop;
- если установлено значение Time, сбор данных начнется в момент времени Time Start и остановится в момент времени TimeStop;
- если установлено значение авто, то все параметры SampleStart, SampleStop, TimeStart, и TimeStop должны быть заранее определены и симулятор сам будет выбирать в реальном времени что использовать: SampleStart/SampleStop или TimeStart/TimeStop. Решение основано на том, имеет ли входной сигнал заданную частоту дискретизации. Если частота дискретизации не определена (например, сигнал сгенерирован SineGen с параметром частоты дискретизации SampleRateOption, установленным в UnTimed), то используется SampleStart/SampleStop. В противном случае используется TimeStart/TimeStop.

Параметры FileI и FileQ используются для определения выходных файлов для данных I и Q. Файлы могут быть указаны с полным или относительным именем пути. Относительные имена путей относятся к каталогу, в котором находится проект.

Параметр Normalize Data определяет, будут ли данные обрезаны или нормализованы перед их записью в файлы:

- если установлено значение NO (НЕТ), нормализация не выполняется. Значения данных за пределами диапазона [-FullScaleFactor, FullScaleFactor] вырезаются в пределах этого диапазона. Кроме того, отображается предупреждающее сообщение, информирующее пользователя о том, что произошла обрезка данных;
- если установлено значение ДА, отсечение не выполняется, и данные, записанные в файлы, являются масштабированной (нормализованной) версией входных данных, так что они всегда находятся в диапазоне [-FullScaleFactor, FullScaleFactor]. Используемый коэффициент масштабирования отображается в информационном сообщении в окне Errors, а также в лог-файле моделирования.

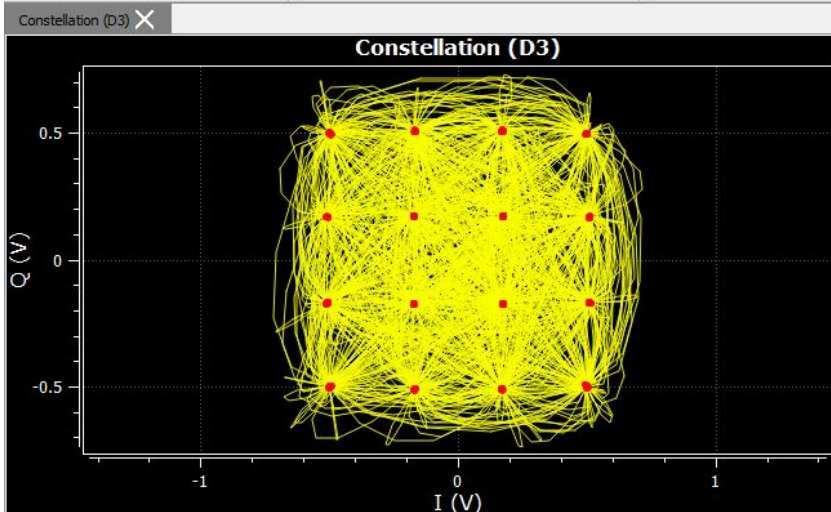
Формат N6030 записи данных в файл

N6030 считывает данные I и Q из файла и файла форматированных файлов N6030 для формирования вывода комплексных значений IQ. Если один файл длиннее другого, то более короткий файл будет заполнен нулями. Для этого условия генерируется предупреждение. Данные IQ представляются в виде комплексного значения со значением I в действительной части и значением Q в мнимой. Заполнение проводится в формате $0.0 + j \cdot 0.0$.

Name	Description	Default	Units	Type	Range	Symbol
FileI	Input file name	file_i.bin		Filename		FI
FileQ	Input file name	file_q.bin		Filename		FQ
Periodic	Repeat file data when end of file is reached: NO, YES	YES		Enumeration		P
ShowAdvancedParams	Show advanced parameters: NO, YES	NO		Enumeration		
SampleRateOption	Sample rate option: UnTimed, Timed from SampleRate, Timed from Schematic	Timed from Schematic		Enumeration		SO
SampleRate	Explicit sample rate	Sample_Rate	Hz	Float	(0:∞)	S
InitialDelay	Output sample delay	0		Integer	[0:∞)	D
BurstMode	Burst mode: OFF, Single, Multiple	OFF		Enumeration		
BurstLength	Burst sample length	100		Integer	[1:∞)	BL
BurstPeriod	Samples from start of one burst to start of next	200		Integer	[1:∞)	BP
BurstDelay	Sample delay within burst before the start of the burst length interval	0		Integer	[0:BurstPeriod-BurstLength]	DB

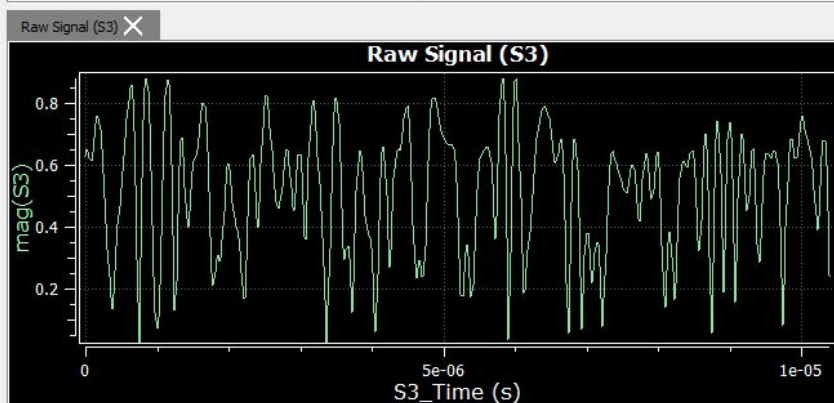
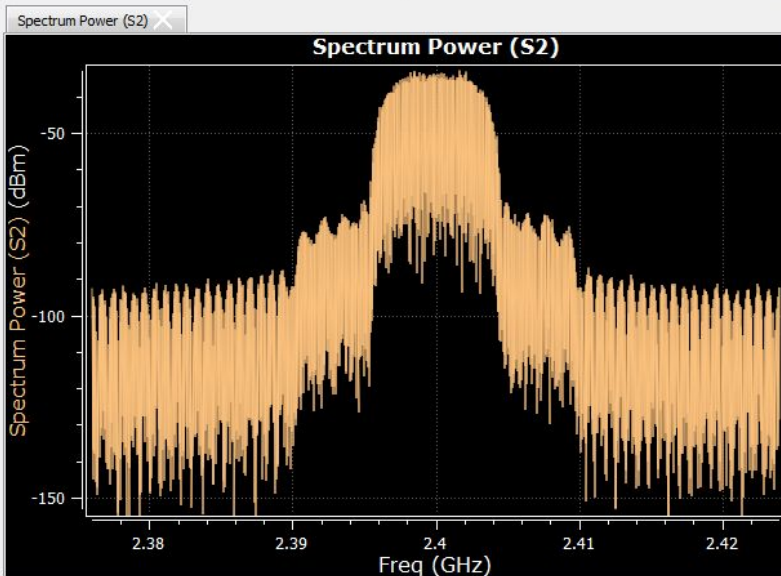
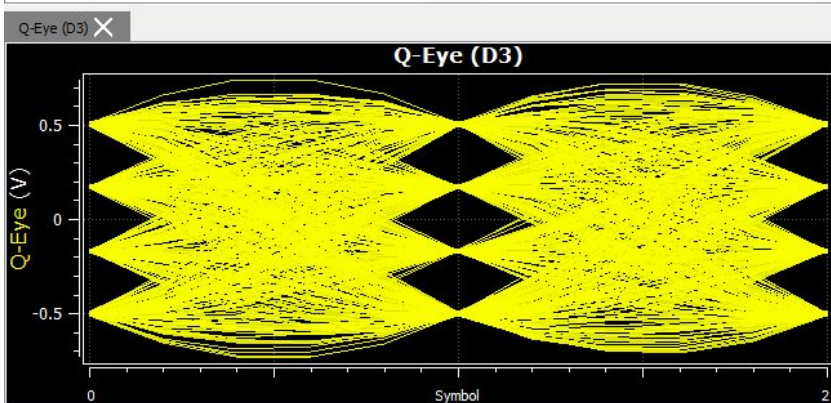
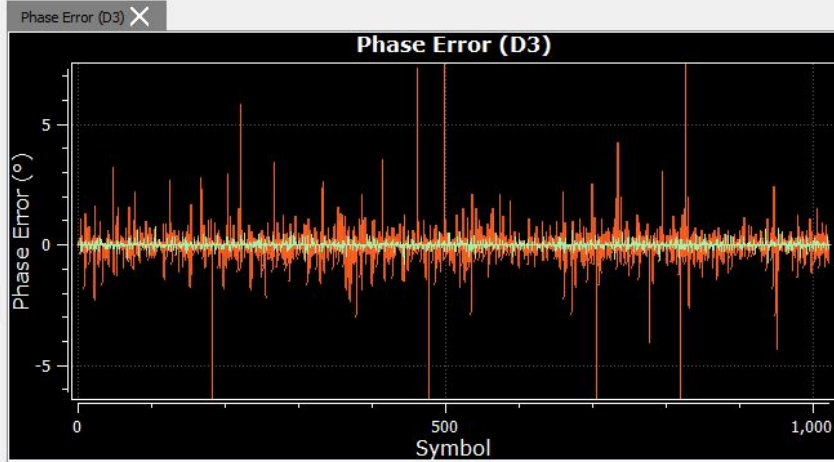
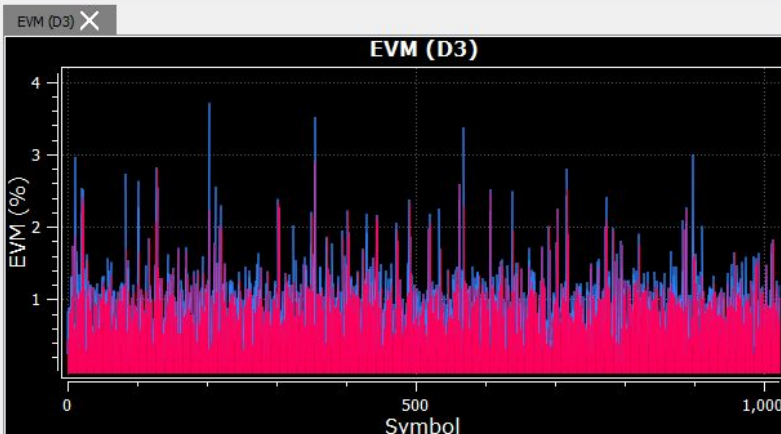
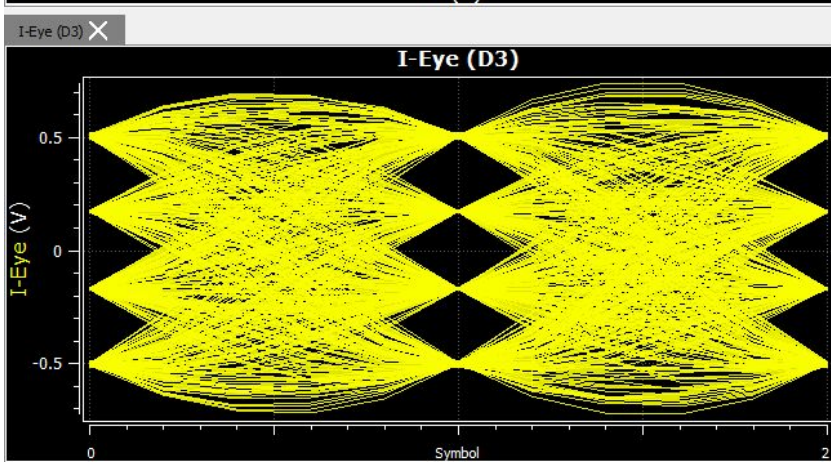
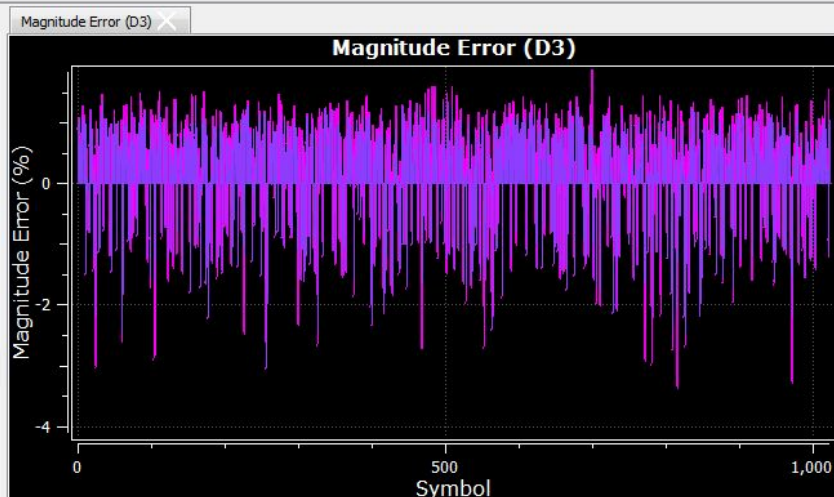
Динамические графики

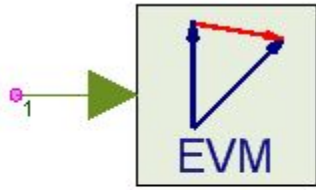
Окно Dynamic Graphs (Динамических графиков) относится к отдельному окну, которое автоматически открывается при некотором выбранном анализе SystemVue, настроенном с помощью динамических приемников, поддерживающих мгновенное отображение результатов во время анализа. Этим окном можно управлять отдельно с точки зрения его расположения и размера—другими словами, оно не ограничено главным окном SystemVue. Частота мгновенного отображения может переключаться между двумя режимами: синхронным и асинхронным. В современных вычислительных технологиях скорость анализа почти всегда выше скорости отображения; таким образом, результат анализа почти всегда генерируется быстрее, чем он может быть отображен. "Синхронный" режим относится к отображаемым данным, которые являются смежными с набором, отображаемым последним, и, следовательно, гарантируется точность данных (и, следовательно, термин "синхронный", указывающий, что данные анализа достоверно представлены в окне динамических графиков без каких-либо пробелов); в то время как "асинхронный" режим относится к отображаемым данным, которые не могут быть смежными с набором, отображаемым последним, и поэтому исключается дросселирование скорости генерации данных моделирования. "Асинхронный" режим является режимом по умолчанию, главным образом для того, чтобы позволить анализу продолжаться как можно быстрее, позволяя при этом извлекать и отображать последний результат в зависимости от скорости отображения. В любом случае, вы можете приостановить и перезапустить мгновенное отображение данных в любое время. Конфигурация окна динамических графиков сохраняется как часть операции сохранения рабочей области проекта в файле .adx, расположенном в том же месте, что и файл рабочей области. Конфигурация по умолчанию используется, когда этот файл .adx отсутствует. Как правило, графические атрибуты, функции и измерения, применимые к отдельным графам в окне динамических графиков, такие же, как и у обычного 2D-или 3D-графика.



Errors Summary (D3)

EVM	=	1.006	%rms	2.894	% pk	at sym	661
Mag Err	=	0.9876	%rms	-2.894	% pk	at sym	661
Phase Err	=	0.1726	° rms	-0.7367	° pk	at sym	242
Freq Err	=	-0.0014	Hz				
IQ Offset	=	-69.36	dB20				
Quad Err	=	-0.00124	°				
Gain Imb	=	0.003512	dB20				
Droop	=	1.58e-05	dB20				





Description: EVM Measurement for Complex Signals
 Domain: Timed
 C++ Code Generation Support: NO
 SystemVueEngine Support: YES (ContinuousMode=YES will NOT bring up dynamic graphs)
 Associated Parts: **EVM Part**

Компонент оценок величины вектора ошибок измерений

Эта модель используется для выполнения измерения EVM (Величины вектора ошибки) для комплексного сигнала. Измерения EVM используются для оценки точности модуляторов.

Определяющие уравнения для измерения EVM следуют определению в стандарте совместимости двухрежимной мобильной станции и базовой станции сотовой системы EIA/TIA IS-54-B TDMA, раздел 2.1.3.3.1.3.3 (Требование к величине вектора ошибки). Пусть $Z(n)$ обозначает действительные комплексные векторы (I и Q), полученные наблюдением реального передатчика через идеальный фильтр приемника в моменты времени n с интервалом в один символ. $S(n)$ определяется как идеальный эталонный символ (нормированный таким образом, чтобы его максимальный энергетический символ приходился на

$Z(n) = [C_0 + C_1 \cdot (S(n) + E(n))] \cdot W^n$ как:

$$W = e^{Dr + j \cdot Da}$$

Где учитывает как частоту смещения (Da radians/symbol phase rotation), так и скорость изменения амплитуды (Dr nepers/symbol)

C_0 - является комплексным постоянным смещением начала координат (в вольтах)

C_1 - является безразмерной комплексной константой, представляющей произвольную фазу и выходную мощность передатчика

$E(n)$ - является вектором остаточной ошибки по выборке $S(n)$

Сумма квадратов вектора ошибок равна:

$$\sum_{n=0}^{N-1} |E(n)|^2 = \sum_{n=0}^{N-1} \left| \frac{Z(n) \cdot W^{-n} - C_0}{C_1} - S(n) \right|^2$$

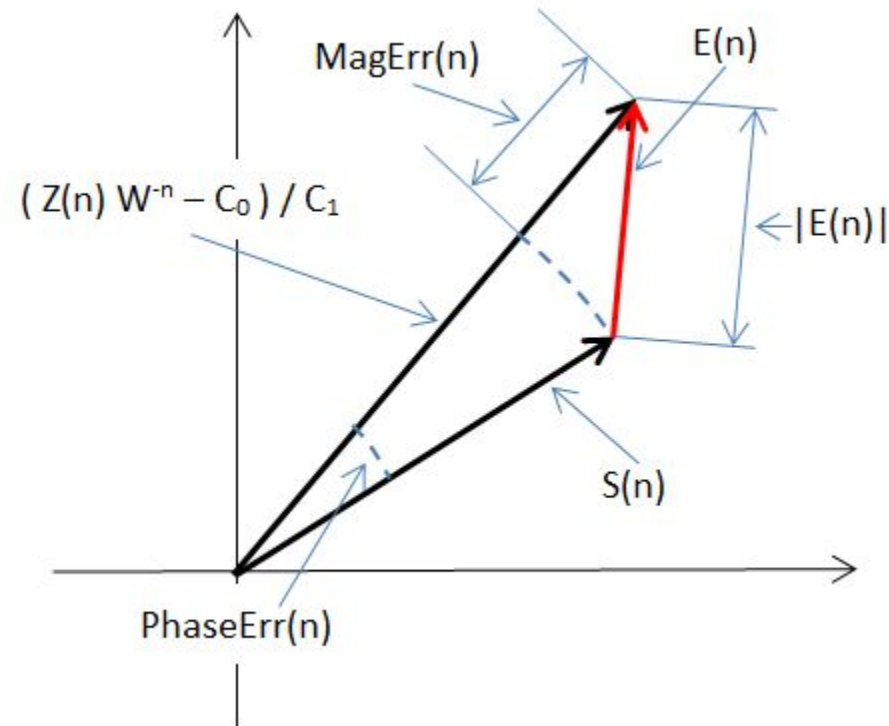
где N равно ResultLength и C0, C1, W выбираются таким образом, чтобы минимизировать приведенное выше выражение. EVM_RMS определяется как среднеквадратичное значение $|E(n)|$, нормализованное к эталонному значению REF (эталонные значения различаются для различных форматов модуляции). Следовательно,

$$EVM_RMS = \sqrt{\frac{1}{N} \cdot \sum_{n=0}^{N-1} |E(n)|^2 \cdot \frac{1}{REF}}$$

EVM определяется как

которая представляет $EVM(n) = \frac{|E(n)|}{REF}$ чину вектора ошибки при символе n, нормализованную к опорному значению REF. На рисунке графически показан вектор ошибки E(n), а также величина вектора ошибки $|E(n)|$, ошибка величины MagErr(n) и фазовая ошибка PhaseErr(n).

Эталонное значение REF, используемое для нормализации ошибок, зависит от значения параметра NormalizationReference. Если NormalizationReference=Constellation Max, REF-это величина самой внешней точки в созвездии (для неквадратных созвездий, таких как 32-QAM, 128-QAM, 512-QAM, опорное значение-это величина самой внешней угловой точки, которая существовала бы, если бы созвездие было квадратным) Reference RMS, REF - среднеквадратичное значение эталонного созвездия. Для созвездий с точками одинаковой величины (BPSK, QPSK, 8-PSK, 16-PSK) нормализация недоступна, так как для этих созвездий величина их самой внешней точки и их среднеквадратичное значение одинаковы.



Параметр Start задает время начала сбора данных. Убедитесь, что вы установили Start, чтобы избежать любых переходных процессов сигнала, которые могут способствовать более высоким, чем ожидалось, уровням EVM.

Для обеспечения возможности установки внутренних фильтров модель будет собирать образцы данных еще до запуска, так что фактическое измерение начнется как можно ближе к началу. Если перед запуском не хватает образцов данных, то Запуск сбрасывается на более высокие значения и отображается предупреждение.

Параметр SymbolRate задает частоту символов, с которой будет выполняться измерение, и должен соответствовать частоте символов входного сигнала.

Параметр ResultLength задает количество символов, по которым будет выполняться измерение.

Параметр ModType выбирает формат модуляции, для которого будет выполняться измерение, и должен соответствовать формату входного сигнала. Для некоторых параметров ModType необходимо установить дополнительные параметры.

Параметр UserDefinedConstellation указывает точки созвездия, когда ModType определяется пользователем.

Параметры Ratio_R2_R1, Ratio_R3_R1 и Ratio_R4_R1 используются для установки отношения второй, третьей и четвертой кольцевых величин к первой кольцевой величине для форматов 16-APSK, 32-APSK, Star 16-QAM и Star 32-QAM.

Ratio_R2_R1 применим для 16-АПСК, 32-АПСК звезды 16-QAM и звезда 32-Кам.

Ratio_R3_R1 применимо к 32-APSK и Star 32-QAM. Ratio_R4_R1 применимы только к Star 32-QAM.

Параметры RingStates, Ringmagnitudes и RingInitialPhase используются для определения пользовательских созвездий APSK, где точки созвездий равномерно распределены вокруг коцентрических колец. Более подробную информацию о созвездиях для поддерживаемых форматов модуляции см. в разделе DigitalMod.

Параметры MeasurementFilter и ReferenceFilter определяют, какая фильтрация будет выполняться на входе и внутри сгенерированных опорных сигналах. Входной сигнал сначала фильтруется с помощью выбранного измерительного фильтра. Затем сигнал на выходе измерительного фильтра понижается в оптимальные моменты дискретизации для создания последовательности символов $Z(n)$. Эта последовательность символов демодулируется для создания идеальной эталонной последовательности символов. Идеальная последовательность опорных символов фильтруется с помощью выбранного фильтра ReferenceFilter. Наконец, сигнал на выходе опорного фильтра понижается в оптимальные моменты дискретизации для создания последовательности опорных символов $S(n)$.

Длина встроенных измерительных и эталонных фильтров составляет 15 символов. Если для любого из фильтров выбран вид характеристики *Root Raised Cosine* или *Raised Cosine*, то параметр Alpha устанавливает их точку среза. Для правильной настройки измерения EVM фильтр формирования импульсов, используемый при генерации сигнала в передатчике TX, каскадируемый измерительным фильтром, должен привести к эталонному фильтру. Измерительный фильтр может быть реализован вне модели EVM (если требуется точное управление параметрами фильтра, например длиной фильтра, точностью фиксированной точки и т. Д.), и в этом случае параметр MeasurementFilter должен быть установлен в значение None. В следующей таблице приведены некоторые примеры правильных настроек (Измерительный фильтр это либо встроенный измерительный фильтр EVM, определенный в параметре

TX Pulse Shaping Filter	Measurement Filter	Reference Filter
Rectangular	None	Rectangular
Root Raised Cosine	None	Root Raised Cosine
Root Raised Cosine	Root Raised Cosine	Raised Cosine
Raised Cosine	None	Raised Cosine

Модель EVM работает с сигналами, которые представляют собой ОДНУ модулированную несущую. Когда входной сигнал включает в себя несколько модулированных несущих, пользователь ДОЛЖЕН отфильтровать нежелательные несущие с помощью внешнего фильтра перед отправкой сигнала в модель EVM. В зависимости от расстояния между модулированными несущими отклик внешнего фильтра может вносить дополнительные искажения, которые увеличат EVM. В этом случае рекомендуется настроить внешний фильтр таким же образом, как и MeasurementFilter, и установить параметр MeasurementFilter модели EVM равным None. Таким образом, внешний фильтр будет действовать как фильтр IF, который будет отфильтровывать нежелательные модулированные несущие, также как и измерительный фильтр. Конечно, если бы изначально был фильтр MeasurementFilter, установленный на None, нужно было бы настроить внешний фильтр по вашему выбору, чтобы отфильтровать нежелательные модулированные носители.

Результаты измерений, сохраненные в наборе данных, описаны ниже.

EVM_RMS: Значение EVM_RMS, выражается в %.

EVM_Peak: максимальное значение символа EVM, $\max \{EVM(n)\} n = 0, 1, \dots, N-1$, выраженное в %.

EVM_PeakSymbolIndex: индекс символа, в котором произошел выброс EVM_Peak.

MagErrorRMS: Среднеквадратичное значение ошибки величины символа (MagErr(n)), нормализованное по опорному уровню REF и выраженное в %.

MagErrorPeak: максимальное значение символа с выбросом данных (превышением пикового уровня) MagErr, $\max \{MagErr(n)\} n = 0, 1, \dots, N-1$, выраженное в %.

MagErrorPeakSymbolIndex: индекс символа, в котором произошел выброс MagErrorPeak.

PhaseErrorRMS: среднеквадратичное значение ошибки фазы символа (PhaseErr(n)).

PhaseErrorPeak: максимальное значение символа PhaseErr, $\max \{PhaseErr(n)\} n = 0, 1, \dots, N-1$.

PhaseErrorPeakSymbolIndex: индекс символа, в котором произошел PhaseErrorPeak.

FreqError: частотная ошибка, $(SymbolRate * Da / (2 * \pi))$.

IQ_Offset: смещение IQ, значение постоянного уровня сигнала несущей; $|C0|$ нормализовано REF и выражено в дБ; $20 * \log_{10}(|C0|/REF)$.

Quad Err: квадратурная ошибка перекоса (также называемая фазовым дисбалансом), мера ортогональности сигналов I и Q, выраженная в градусах.

GainImbalance: дисбаланс усиления, мера разницы в усилении сигналов I и Q, выраженная в дБ.

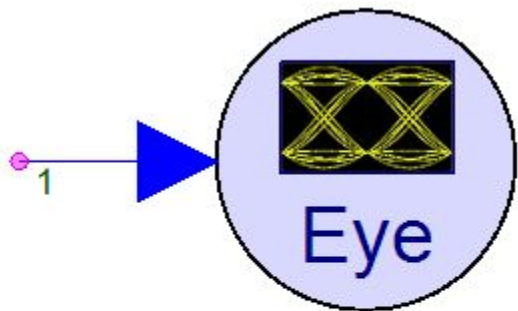
Droop: выражается в дБ; $-20 * \log_{10}(eDr)$.

SNR: отношение сигнал / шум выражается в децибелах.

Если параметр ContinuousMode установлен в значение ДА, то графики будут динамически обновляться на протяжении всего моделирования.

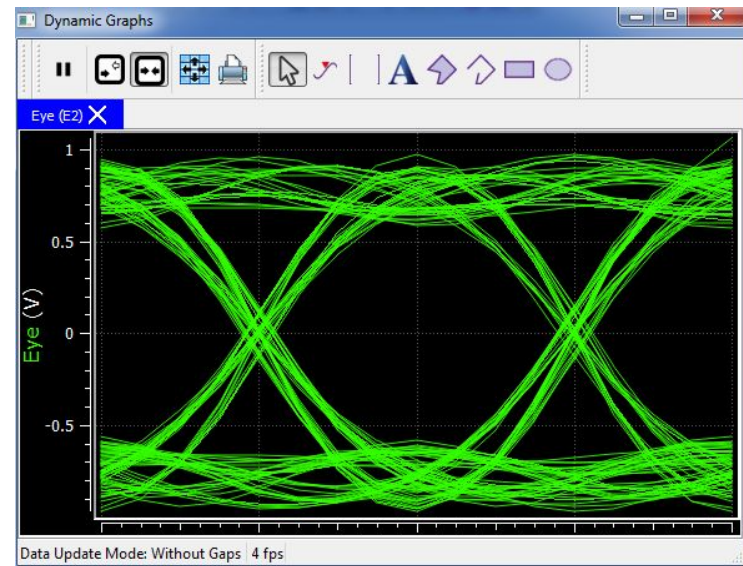


Глазковая диаграмма



Description: Dynamic Eye Diagram
Domain: Timed
C++ Code Generation Support: NO
SystemVueEngine Support: NO
Associated Parts: EyeDiagram Part

Глазковая диаграмма представляет собой "осциллограмму" аналогового демодулированного сигнала, построенную при длительности "прямого хода развертки", равной одному символному такту, и бесконечном "времени послесвечения экрана". В точках оптимальной дискретизации линии на такой диаграмме образуют узкие пучки, свободное пространство между которыми по форме напоминает раскрытый глаз.



Эта модель создает динамический график, отображающий глазковую диаграмму входного сигнала.

Параметр `SymbolRate` определяет частоту символов входного сигнала. Она используется вместе с частотой дискретизации входного сигнала, чтобы определить, сколько выборок на символ имеет входной сигнал. Пусть это число равно N . N должно быть точным целым числом, чтобы эта модель работала правильно. В противном случае диаграмма будет непрерывно смещаться вдоль нижней оси графика.

Параметр `Offset` можно использовать для смещения диаграммы вдоль нижней оси графика так, чтобы вы могли центрировать ее в середине графика. Этот параметр настраивается во время выполнения (настраивается во время моделирования), поэтому нет необходимости знать оптимальное значение до начала моделирования. Допустимый диапазон смещения - $[0, N-1]$. Если во время настройки присваивается значение, выходящее за пределы этого диапазона, то значение обрезается до пределов этого диапазона. В общей сложности символы результирующей длины показаны на диаграмме. Каждый след имеет длину 2 символа, поэтому общее количество следов равно $N / 2$. Эта модель НЕ делает никакой демодуляции. Входной сигнал уже должен быть демодулирован с целым числом отсчетов на символ. Эта модель НЕ делает никакой интерполяции. Поэтому даже для идеального или близкого к идеальному сигнала, если входной сигнал не включает выборку в оптимальный момент выборки (например, из-за частичных задержек выборки, вносимых фильтрами), вы не сможете получить "чистую" диаграмму с "идеальным" открытием.

Основы глазковой диаграммы: чтение и применение глазковых диаграмм

Увеличение скоростей передачи данных, бóльшая сложность проектирования, требования стандартов и сокращение временных периодов предъявляют повышенные требования к инженерам-проектировщикам для скорейшего устранения сложных проблем целостности сигналов. Поскольку сегодняшние последовательные каналы передачи данных работают на гигагерцовых частотах передачи, на целостность сигналов может повлиять множество переменных, включая влияние линии передачи, рассогласование импедансов, разводку сигналов, оконечные схемы и схемы земли. Используя осциллограф для создания глазковой диаграммы, инженеры могут быстро оценить производительность системы и понять природу дефектов канала связи, которые могут привести к ошибкам, когда приемник пытается интерпретировать значение бита.

При передаче от передатчика к приемнику последовательный цифровой сигнал может ухудшаться. Передатчик, дорожки печатной платы, разъемы и кабели будут создавать помехи, которые ухудшают сигнал, как по амплитуде, так и по времени. Сигнал также может пострадать от внутренних источников. Например, когда сигналы на соседних парах трасс печатной платы или на выводах микросхемы переключаются, перекрестные связи между этими сигналами могут создавать помехи другим сигналам. Таким образом, вам нужно определить, в какую точку поместить щуп осциллографа, чтобы получить глазковую диаграмму, которая поможет вам определить источник проблемы. Кроме того, в зависимости от того, куда вы поместите щуп осциллографа, сигналы на экране будут различаться.

Получение глазковой диаграммы

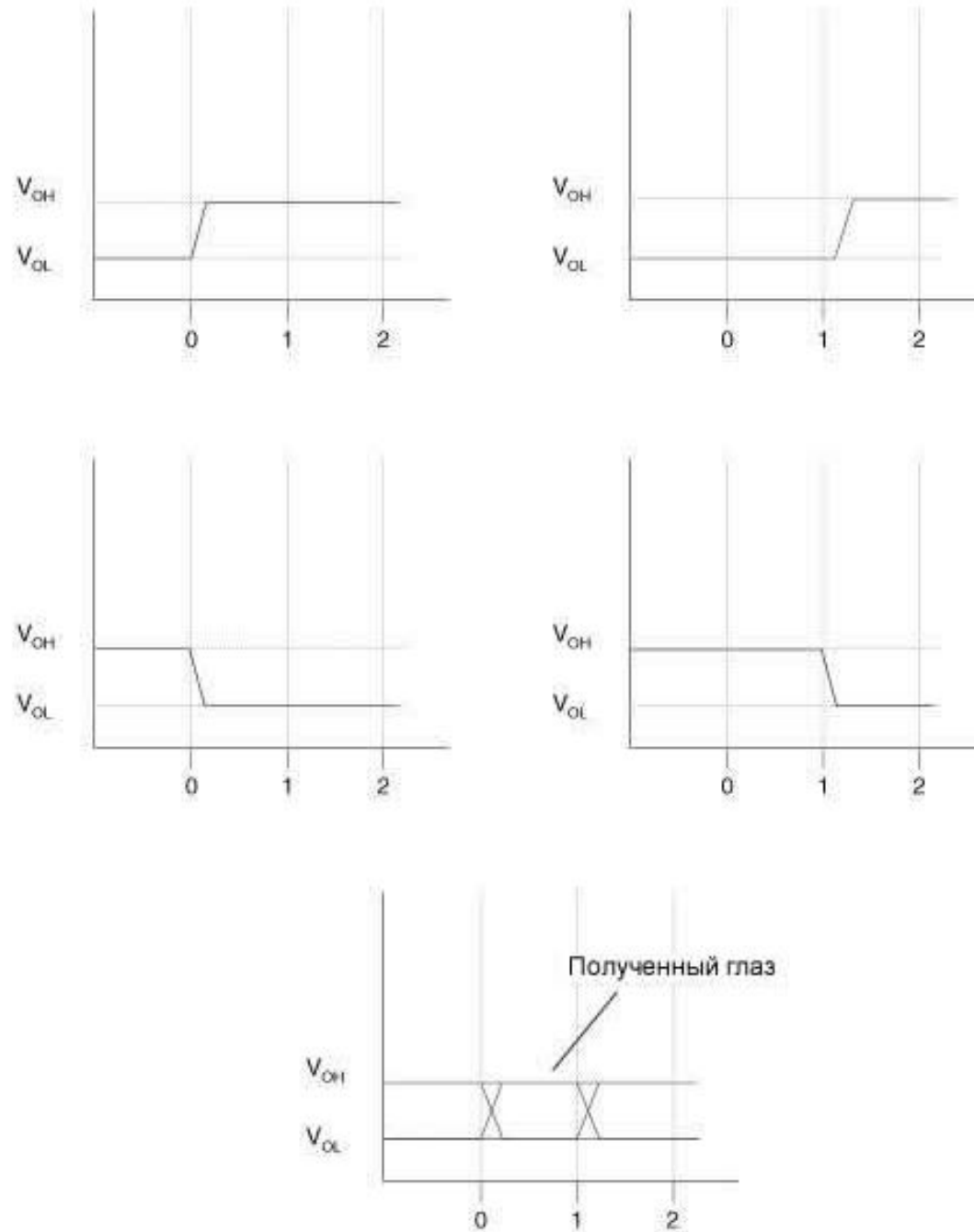
Глазковая диаграмма – это распространенный индикатор качества сигналов в высокоскоростных цифровых системах передачи. Осциллограф формирует глазковую диаграмму, накладывая развертки различных сегментов длинного потока данных, управляемые главным источником синхронизации. Фронт запуска может быть положительным или отрицательным, но отображаемый импульс, который появляется после периода задержки, может идти в любом направлении; нет способов заранее узнать значение произвольного бита. Следовательно, когда наложено много таких переходов, положительные и отрицательные импульсы накладываются друг на друга. Наложение множества битов дает глазковую диаграмму, называемую так, потому что получающееся изображение выглядит как открывание глаза.

В идеальном мире глазковые диаграммы выглядят как прямоугольные коробки. В реальности, каналы связи не идеальны, поэтому переходы не идеально выровнены друг с другом, и в результате получается рисунок в форме глаза. Форма глазковой диаграммы на осциллографе будет зависеть от различных типов сигналов запуска, таких как сигнал запуска с тактовой частотой потока, сигнал запуска с деленной тактовой частотой потока, запуск по кодовой комбинации (pattern trigger). Различия по времени и амплитуде от бита к биту приводят к уменьшению открытия глаза.

Интерпретация глазковой диаграммы

Правильно построенный глаз должен содержать каждую возможную битовую последовательность от простых чередующихся единиц и нулей до изолированных единиц после продолжительных последовательностей нулей и всех других шаблонов, которые могут обнаружить недостатки в конструкции. Глазковые диаграммы обычно включают в себя выборки напряжения и времени для данных, полученные с некоторой частотой дискретизации ниже скорости передачи данных. На рисунке битовые последовательности 011, 001, 100 и 110 накладываются друг на друга, чтобы получить окончательную глазковую диаграмму.

Идеальная глазковая диаграмма содержит огромное количество параметрической информации о сигнале, подобно эффектам, вытекающим из физики, независимо от того, как эти эффекты возникают. Если логическая 1 настолько искажена, что приемник на дальнем конце может неверно принять ее за логический 0, вы легко заметите это по глазковой диаграмме. Однако по ней вы не сможете обнаружить проблемы логики или протокола, например, когда система должна передавать логический 0, но отправляет логическую 1, или когда логика конфликтует с протоколом.

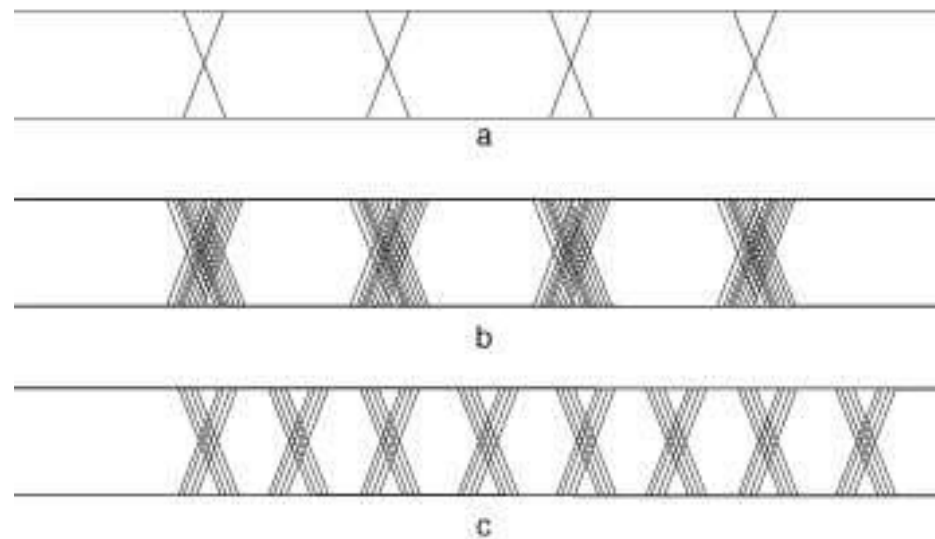


Джиттер

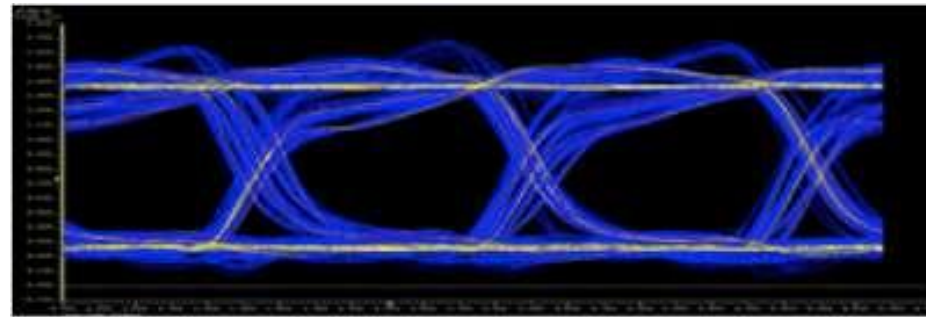
Хотя теоретически глазковые диаграммы должны выглядеть как прямоугольные коробки, конечное время нарастания и спада сигналов и осциллографов приводит к тому, что глазковые диаграммы на самом деле больше похожи на изображение на рисунке *a*. Когда передаются высокоскоростные цифровые сигналы, искажения, вносимые на различных этапах, приводят к ошибкам синхронизации. Одной из таких временных ошибок является «джиттер» («jitter», фазовое дрожание), который возникает из-за несовпадения времени нарастания и спада.

Джиттер возникает, когда нарастающие и спадающие фронты происходят в моменты, отличные от идеального времени. Некоторые фронты появляются рано, некоторые – поздно. В цифровой схеме все сигналы передаются относительно тактовых сигналов. Отклонение цифровых сигналов в результате отражений, межсимвольных помех, перекрестных помех, влияний изменений напряжения и температуры и других факторов приводит к джиттеру. Некоторый джиттер будет просто случайным.

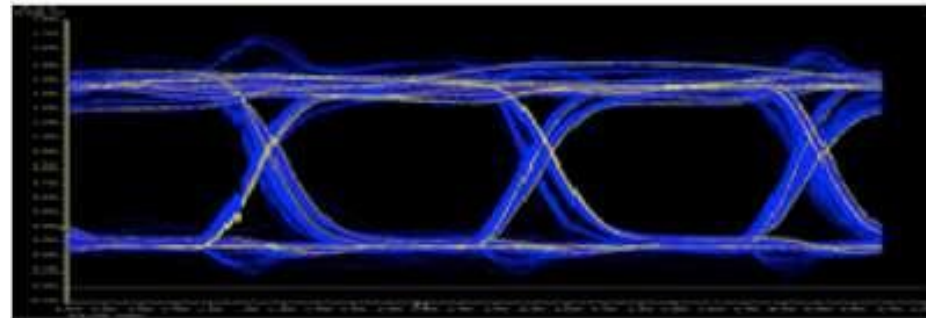
На рисунке *c* абсолютная ошибка по времени или поле фазового дрожания меньше, чем на рисунке *b*, но открытие глаза на рисунке *c* меньше из-за более высокой скорости передачи данных. С увеличением скорости передачи данных абсолютная ошибка по времени представляет собой увеличивающуюся часть периода сигнала, таким образом, уменьшая размер открытия глаза. Это может увеличить вероятность ошибок в данных.



На сгенерированных глазковых диаграммах хорошо виден эффект разрыва. При неправильной нагрузке конца линии глаз выглядит ограниченным или «напряженным» (сдавленным) (рисунок a), а с улучшенными схемами нагрузки концов линии глаз становится «расслабленным» (рисунок b). Неправильно нагруженная линия страдает от множества отражений. Отраженные волны имеют значительную амплитуду, что может сильно сузить глаз. Как правило, это наихудший режим работы приемника, и если приемник может работать без ошибок при наличии таких помех, то он соответствует требованиям спецификации.



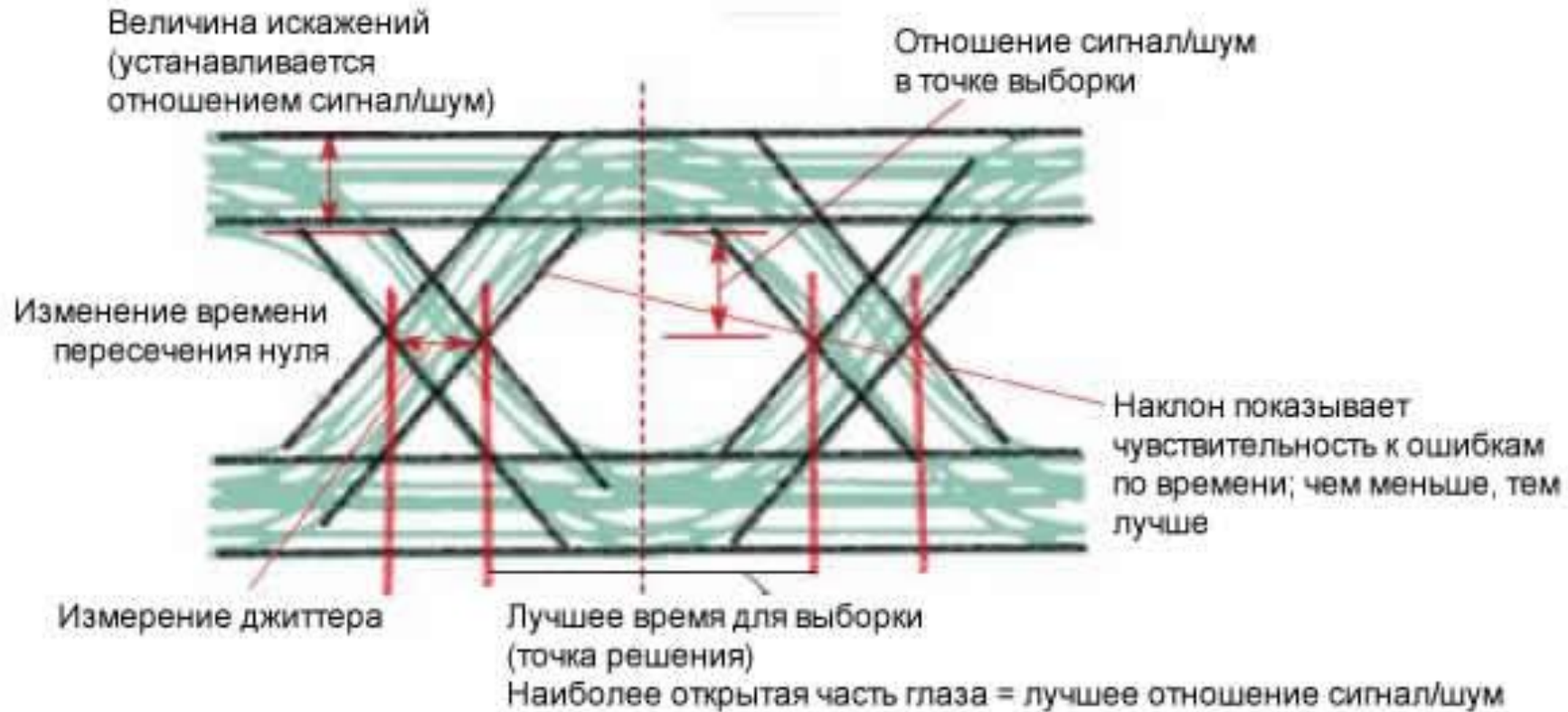
(a)



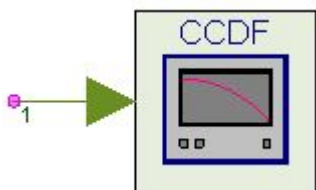
(b)

- (a) Неправильная нагрузка заставляет глазковую диаграмму выглядеть «напряженной» (сдавленной).
(b) Правильная нагрузка «расслабляет» глаз.

Как видно на рисунке, глазковая диаграмма может показать важную информацию. Она может указывать лучшую точку для выборки, показывать SNR (отношение сигнал/шум) в точке выборки и указывать величины джиттера (фазового дрожания) и искажений. Кроме того, она может показать изменения по времени при пересечении нуля, что является измерением джиттера. Глазковые диаграммы предоставляют мгновенные визуальные данные, которые инженеры могут использовать для проверки целостности сигнала в проекте и для выявления проблем на ранних стадиях процесса проектирования. Используемая в сочетании с другими измерениями, такими как частота битовых ошибок, глазковая диаграмма может помочь разработчику предсказать производительность и определить возможные источники проблем.



Глазковая диаграмма может помочь вам интерпретировать сигнал и определить наилучшее время для измерений.



CCDF_Cx

Модель CCDF_Cx вычисляет комплементарную кумулятивную функцию распределения (CCDF) сложного сигнала.

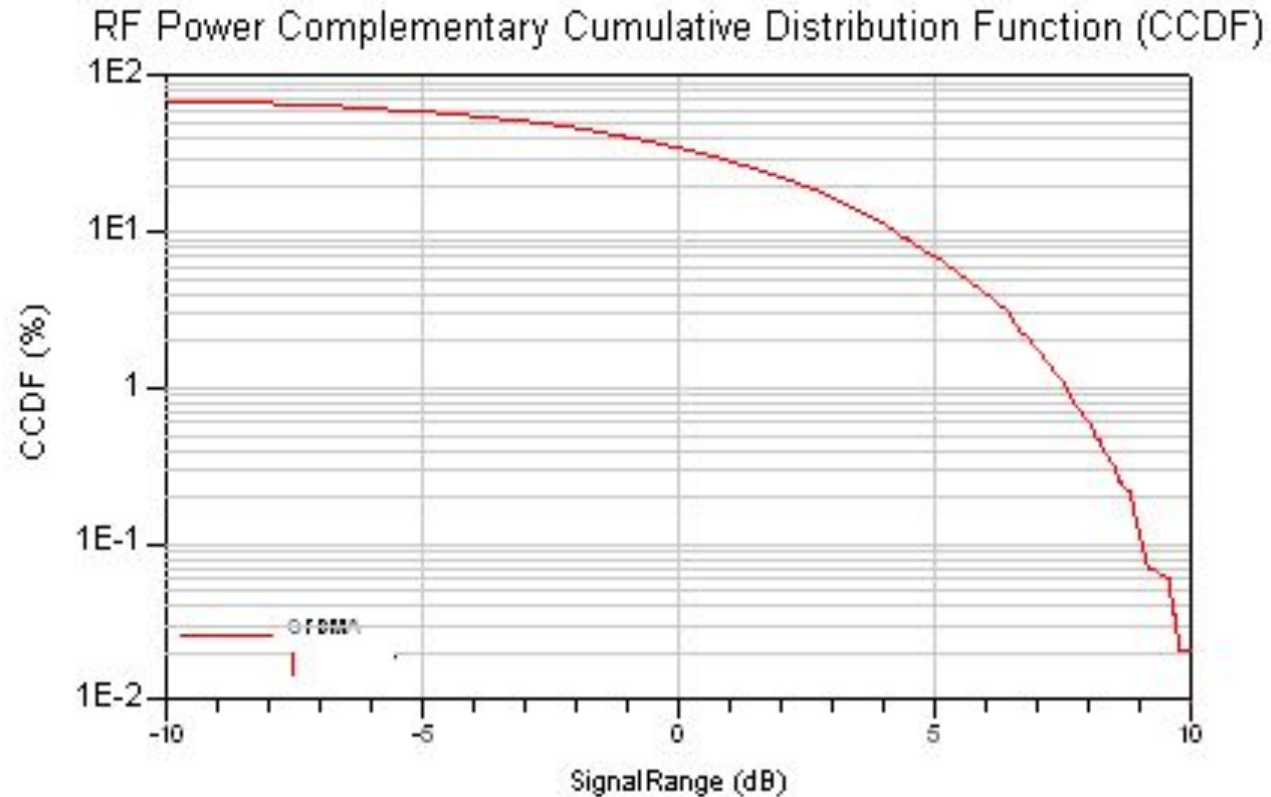
Расчет CCDF описан ниже:

- Вычислите среднеквадратичное значение для всех измеренных выборок; оно становится точкой 0 дБ на оси x.
- Нормализуйте все выборки до среднеквадратичного значения в единицах дБ.
- Разделите ось x на равные по ширине ячейки NumBin, начиная с минимальной измеренной мощности и заканчивая максимальной измеренной мощностью.
- Определите, к какой ячейке по оси x принадлежит каждый отсчет.
- Вычислите общее количество выборок, и выведите его в процентах от числа измеренных выборок.

В дополнение к измерению CCDF эта модель может обеспечить пиковую мощность (пиковую мощность входного сигнала; это НЕ абсолютный пик, а 99,9-й перцентиль) и среднюю мощность (среднюю или среднюю мощность входного сигнала). Чтобы вычислить и вывести MeanPower и PeakPower, установите для параметра OutputPeakMean значение YES.

Измерение CCDF-это очень распространенное измерение, выполняемое на беспроводных сигналах 2G, 3G и 4G. Кривая CCDF показывает вероятность того, что мгновенная мощность сигнала будет выше средней мощности сигнала на определенную величину дБ. Независимая ось кривой CCDF показывает уровни мощности в дБ по отношению к среднему уровню мощности сигнала (0 дБ соответствует среднему уровню мощности сигнала). Зависимая ось кривой CCDF показывает вероятность того, что мгновенная мощность сигнала превысит соответствующий уровень мощности на независимой оси. На следующем рисунке показана кривая CCDF для нисходящего сигнала WiMAX 802.16 e. На рисунке видно, что мгновенная мощность сигнала превышает среднюю мощность сигнала (0 дБ) на 35% времени. Вы также можете видеть, что мгновенная мощность сигнала превышает среднюю мощность сигнала на 5 дБ только в 7% случаев.

CCDF для WiMax 802.16e Downlink Signal



В большинстве систем беспроводной связи с кадровыми или пакетными данными измерение CCDF должно выполняться на активной части сигнала (нулевые части между кадрами/пакетами должны быть исключены).