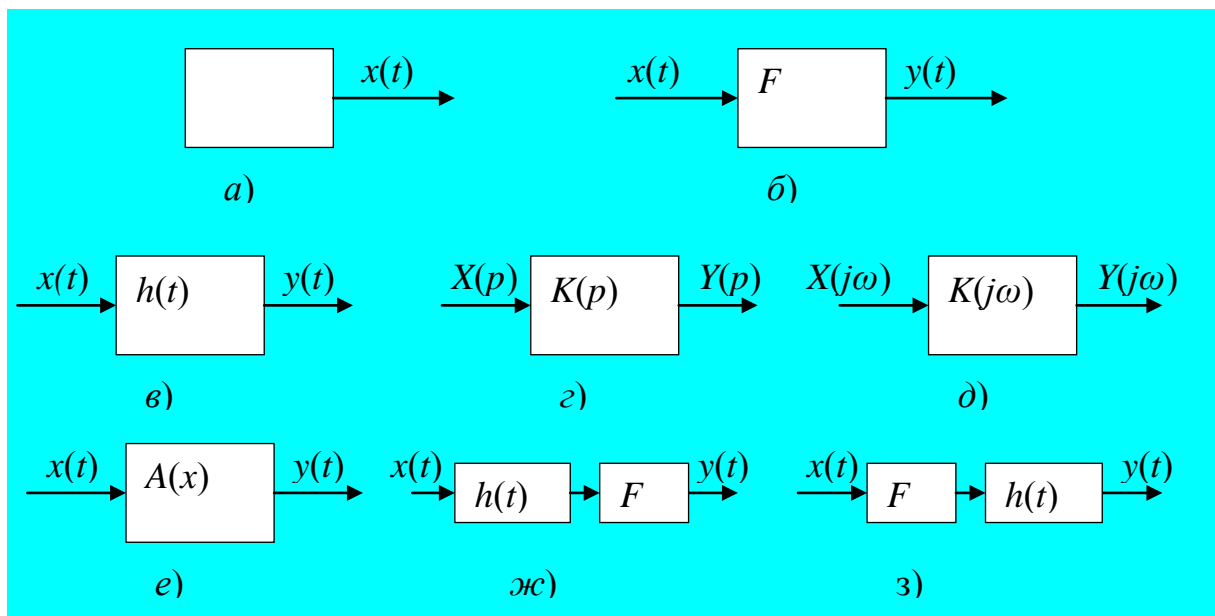


## 4. МАТЕМАТИЧЕСКИЕ МОДЕЛИ БАЗОВЫХ БЛОКОВ НА УРОВНЕ АФЛП

### 4.1. Базовые блоки функциональных схем

Несмотря на многообразие таких блоков (элементов), их можно свести к четырём основным типам [21], которые в дальнейшем будем называть базовыми: генераторы сигналов, линейные и нелинейные безынерционные блоки, линейные инерционные блоки и нелинейные инерционные блоки (рис. 4.1). Здесь обозначены:  $F$  – линейный или нелинейный оператор,  $h(t)$  – импульсная характеристика,  $K(p)$  – операторный коэффициент передачи,  $K(j\omega)$  – комплексный коэффициент передачи.

Основная характеристика элемента при функциональном моделировании – его функция преобразования, которая связывает входной и выходной сигналы. Рассмотрим вид этих функций для перечисленных выше четырех типов базовых блоков (элементов).



**Рис. 4.1. Базовые блоки (элементы) функциональных схем:**

$a$  – генератор сигнала;  $б$  - безынерционный блок (линейный и нелинейный);  
 $в, г, д$  - линейные инерционные блоки;  $ж, з$  – эквивалентное представление нелинейного инерционного блока

**Генераторы (или источники) сигналов.** Этот тип блоков включает две разновидности: независимые генераторы, задающие сигнал  $x(t)$  на входе конкретной функциональной схемы, и управляемые генераторы, формирующие ту или иную форму сигнала  $x(t)$  в зависимости от управляющего воздействия  $E$ .

Типичные примеры независимых генераторов: генератор гармонического сигнала, генератор импульсного сигнала, генератор частотно-модулированного сигнала, генератор широкополосного случайного процесса, генератор псевдослучайной импульсной последовательности.

Для управляемого генератора сигналов функция преобразования имеет следующий вид:

$$x(t) = \begin{cases} x_1(t) & \text{при } E = E_1, \\ \dots & \dots \\ x_n(t) & \text{при } E = E_n. \end{cases}$$

Типичные примеры управляемого генератора: генератор сигналов управляемый напряжением (ГУН), который применяется в системах ФАПЧ, а также различные пороговые устройства, выходной сигнал которых зависит от превышения уровнем входного сигнала некоторого порогового значения.

**Безынерционный линейный или нелинейный блок.** Функция преобразования в этом случае представляет собой линейную или нелинейную функцию  $F$ , которая устанавливает связь между входным сигналом  $x$  и выходным сигналом  $y$ :

$$y = F(x).$$

Примеры безынерционных линейных блоков: масштабные усилители (с инверсией и без инверсии), сумматоры и перемножители.

Нелинейный безынерционный блок позволяет преобразовать форму входного сигнала  $x(t)$  в любую другую форму выходного сигнала

$y(t) = F(x(t))$ . Например, ограничивать амплитуду  $x(t)$ , формировать из импульса  $x(t)$  импульс заданной формы  $y(t)$ , выполнять достаточно сложные нелинейные преобразования сигнала, как то: взятие модуля, квантование, возведение в степень, извлечение корня квадратного и др.

К числу безынерционных линейных блоков относятся также устройства, реализующие логические функции, и триггер, который является безынерционным элементом с памятью. Функция преобразования последнего неоднозначна по отношению к входному сигналу  $x$  и зависит не только от  $x$ , но и от состояния элемента  $s$ :

$$y = F(x, s).$$

**Линейный инерционный блок.** Его функция преобразования во временной области – это импульсная характеристика  $h(t)$ , а в частотной области – коэффициент передачи  $K(j\omega)$  (или  $K(p)$ ). В первом случае

$$y(t) = \int_0^t x(\tau)h(t - \tau)d\tau, \quad (4.1)$$

а во втором

$$Y(j\omega) = K(j\omega) \cdot X(j\omega),$$

или

$$Y(p) = K(p) \cdot X(p)$$

К инерционным линейным блокам относится весьма широкий класс устройств. Например, различные типы частотных фильтров, устройства временной задержки, линейные широкополосные и полосовые усилители, операционные усилители, выполняющие функции интегрирования, дифференцирования и др.

**Инерционный нелинейный блок.** Его функция преобразования – некоторый нелинейный оператор  $A(x)$ . Например, нелинейное дифференциальное уравнение, которое ставит в соответствие каждой

временной реализации  $x(t)$  реализацию  $y(t)$ . Подчеркнем, что дифференциальные уравнения являются наиболее общей и универсальной формой представления моделей инерционных элементов любого типа – линейных и нелинейных.

Такая форма представления удобна для автоматизированного функционального проектирования, так как позволяет использовать аппарат численных методов, применяемый для функционального моделирования. В большинстве случаев реальный нелинейный инерционный блок можно описать системой дифференциальных уравнений такого вида:

$$\frac{dy^i(t)}{dt^i} + q_i(y(t)) = \frac{dx^i(t)}{dt^i} + f_i(x(t)) \quad , \quad (4.2)$$

где  $x(t)$ ,  $y(t)$  – входной и выходной сигналы соответственно;  $i = 1, \dots, n$ ;  $n$  – порядок системы.

В тех случаях, когда составление модели инерционного нелинейного блока в форме дифференциальных уравнений затруднительно, этот блок приближенно моделируют последовательным соединением инерционного линейного элемента и безынерционного нелинейного элемента (см. рис. 4.1, ж, з), так что соответственно имеем:

$$y(t) = F\left(\int_0^t x(\tau)h(t-\tau)d\tau\right),$$

$$\text{или } y(t) = \int_0^t F(x(\tau))h(t-\tau)d\tau.$$

Отметим, что выходной сигнал существенно зависит от последовательности включения нелинейного и инерционного элементов. Действительно, схема на рис. 4.1, ж имеет большую вычислительную погрешность, чем схема на рис.4.1, з, так как сигнал на выходе инерционного

элемента включает ошибку численного расчета интеграла, которая затем усиливается на нелинейном элементе.

Примеры таких блоков: смесители, модуляторы, демодуляторы сигналов, в которых нелинейные компоненты являются нелинейными инерционными элементами.

#### 4.2. Алгоритмы моделирования отдельных базовых блоков

Под моделированием отдельного блока понимается вычисление его выходной величины  $y$  по заданному значению  $x$  с помощью численных методов.

**Генераторы (или источники) сигналов.** Моделирование генератора сигнала  $x(t)$  заключается в вычислении заданной функции  $x(t)$  в известные моменты времени  $t_n$ . В результате непрерывная функция  $x(t)$  заменяется дискретной решетчатой функцией  $x_n = x(t_n)$ . Часто полученную сложную функцию для экономии временных затрат на численный расчет записывают в виде более простой рекуррентной формулы

$$x_n = f(x_{n-1}, x_{n-2}, \dots).$$

Рассмотрим пример. Пусть нужно смоделировать сигнал  $x(t) = k \cdot \exp(\alpha \cdot t)$ . Для моделирования используем следующее соотношение:

$$\begin{aligned} x_n = x(t_n) &= k \cdot \exp(\alpha \cdot t_n) = k \exp(\alpha \cdot (t_{n-1} + \Delta t)) = \\ &= \exp(\alpha \cdot \Delta t) \cdot k \cdot \exp(\alpha \cdot t_{n-1}) = \exp(\alpha \cdot \Delta t) \cdot x_{n-1}. \end{aligned}$$

Если сигнал моделируется через равные промежутки времени  $\Delta t$ , то, вычислив один раз  $C_0 = \exp(\alpha \cdot \Delta t)$  все последующие значения  $x$  можно рассчитывать по простой рекуррентной формуле

$$x_n = C_0 \cdot x_{n-1}, \quad x_0 = k,$$

избегая вычисления в каждой точке нового значения экспоненты.

**Безынерционные линейные и нелинейные блоки.** Здесь мы ограничимся рассмотрением статических безынерционных блоков, когда

функция преобразования  $y = F(x)$  может быть вычислена в любой момент времени. Примеры таких блоков: масштабные усилители, усилитель-компрессор, усилитель-экспандер, квадратор, умножители частоты, в которых усилительные приборы являются нелинейными безынерционными элементами. Тогда моделирование блоков без памяти сводится к вычислению функции  $y = F(x)$  для любого заданного значения  $x$  (например, в табличной форме), а моделирование блока с памятью – к вычислению одной из функций

$$y = \begin{cases} F_1(x) & \text{при } S = S_1, \\ F_2(x) & \text{при } S = S_2. \end{cases}$$

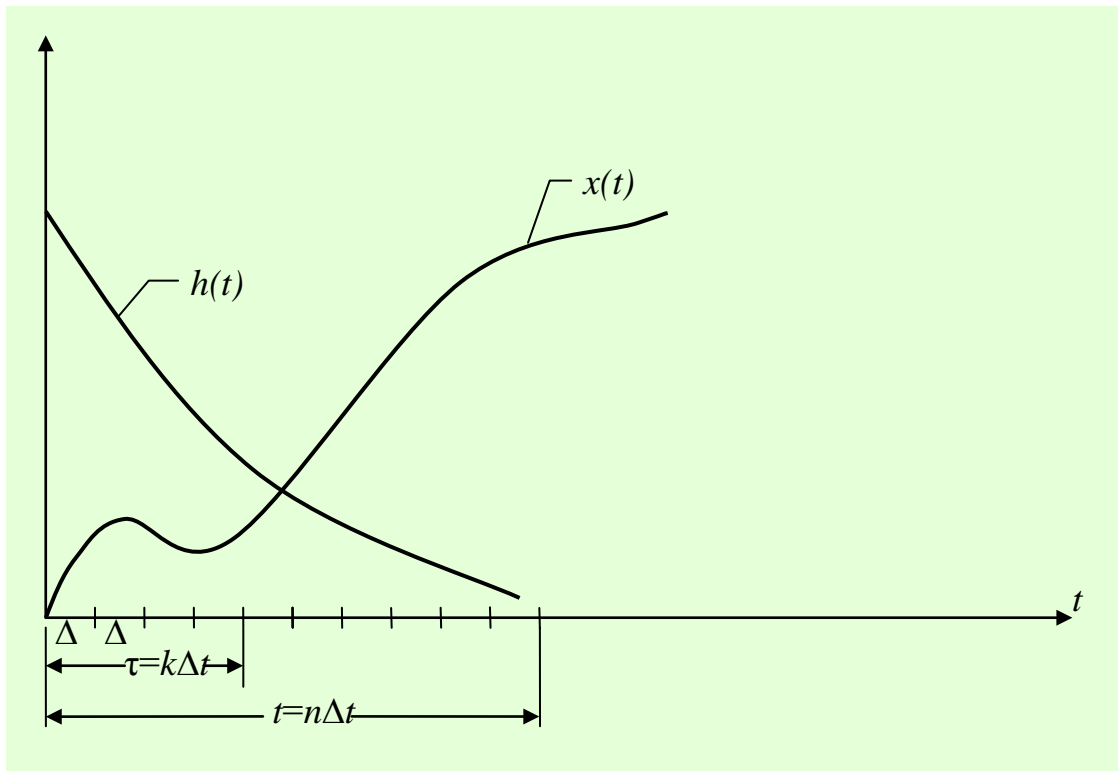
Например, для элемента с памятью – триггера функция вычисляется либо когда триггер находится в состоянии  $S = 1$ , либо когда триггер находится в состоянии  $S = 0$ .

**Линейный инерционный базовый блок.** Способы моделирования этих блоков зависят от способа задания характеристики преобразования: переходная характеристика  $g(t)$ , импульсная характеристика  $h(t)$ , комплексный коэффициент передачи  $K(j\omega)$  или операторный коэффициент передачи  $K(p)$ , – так и от области, в которой выполняется моделирование (временной или частотной), т.е. от вида входного сигнала  $x(t)$  или  $X(j\omega)$ .

В качестве примера рассмотрим моделирование блоков, заданных импульсной характеристикой  $h(t)$ , во временной области. Часто численное интегрирование выражения (4.1) выполняется методом скользящего суммирования. Суть данного метода заключается в следующем (рис. 4.2).

Разбив заданный интервал интегрирования  $[0, t]$  и интервал текущего времени интегрирования  $[0, \tau]$  на равные отрезки  $\Delta t$ , получим

$$t = n \cdot \Delta t, \quad \tau = k \cdot \Delta t, \quad t - \tau = (n - k) \cdot \Delta t.$$



**Рис. 4.2.** К моделированию инерционного линейного блока во временной области

Полагая на каждом из отрезков  $\Delta t$  значения  $x(t)$  и  $h(t - \tau)$  постоянными, что соответствует численному интегрированию (4.1) методом прямоугольников, находим при  $d\tau \approx \Delta t$  вместо интеграла (4.1) его дискретную аппроксимацию в виде суммы

$$y_n = \sum_{k=0}^n x(k\Delta t) \cdot h[(n-k)\Delta t] \cdot \Delta t = \Delta t \sum_{k=0}^n x_k \cdot h_{n-k} \quad (4.3)$$

Таким образом, вычисление значений выходного сигнала  $y_{n-1}, y_n, y_{n+1}, \dots$  в каждый момент времени  $t_{n-1}, t_n, t_{n+1}, \dots$  сводится к вычислению сумм

$$\begin{aligned} y_{n-1} &= \Delta t(x_0 h_{n-1} + x_1 h_{n-2} + \dots + x_{n-1} h_0), \\ y_n &= \Delta t(x_0 h_n + x_1 h_{n-1} + \dots + x_n h_0), \\ y_{n+1} &= \Delta t(x_0 h_{n+1} + x_1 h_n + \dots + x_{n+1} h_0) \end{aligned}$$

с помощью метода скользящего суммирования значений  $x_k$  со скользящим весом  $h_{n-k}$ .

Соотношение (4.3) называют дискретной сверткой интеграла (4.1). Очевидно, применяя вместо прямоугольников другие виды аппроксимации выражений  $x(t)$ ,  $h(t)$  на интервале  $\Delta t$ , можно получить более точные формулы дискретной свертки, соответствующие вычислению интеграла методом трапеций, парабол (Симпсона) и др.

Другие способы моделирования линейных инерционных базовых блоков можно найти в [21].

**Нелинейные инерционные базовые блоки.** Моделирование блоков этого типа возможно двумя способами. Первый способ состоит в описании блока нелинейным дифференциальным уравнением вида (4.2), после чего это уравнение можно решить численными методами (явными и неявными), которые будут рассмотрены нами в подразделе 7.4.

Однако этот способ может потребовать больших временных затрат, так как он связан с решением уравнений. Более эффективным является второй способ, основанный на представлении нелинейного инерционного блока упрощенной функцией нелинейного преобразования (или упрощенной функциональной схемой замещения (см. рис. 4.1, ж,з).

Ниже рассмотрены особенности реализации этого второго способа на примере моделирования нелинейных преобразований в радиочастотных устройствах: модуляторах, преобразователях частоты, амплитудных ограничителях, детекторов (демодуляторов) и др. При этом возможны два подхода: идеальное и реальное моделирование.

**Идеальное моделирование** не учитывает особенности реальных характеристик преобразующих устройств, например, нелинейность характеристик в смесителях частоты, зависимости свойств детектора от амплитуды входного сигнала и т.д. Вследствие этого все побочные эффекты, которые сопровождают реальное преобразование (появление комбинационных продуктов в преобразователях частоты, нелинейные искажения огибающей в АМ-детекторах и др.) не учитываются. Таким



образом, идеальное моделирование преобразующего нелинейного элемента сводится к выполнению им заданной математической операции над входным сигналом, результат которой рассматривается как выходной сигнал.

Поясним это на примере моделирования идеального амплитудного детектора (рис. 4.3). Пусть на вход формирователя квадратурных компонент (ФКК) поступает высокочастотное амплитудно-модулированное колебание  $S_x(t) = U(t)\sin(\omega_0 t + \varphi)$  с известным законом модуляции  $U(t)$ .

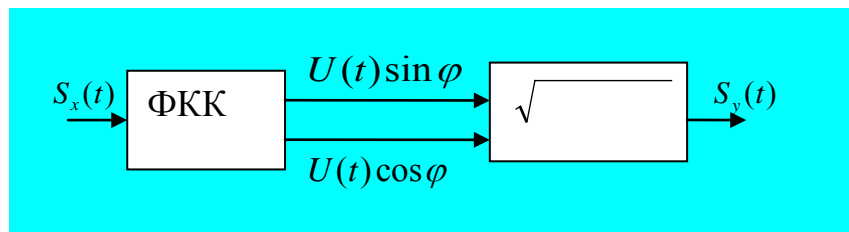


Рис. 4.3. Схема идеализированного функционального детектирования

Здесь ФКК выполняется по структурной схеме, изображенной на рис.4.4.

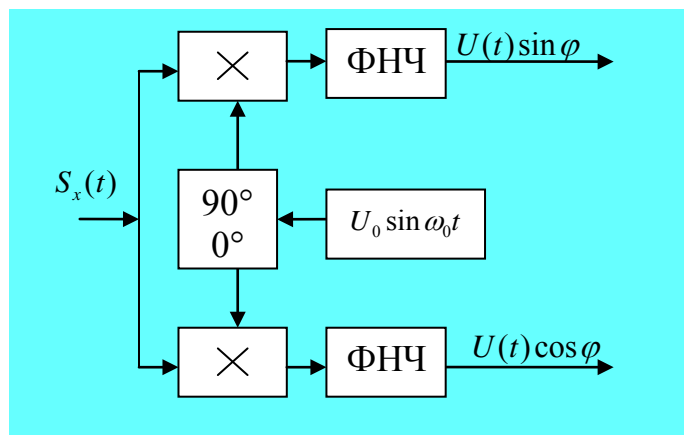


Рис. 4.4 Вариант построения ФКК

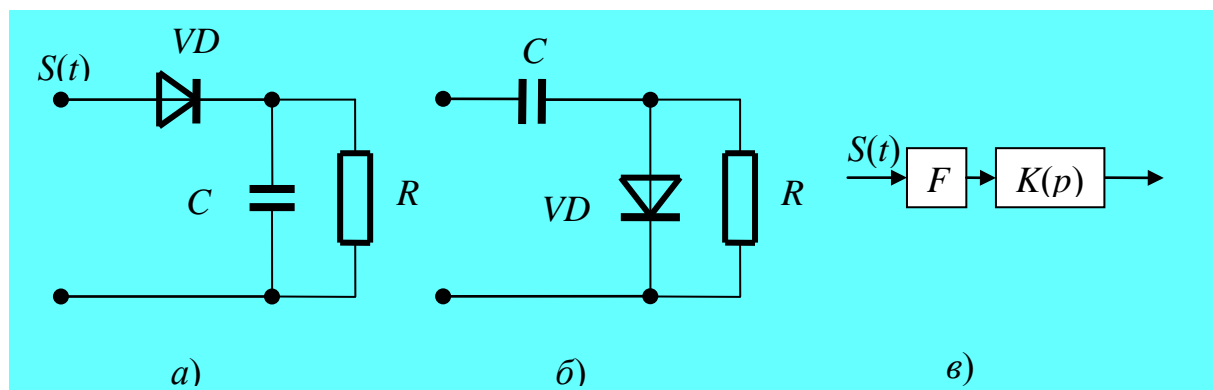
Квадратурные компоненты  $U(t)\sin \varphi$  и  $U(t)\cos \varphi$  поступают на устройство извлечения корня квадратного, на выходе которого получается колебание  $U(t)$ , т.е. закон амплитудной модуляции

$$\sqrt{U^2(t)\sin^2 \varphi + U^2(t)\cos^2 \varphi} = U(t).$$

Таким образом, данный подход предполагает при проведении численных расчетов использование комплексной огибающей, точнее ее

дискретных значений. При этом сами процессы детектирования, протекающие в АМ-детекторе, не моделируются, и в формулах свойства реального детектора не учитываются.

**Реальное функциональное моделирование** нелинейных преобразователей должно проводиться с учетом свойств преобразующего элемента. Пусть АМ-детектор построен по схеме рис. 4.5, а. Тогда его функциональную схему можно представить в виде, показанном на рис. 4.5,в, где  $F$  – вольтамперная характеристика диода,  $K(p)$  – коэффициент передачи  $RC$ -цепи нагрузки.



**Рис. 4.5. К понятию модели реального АМ-детектора:**

а, б – принципиальные схемы последовательного и параллельного АМ-детектора соответственно; в – функциональная схема замещения

Заметим, что структура функциональной схемы детектора не обязательно должна отражать структуру физической схемы, так как одной и той же функциональной схеме (см. рис.4.5, в) могут соответствовать разные физические схемы (см. рис.4.5, а, б). Необязательность соответствия функциональной и физической схем – характерная черта функционального моделирования, целью которого является воспроизведение последовательности процессов преобразования сигнала, которое не всегда

совпадает с последовательностью расположения реальных физических компонентов схемы.

В заключение данного параграфа следует отметить, что при втором способе, основанном на представлении нелинейного инерционного элемента упрощенной функциональной схемой замещения типа изображенной на рис. 4.5, в, можно достаточно эффективно использовать математический аппарат функциональных рядов Вольтерры-Винера [9], как это сделано в пакете прикладных программ *Microwave Office* [3].

### 4.3 Общая характеристика программных пакетов, используемых на уровне АФЛП

Наибольшей известностью среди разработчиков РЭА пользуются три программных пакета: *ADS*, *MATLAB* и *System View* (или *System Vue*<sup>TM</sup>).

Дадим краткую характеристику этим пакетам:

- *System View* фирмы *Elanix* ([www.elanix.com](http://www.elanix.com)) – предназначен для системотехнического моделирования систем аналоговой, цифровой и аналого-цифровой обработки сигналов, систем связи, систем автоматического регулирования и управления и др. Он представляет собой «конструктор», позволяющий из стандартных «кубиков – черных ящиков» (как - то: усилителей, перемножителей, модуляторов, демодуляторов, генераторов, источников различных сигналов и др.) создавать функциональные схемы устройств и выполнять моделирование при воздействии на них различных сигналов и помех.

- *Matlab* (сокращение от англ. "Matrix Laboratory", см. сайт [www.mathworks.com](http://www.mathworks.com)) – это пакет прикладных программ для решения задач технических вычислений и одноименный язык программирования, используемый в данном пакете. Он является высокоэффективным инструментом для научных и инженерных вычислений, а также для

программирования с использованием несложно осваемой операционной среды. Наиболее известные области применения системы *Matlab* : математика и вычисления, разработка алгоритмов, вычислительный эксперимент, имитационное моделирование, анализ данных, научная и инженерная графика.

– *ADS (Advanced Design System)* фирмы *Agilent* – это пакет, предназначенный для моделирования с целью анализа характеристик современных радиосистем, содержащих блоки аналоговой и цифровой обработки сигналов различного функционального назначения.

Необходимо также упомянуть о существовании современных интегрированных САПР для проектирования сверхбольших ИМС (СБИС), например, СБИС типа систем на одном кристалле – «*system on chip*». Такие СБИС содержат до  $10^6$  и более электронных компонентов на одном кристалле. Они выполняются по совмещенной *BiCMOS*-технологии и содержат, как правило, три части: аналоговую, аналого-цифровую (АЦП и ЦАП) и цифровую (на процессорах цифровой обработки сигналов). Эти САПР состоят из большого числа программ, различающихся ориентацией на различные проектные процедуры и разные типы схем. Наиболее известными разработчиками интегрированных САПР являются фирмы *Synopsys*, *Cadence Design System*, *Mentor Graphics*.

В нашем курсе в качестве рабочего выбран пакет *System View* (версии 4.0, 4.5, 5.0 и 6.0) [4,5,6]. Его возможности расширяются за счет:

- использования библиотек пользователей, написанных на языках C и C++;
- использования моделей, созданных в программе *Matlab*;
- использования описаний цифровых процессоров обработки сигнала на языках *VHDL* и *Verilog*;
- создания специализированных ( иерархических) структур (метасистем).

В апреле 2005 года в связи с объединением компаний *Eagleware* и *Elanix* название пакета было изменено на *System Vue*<sup>TM</sup>. В августе 2005 года компания *Agilent Technologies* ([www.agilent.com](http://www.agilent.com)) приобрела объединенную компанию *Eagleware-Elanix* и в 2008 году анонсировала свою новую разработку *SystemVue2008* – новую платформу САПР для проектирования на системном уровне. По мнению специалистов, платформа *SystemVue2008* позволяет вдвое сократить время проектирования высокопроизводительных алгоритмов связи как в области беспроводных приложений, так и для аэрокосмической и оборонной промышленности.

#### 4.4. Математические модели базовых блоков, используемых в пакете *System View*

Следует отметить, что основное достоинство этого пакета – наличие большого количества специализированных библиотек. В их состав входят модели различных функциональных блоков, которые принадлежат к классу цифровых моделей.

Основной набор (*Main Libraries*) содержит следующие библиотеки:

- *Source* – библиотека источников внешних детерминированных и случайных сигналов (группы *Periodic*, *Noise/PN*, *Aperiodic*). При необходимости отсчеты сигналов сложной формы можно считывать из внешних файлов (группа *Import*);

- *Adder* - сумматор, формирует сумму от 2 до 20 сигналов;

- *Multiplier* – перемножитель, формирует произведение от 2 до 20 сигналов;

- *Function* – библиотека нелинейных функций содержит блоки, описанные математическим выражением  $y(t) = F(x(t))$ , где  $x(t)$ - входная,  $y(t)$  – выходная функция времени и  $F$  – нелинейный оператор. В состав этой

библиотеки входит разнообразный набор операторов, содержащихся в разделах *NonLinear*, *Functions*, *Complex*, *Algebraic*, *Phase/Freq*, *Multiplex*.

В группу *NonLinear* помещены разнообразные нелинейные безынерционные функциональные блоки: ограничители, квантователи, выпрямители, устройства взятия модуля и др. Предусмотрена возможность табличного задания нелинейной функции во внешнем файле.

В группу *Functions* помещены блоки для расчета тригонометрических и других функций.

В группу *Complex* помещены функциональные блоки, предназначенные для выполнения операций с комплексными переменными.

В группу *Algebraic* помещены блоки, предназначенные для выполнения алгебраических операций.

В группу *Phase/Freq* помещены фазовые и частотные модуляторы при гармонической несущей.

Группа *Multiplex* содержит устройство прореживания и уплотнения данных.

- *Operator* – библиотека операторов содержит функциональные блоки, выполняющие линейные преобразования входных сигналов. В ее состав входит разнообразный набор операторов, содержащихся в группах *Filter/Systems*, *Sample/Hold*, *Logic*, *Integral/Diff*, *Delays*, *Gain/Scale*.

Блоки группы *Filter/Systems* выполняют разнообразные линейные операции, включая цифровую и аналоговую фильтрацию.

Операторы *Sample/Hold* изменяют частоту дискретизации сигналов.

Операторы группы *Logic* выполняют логические операции над выборочными значениями сигнала (устройства выполнения

поразрядных логических операций, включенных в дополнительные библиотеки цифровых процессов обработки сигналов *DSP*).

Операторы группы *Integral/Diff* выполняют операции интегрирования и дифференцирования.

Операторы группы *Delay* выполняют задержку сигналов на заданное время или количество выборок.

Операторы *Gain/Scale* выполняют усиление (или масштабирование) аналоговых и цифровых сигналов.

- *Sink* – библиотека анализаторов данных, в которых запоминаются выходные данные блоков, к которым они подсоединены. Анализаторы данных сосредоточены в группах *Analysis*, *Numeric*, *Graphic* и *Export*.

В группе *Analysis* размещены анализаторы, предназначенные для сбора и последующего анализа данных.

Анализаторы группы *Numeric* обеспечивают вывод данных в окне проекта в текстовом виде.

Анализаторы группы *Graphic* обеспечивают вывод данных в окне проекта в виде графиков.

Анализаторы группы *Export* обеспечивают вывод данных во внешний файл.

В дополнительный набор (*Optional Libraries*) входят ряд библиотек, в том числе:

- *Communication Library* - библиотека компонентов систем связи включает в себя функциональные блоки, осуществляющие модуляцию и демодуляцию, кодирование и декодирование, передачу и восстановление данных, а также имитацию каналов связи. В ней имеется модель квадратурного демодулятора Костаса и модель системы фазовой автоподстройки частоты (ФАПЧ).

- *RF/Analog* – библиотека моделей высокочастотных аналоговых устройств содержит модели современных усилителей и аттенюаторов,

активных и пассивных смесителей, делителей мощности и сумматоров, а также все возможные  $RC$ - и  $LC$ -фильтры и ряд схем на интегральных операционных усилителях.

- *Logic* и *DSP* – библиотеки для моделирования цифровых устройств.

Как правило, каждый функциональный блок из перечисленных выше библиотек имеет определенное назначение, которое формулируется в его описании и характеризуется перечнем параметров своей модели.

Приведем примеры задания параметров некоторых функциональных блоков.

1. Для модели источника гармонического сигнала (*Sinusoid*) задаются следующие входные параметры:

- амплитуда (*Amplitude*), В;
- частота (*Frequency*), Гц;
- начальная фаза (*Phase*), град.

2. Для модели псевдослучайной импульсной последовательности (*PN Sequence*), которая генерирует псевдослучайную последовательность многоуровневых импульсов равной длительности, задаются следующие входные параметры:

- амплитуда (*Amplitude*), В;
- количество уровней (*No.Levels*);
- смещение (*Offset*) В;
- начальная фаза (*Phase*), град.

3. Для модели усилителя (*Gain*) указывается коэффициент усиления и выбор способа представления коэффициента усиления (*Gain Units*).

4. Для модели амплитудного ограничителя (*Limit*) указываются как входные, так и выходные параметры:

- максимальное входное напряжение ( $Input_{max}$ ) и
- максимальное выходное напряжение ( $Output_{max}$ ).



5. Для модели частотного модулятора (*Freq.Mod*), на выходе которого формируются две квадратурные составляющие, задаются следующие входные параметры:

- амплитуда (*Amplitude*), В;
- частота несущей (*Frequency*), Гц;
- начальная фаза несущей (*Phase*), град;
- масштабный коэффициент  $G$  (*Mod Gain*).

6. Для модели амплитудного модулятора (*DSB-AM*), который выполняет функцию модулятора амплитудно-модулированного сигнала *DSBAM*, задаются следующие входные параметры:

- амплитуда несущей (*Amplitude*), В;
- частота несущей (*Frequency*), Гц;
- фаза несущей (*Phase*), град;
- коэффициент модуляции (*Mod Index*).

7. Для модели двухвходового аналогового компаратора задаются следующие выходные параметры:

- задержка (*Gate Delay*), с;
- напряжение логической единицы (*True Output*), В;
- напряжение логического нуля (*False Output*), В.

### *Контрольные вопросы*

1. Рассмотрите функцию преобразования для двух базовых блоков: генераторов сигналов и безынерционных блоков (линейных и нелинейных).
2. Проанализируйте функцию преобразования для инерционных блоков: линейного и нелинейного.
3. Обсудите модели четырех типовых функциональных блоков.

4. Поясните, какова структура пакета *System VIEW*, предназначенного для функционального моделирования и проектирования.
5. Приведите примеры моделей функциональных блоков, используемых в программном пакете *System VIEW*, а также их входные и выходные параметры.