

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ
БЕЛАРУСЬ**

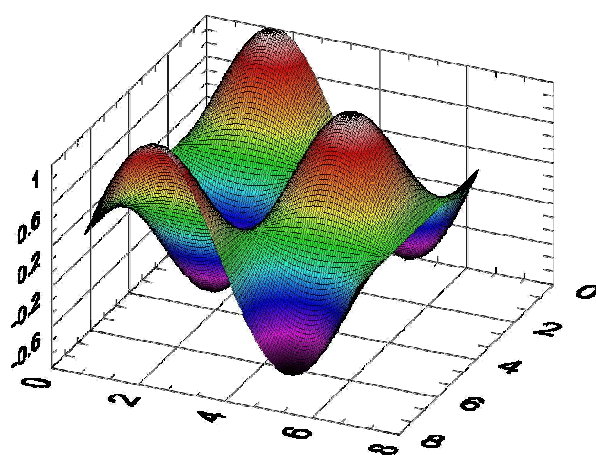
**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТРАНСПОРТА»**

**Кафедра
«Микропроцессорная техника и информационно-управляющие
системы»**

М. В. НАФТОЛЬСКИЙ, В. В. ШЕВЦОВ

СПЕЦИЗМЕРЕНИЯ И ТЕХНИЧЕСКАЯ ДИАГНОСТИКА

Лабораторный практикум



Гомель 2016

ЛАБОРАТОРНАЯ РАБОТА № 2

ИССЛЕДОВАНИЕ ФУНКЦИЙ И ПОСТРОЕНИЕ СЛОЖНЫХ КРИВЫХ В СРЕДЕ LABVIEW

Цель работы: изучение возможностей среды LabVIEW. Основные приемы программирования и отладки программ.

Задание 1: разработать программу вычисления координат и построения графика окружности, заданной параметрическим способом:

$$X=X_0+R*\text{COS}(A),$$

$$Y=Y_0+R*\text{SIN}(A),$$

где R – радиус окружности, $R = 3$;

X_0 – координата центра окружности, $X_0 = -1$;

Y_0 – координата центра окружности, $Y_0 = 2$;

A – параметр, $A [0; 360^\circ]$.

Задание 2: разработать виртуальный прибор, который демонстрирует на графическом экране фигуры Лиссажу. Модифицировать разработанную схему для получения фигур Лиссажу, дополнив ее изменением не только частот, но и фазовой составляющей (по заданию преподавателя).

Задание 3: в качестве настоящего задания предлагается найти примеры использования среды для исследования функций, то есть построения графиков самой функции и ее производной, нахождения нулей и экстремальных значений. Для этого необходимо выбрать меню: Help → Find Examples → Analyzing and Processing Signals → Signal Processing → Waveform Generation Using Formula.vi и найти в нем подходящий аналог решаемой задачи.

Основные сведения

Инженерная среда программирования LabVIEW создана для разработки компьютерных систем измерения и управления экспериментальными средствами и технологическими процессами. Ее главной особенностью является графический код, который позволяет быстро проектировать и отлаживать сложные программы. В LabVIEW существует большая библиотека примеров

использования этой среды в различных областях знаний и практического использования в управлении технологическими процессами. Особенности этих примеров является возможность использования предлагаемых решений в целом или частично в качестве программ или подпрограмм в самостоятельных разработках.

Кроме того, LabVIEW может использоваться для проведения различных расчетов, вычисления аналитических функций и построения их графиков; анализа и архивирования данных и генерации отчетов. Среда обладает уникальными возможностями быстрой отладки программ и наблюдения за потоком данных в анимационном режиме.

Порядок выполнения задания 1

1.1. Осуществите запуск среды LabVIEW из каталога (например, D:\LABV). В появившемся главном окне программы выберите команды: *New* → *Blank VI* для создания нового файла. Далее выберите меню: *Window* → *Tile Left and Right* для одновременного отображения на экране двух окон программы: серой и белой панелей. Серая *Лицевая панель* (обычно располагается слева) – инструмент пользователя, который предназначен для размещения элементов ввода и вывода данных в виде привычных технических устройств, таких как: цифровые указатели, ползунковые реостаты, регуляторы громкости, осциллографы, самописцы, графопостроители и т.д. Белая (обычно располагается справа) – *Блок-диаграмма*, на которой вызываются пиктограммы различных функций и структур, и составляет графический код программы.

Для совершения различных операций с помощью курсора необходимо вызвать *Палитру инструментов* с помощью меню: *View* → *Show Tools Palette* на *Лицевой панели* или на *Блок-диаграмм*.

1.2. Создайте на *Лицевой панели* четыре цифровых элемента управления для исходных данных задачи: X0, Y0, R, A, как показано на рисунке 1. Для этого щелчком ПКМ (правой кнопки мыши) по серой панели вызовите *Палитру элементов управления* (Controls) и закрепите ее, активизировав кнопку в левом верхнем углу палитры.

Откройте пункт меню *Num Ctrl*, выберите в нем первый элемент в верхнем ряду. На открывшейся *Палитре элементов управления* выделить элемент *Num Ctrl* со спаренной кнопкой изменения значения параметров (рисунок 2.1). Переместите четыре элемента поочередно перетаскиванием на *Лицевую панель* и расположите их горизонтально в одну строку.

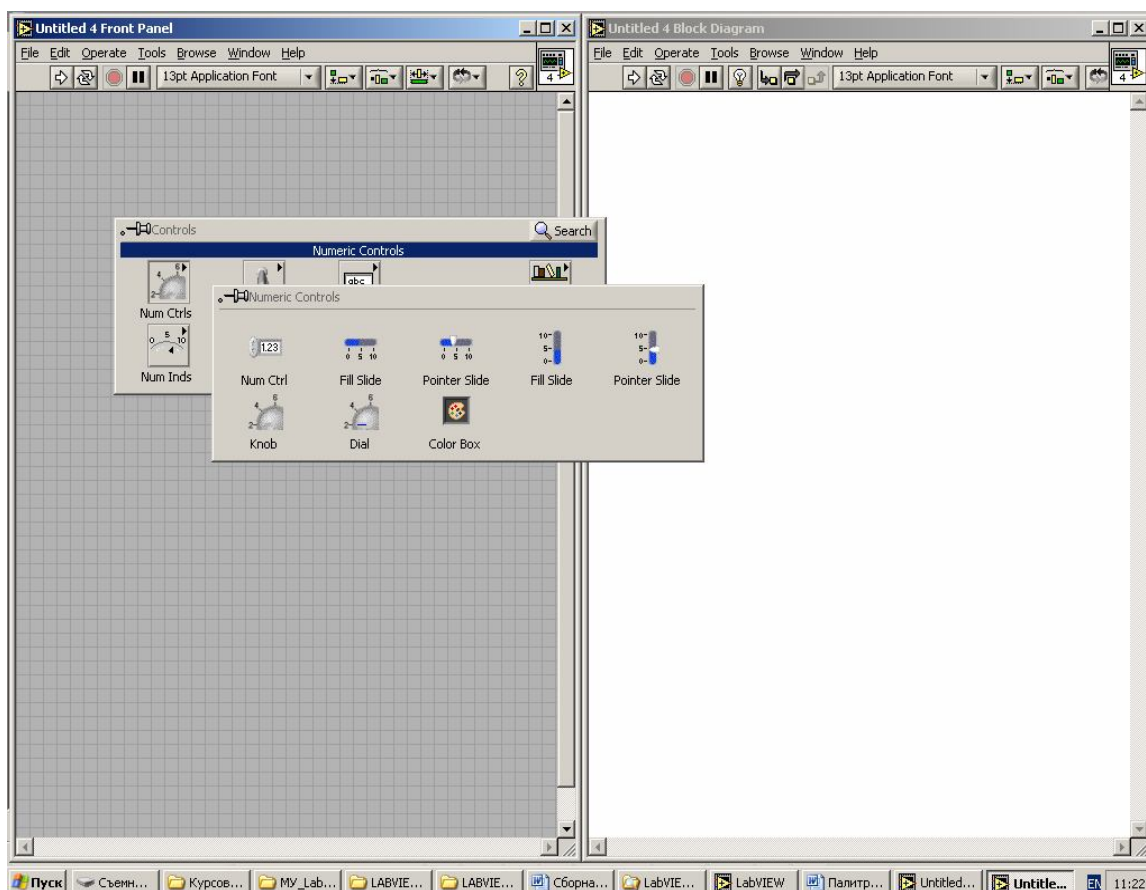


Рисунок 2.1 - Палитра элементов управления

Измените собственные метки вызванных регуляторов, подписав вместо *Numeric* новые обозначения: X_0 , Y_0 , R , A . Установите в окошках регуляторов соответствующие значения исходных данных. Значения $X_0=-1$, $Y_0=2$, $R=3$ набираются с помощью клавиш указателей, а число A , равное 360 , с помощью инструмента ВВОД ТЕКСТА.

Создайте на *Лицевой панели* два прибора для отображения полученных данных - двухлучевой запоминающий осциллограф, работающий в режиме реального времени, и двухкоординатный самописец. Для этого вернитесь на главную панель *Палитры элементов управления* (Controls), откройте графические

индикаторы (Graph Inds), из которых на *Лицевую панель* выносятся первый (Waveform Chart) и третий элемент (Express XY-Graph) (рисунок 2.2).

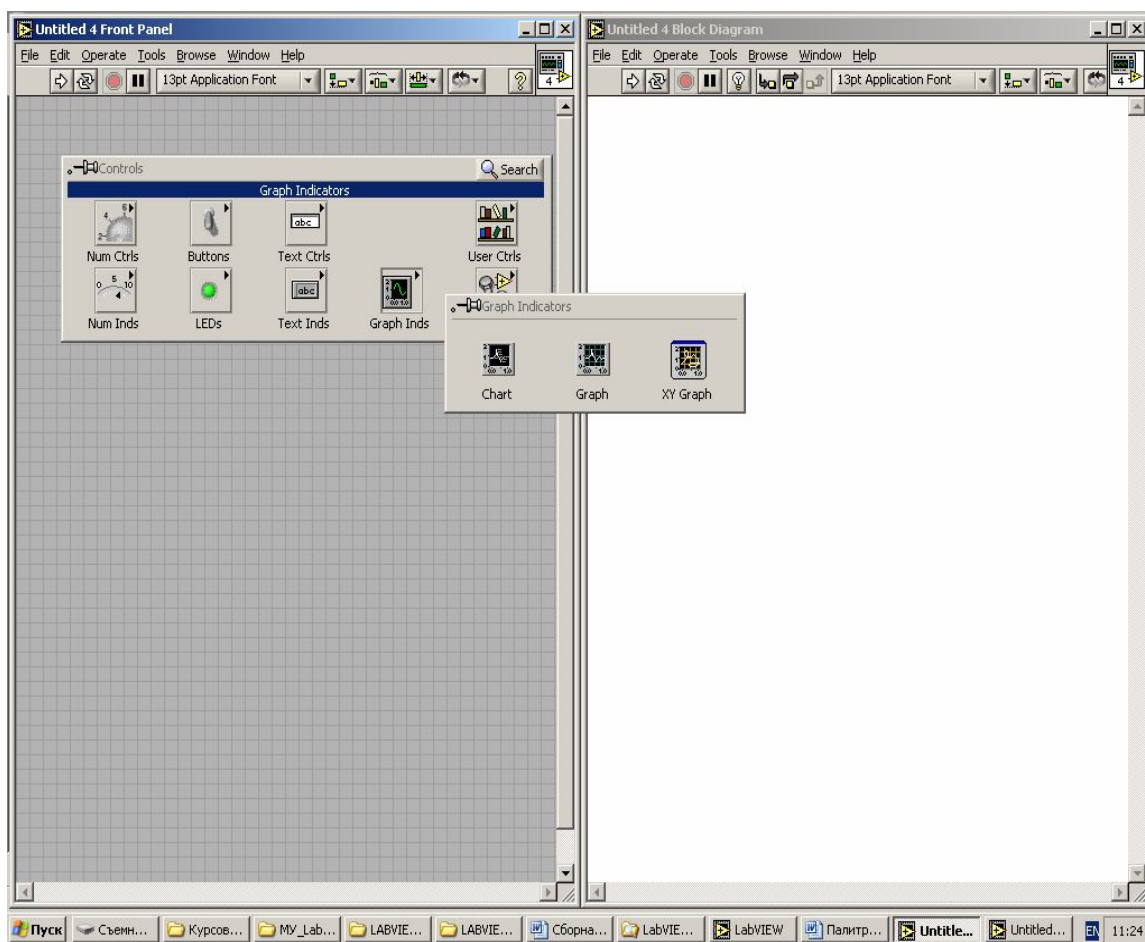


Рисунок 2.2 - Графические индикаторы

1.3 Обратите внимание на то, что при появлении элементов на *Лицевой панели*, их пиктограммы сразу же появляются на *Блок-диаграмме*. Освободите среднюю часть *Блок-диаграммы* для построения графического кода программы путем перетаскивания имеющихся элементов в верхнюю и нижнюю части экрана.

Далее щелчком ПКМ на *Блок-диаграмме* вызовите *Палитру функций* (Functions → Express → V (выпадающее меню после Select a VI ...) → Programming. Используя кнопку в верхнем левом углу палитры, нужно зафиксировать ее на экране. В *Палитре Programming* вызываем первый элемент «Структуры» в первом ряду в виде квадрата с утолщенными сторонами, далее в нем выбираем «цикл, повторяющий вычисления с заданным числом

итераций - For loop». Перетаскиваем его на блок-диаграмму и растягиваем на большую часть экрана (рисунок 2.3).

Вернитесь к *Палитре Express*, откройте панель для операций с числами (*Express* → *Arith/Compare* → *Numeric*) и зафиксируйте ее на экране. Далее реализуйте формулы, заданные в условии задачи, по пунктам 1.4 - 1.6. с помощью узлов подпалитры *Function* → *Express* → *Numeric* (рисунок 2.4).

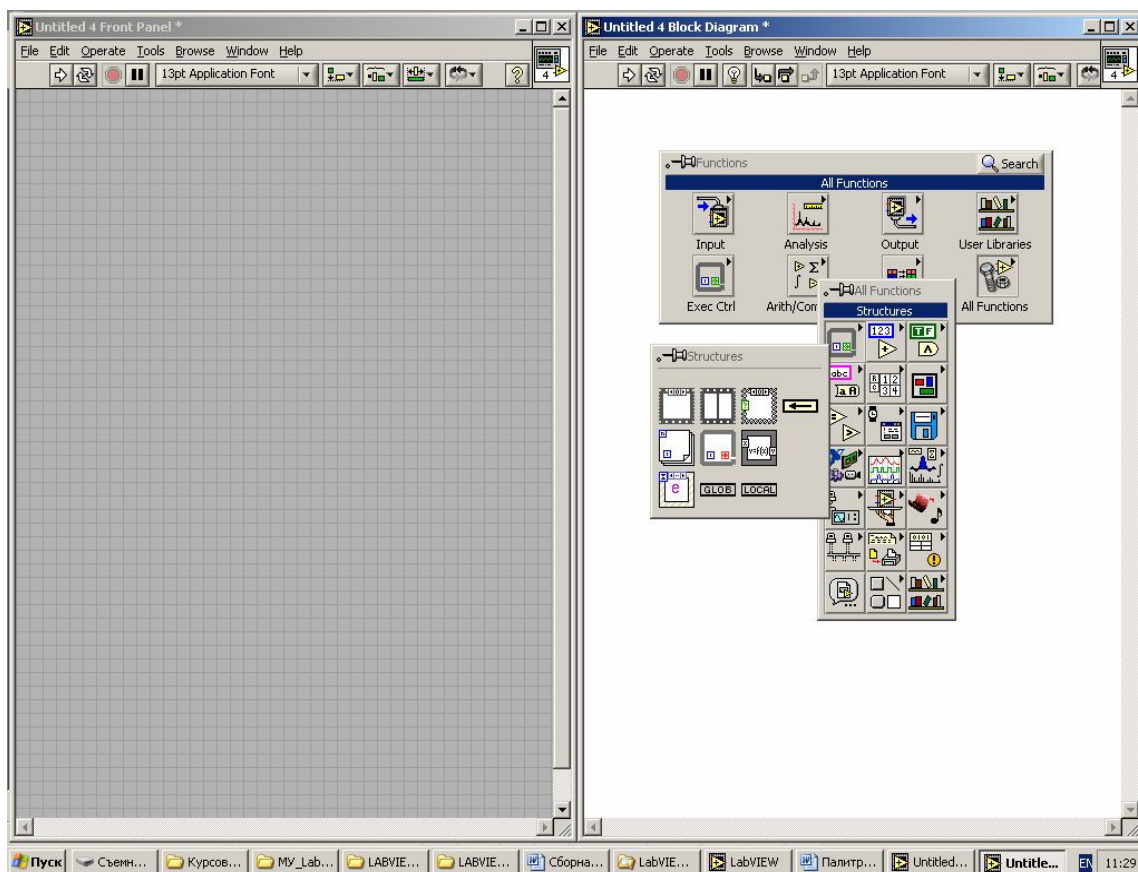


Рисунок 2.3 - Палитра функций для выбора циклических структур

1.4. Поскольку здесь, как и во многих других средах, тригонометрические функции работают с радианной мерой угла, переведите градусы в радианы. Для этого в палитре операций с числами выберите число 2π в *Numeric* → *Math Const...* → 2π и перетащите его внутрь цикла на *Панели блок-диаграмм*.

Вернувшись в *Numeric* (можно - стрелка вверх в левом верхнем углу), перенесите на *Блок диаграмму* два арифметических узла: деление и сложение. С помощью "катушки" выведите значение элемента цифрового управления A на границу цикла.

Дополнительным щелчком создайте здесь терминал (заштрихованный квадратик) ввода данных в цикл. Соедините выход узла 2π с верхним терминалом узла деления, а к нижнему с помощью проводника подведите значение A . Входы каждой пиктограммы располагаются на ее левой стороне, а выходы – на правой.

Выход узла деления соединяем с одним из входов узла умножения, а на другой подаем значение текущей итерации от узла "i" внутри цикла. Полученное на его выходе значение равно текущему значению A в радианной мере. При выполнении цикла "i" последовательно принимает значения от 0 до 359, а величина A меняется от 0 до 2π .

1.5 Найдите тригонометрические функции и пиктограммы функций SIN и COS (Functions → Arith/Compare → Math → Trigonometric → SIN или COS). Перенесите их внутрь цикла и с помощью инструмента "катушка" подайте на их входы значение A .

1.6 Вернитесь на панель арифметических действий и поставьте за каждой функцией SIN и COS пиктограмму умножения "*" и сложения "+" (Functions → Arith/Compare → Numeric → элемент умножения "*" или сложения "+").

Умножьте R на значение $\text{SIN}(A)$ и $\text{COS}(A)$. Выходы узлов умножения просуммируйте с Y_0 и X_0 соответственно. Выходы сумматоров выведите на правую границу цикла.

1.7 Соедините образовавшиеся массивы данных X и Y с соответствующими входами XY-самописца для получения графика окружности. Обратите внимание на «сломанную стрелку» запуска программы. Активизируйте ее ЛКМ (левой кнопкой мыши), в появившемся окне появится перечень ошибок, допущенных при составлении программы. Одна из них: не заполненный терминал счетчика итераций, который подает команду на завершение циклических вычислений. Исправьте эту ошибку, продублировав соединение элемента управления A с внешним входом счетчика итераций «N». Остальные ошибки, если они присутствуют, исправьте самостоятельно.

1.8 После исправления ошибок, запустите программу и убедитесь, что построенная кривая является окружностью с радиусом $R=3$ и центром окружности, расположенным в точке $X_0=-1$, $Y_0=2$.

Запустите программу в режиме анимации потоков данных. Для этого активизируйте в строке команд на *Блок-диаграмме* элемент в виде "лампочки", а затем стрелку запуска программы. Изучите движение данных и их трансформацию в различных узлах *Блок- диаграммы* и сделайте выводы о возможности использования этого режима при отладке сложных программ.

1.9 Для отображения изменения координат окружности в зависимости от параметра внесите внутрь цикла двухлучевой осциллограф. Для преобразования его в многолучевой прибор на *Палитре функций* (Functions→ Express→Programming) откройте третью пиктограмму во второй строке (Cluster), а в ней пиктограмму – Bundle (объединение), захватите ее и перенесите внутрь цикла, к входу осциллографа. К двум ее входам подключите X и Y, взятые от проводников соответствующих данных. Выход объединительной панели соедините с осциллографом.

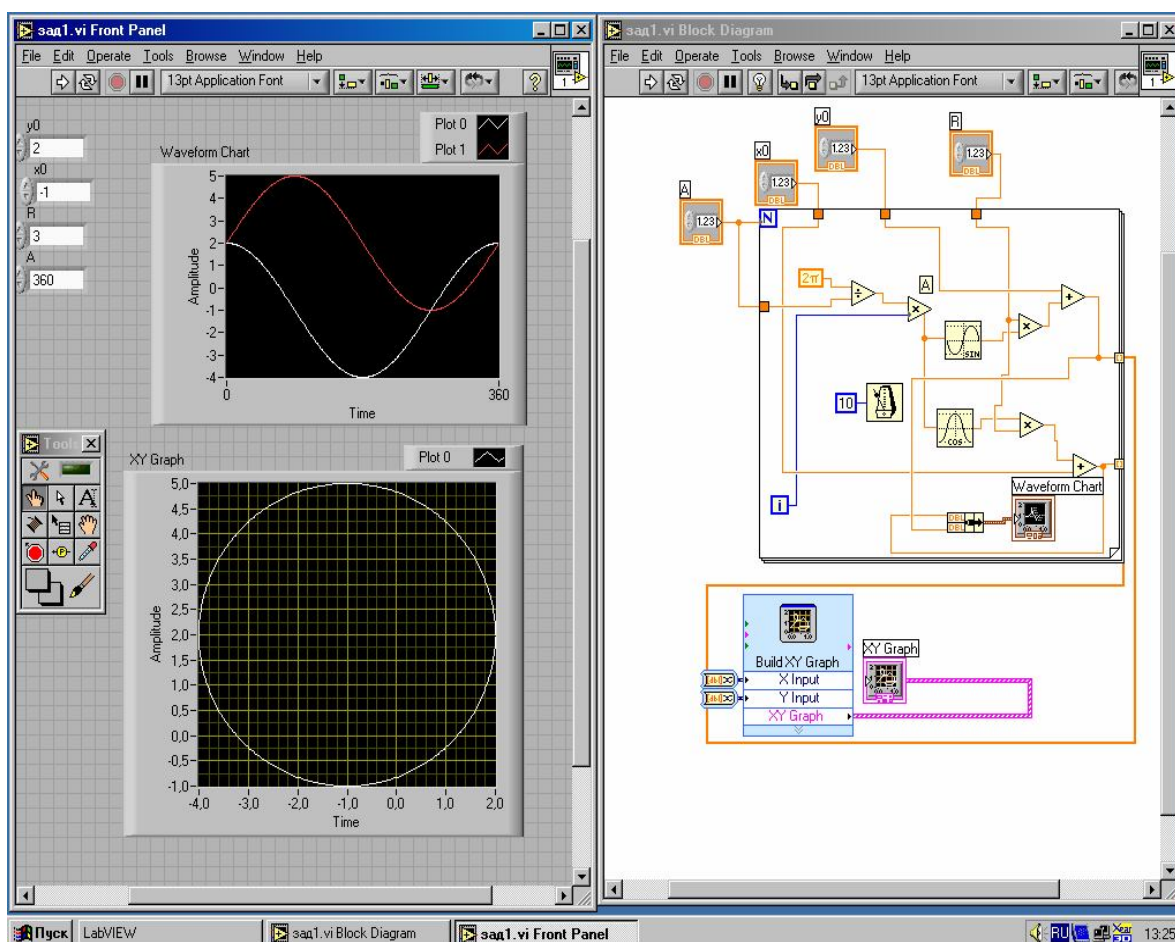


Рисунок 2.4 - Лицевая панель и блок-диаграмма расчета координат и построения кривой окружности

1.10. Для того чтобы замедлить вычисления и наблюдать построение кривых координат окружностей во времени, с *Палитры функций* (ПКМ→Functions→Express→Programming→Timing) вызывается пиктограмма Wait Unit Next ms Multiple, в виде метронома, который переносится внутрь цикла. По умолчанию, цикл рассчитывает всю программу со скоростью 1000 раз в секунду. Подведите инструмент "катушка" с левой стороны "метронома" к терминалу. Выберите команды: ПКМ→Create→Constant и с клавиатуры наберите время задержки 20 миллисекунд. При этом процесс расчета и отображения координат X и Y замедлится в 20 раз.

Запустите программу и просмотрите процесс построения графиков. Обратите внимание, что первые графики строятся в реальном времени, а сама кривая в координатах X-Y на самописце только после окончания всего цикла вычислений.

1.11. Перенесите в отчет по лабораторной работе блок-диаграмму вычислений и построения графиков в соответствии с рисунком 2.4. Сделайте выводы по результатам выполненной работы.

Порядок выполнения задания 2

Фигуры Лиссажу – замкнутые траектории, прочерчиваемые точкой, совершающей одновременно два гармонических колебания в двух взаимно перпендикулярных направлениях. Впервые изучены французским учёным Жюлем Антуаном Лиссажу. Вид фигур зависит от соотношения между периодами (частотами), фазами и амплитудами обоих колебаний (рисунок 2.5). В простейшем случае равенства обоих периодов фигуры представляют собой эллипсы, которые при разности фаз 0 или π вырождаются в отрезки прямых, а при разности фаз $\frac{\pi}{2}$ и равенстве амплитуд превращаются в окружность. Если периоды обоих колебаний неточно совпадают, то разность фаз всё время меняется, вследствие чего эллипс всё время деформируется. При существенно различных периодах фигуры Лиссажу не наблюдаются. Однако если периоды относятся как целые числа, то через промежуток времени, равный наименьшему кратному обоим периодам, движущаяся точка снова возвращается в то же положение — получаются фигуры Лиссажу более сложной формы. Фигуры Лиссажу вписываются в прямоугольник, центр

которого совпадает с началом координат, а стороны параллельны осям координат и расположены по обе стороны от них на расстояниях, равных амплитудам колебаний.

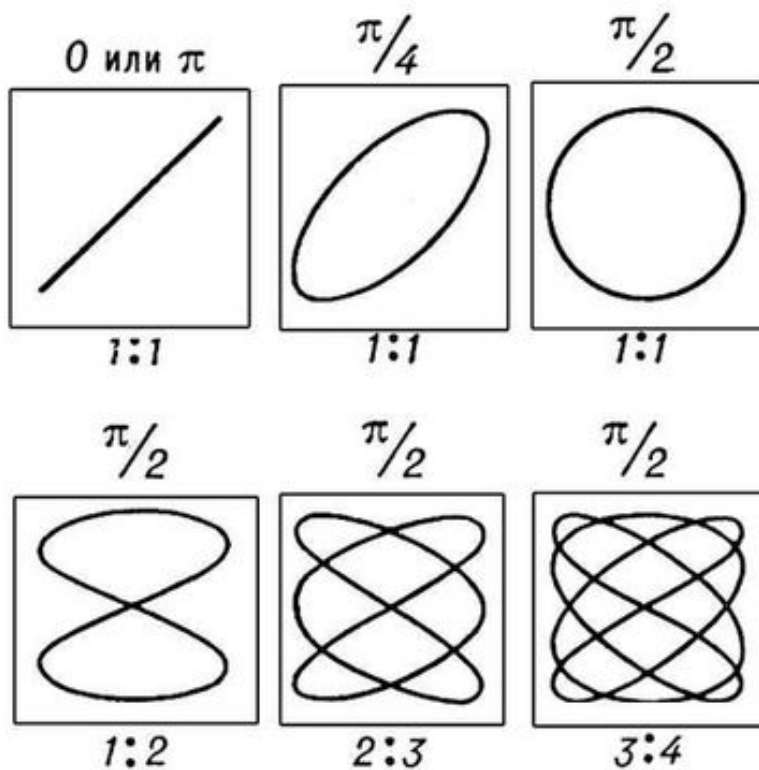


Рисунок 2.5 - Соотношения между периодами, фазами и амплитудами обоих колебаний

Математическое выражение для кривой Лиссажу

$$\begin{cases} x(t) = A \sin(at + \delta) \\ y(t) = B \sin(bt) \end{cases}$$

где A, B — амплитуды колебаний, a и b — частоты, δ — сдвиг фаз

Вид кривой сильно зависит от соотношения a/b . Когда соотношение равно 1, фигура Лиссажу имеет вид эллипса, при определённых условиях она имеет вид прямой ($A = B, \delta = \pi/2$ радиан) и отрезка прямой ($\delta = 0$). Ещё один пример фигуры Лиссажу — парабола ($a/b = 2, \delta = \pi/2$). При других соотношениях фигуры Лиссажу представляют собой более сложные фигуры,

которые являются замкнутыми при условии a/b — рациональное число.

2.1. Виртуальный прибор для наблюдения фигур Лиссажу можно создать, если модифицировать блок-диаграмму расчета координат и построения кривой окружности, представленной на рисунке 2.4. Для этого используются два формирователя гармонических сигналов, один (или оба) из которых позволяет плавно регулировать частоту колебаний.

Требования к отчету

Наименование и цель работы; краткое описание интерфейса LabVIEW (меню, тулбары, горячие клавиши), палитра-меню инструментов (Tools Palette); разработанная программа вычисления координат и построения графика окружности, заданной параметрическим способом (лицевая панель и блок-диаграмма); создание виртуального прибора в соответствии с заданием 2; внешний вид передней панели, блок-диаграмма виртуального прибора; демонстрация работы программы в соответствии с заданием преподавателя (по рисунку 2.5); пояснение работы блок диаграммы выбранного примера использования среды программирования LabVIEW для исследования функций (построения графиков самой функции и ее производной, нахождения нулей и экстремальных значений); заключение.

Контрольные вопросы

1. Назовите области возможного применения в процессе дальнейшего обучения.
2. Каковы отличительные черты интерфейса среды?
3. В чем заключаются принципиальные отличия разработки программ в графической и текстовой средах программирования?
4. Как увеличить число каналов компьютерного осциллографа?
5. Какие инструменты служат для отладки программ и обнаружения ошибок? Для каких целей используется режим анимации потоков данных?

6. Чем отличается работа узлов и индикаторов, расположенных внутри и снаружи цикла?
7. Чем отличаются циклы For и While?
8. В каких случаях применяется функция задержки времени?

СПИСОК ЛИТЕРАТУРЫ

1. Бутырин, П.А. Автоматизация физических исследований и эксперимента: компьютерные измерения и виртуальные приборы на основе LabVIEW 7 Express / П.А. Бутырин, Т.А. Васьковская, В.В. Каратаев, С.В. Материкин – М.: ДМК Пресс, 2005. - 264 с.
2. Суранов, С.Я. LabVIEW 8.5: справочник по функциям / С.Я. Суранов. - М.: ДМК Пресс, 2007. - 572 с.
3. Визильтер, Ю.В. Обработка и анализ цифровых изображений с примерами на LabVIEW IMAQ Vision / С.Ю. Желтов, В.А. Князь, А.Н. Ходарев, А.В. Моржин. - М.: ДМК Пресс, 2007. - 464 с.
4. Батоврин, В.К. LabVIEW: практикум по электронике и измерительной технике / В.К. Батоврин, А.С. Бессонов, В.В. Мошкин. - М.: ДМК Пресс, 2007.
5. Батоврин, В.К. LabVIEW: практикум по основам измерительных технологий / В.К. Батоврин, А.С. Бессонов, В.В. Мошкин. - М.: ДМК Пресс, 2007.
6. Евдокимов, Ю.К. LabVIEW для радиоинженера: от виртуальной модели до реального прибора / Ю.К. Евдокимов, В.Р. Линдваль, Г.И. Щербаков. - М.: ДМК Пресс, 2007.
7. Федосов, В.П. Цифровая обработка сигналов в LabVIEW / В.П. Федосов, А.К. Нестеренко. - М.: ДМК Пресс, 2007.

Краткое описание интерфейса LabVIEW - меню, тулбары, горячие клавиши

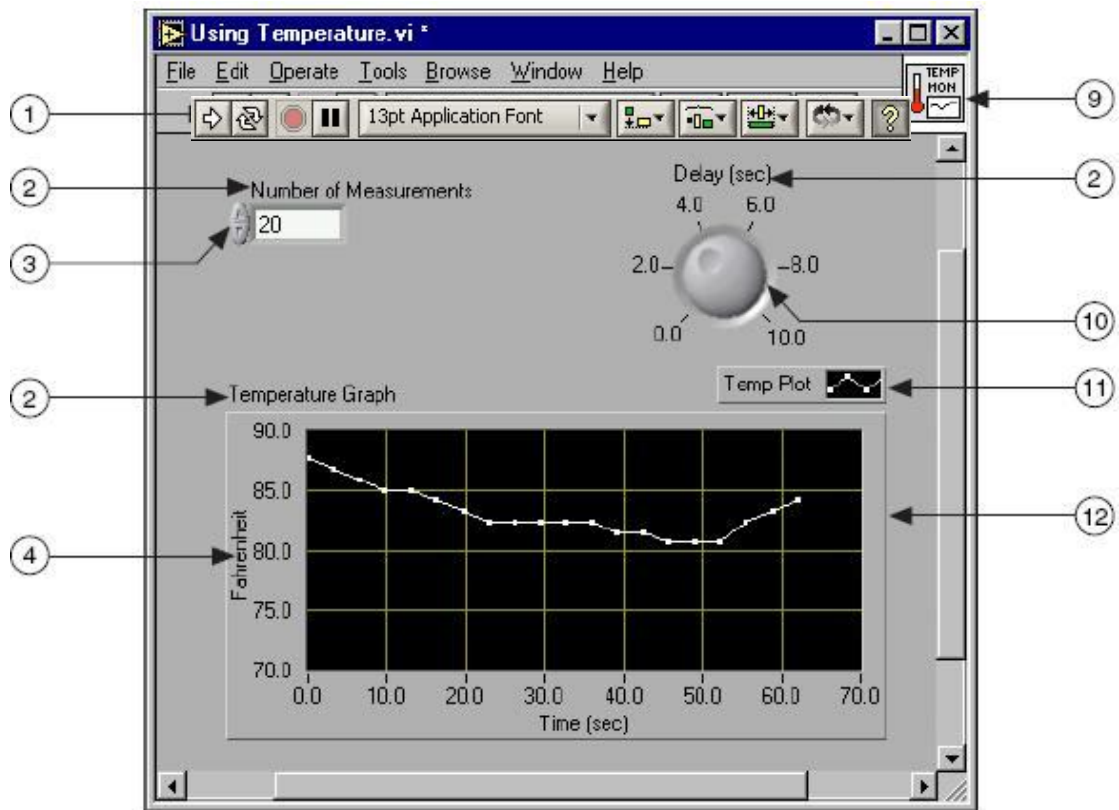


Рис. П1. Передняя панель виртуального прибора.

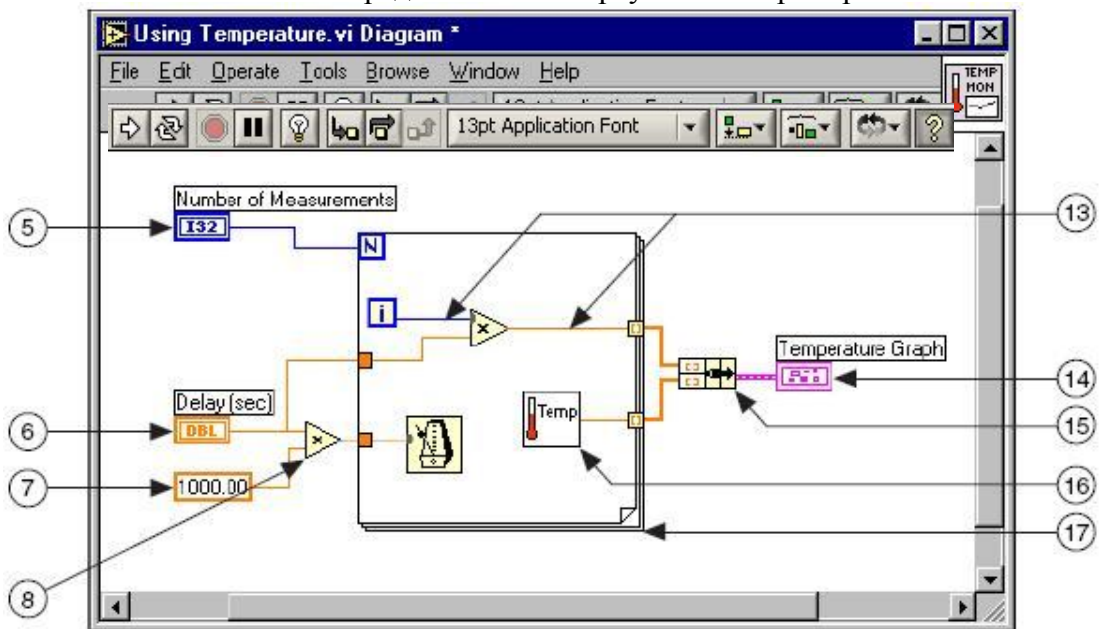


Рис. П2. Блок-диаграмма виртуального прибора.

1. Тулбар (Toolbar)
2. Метка-название элемента передней панели (Label)
3. Элемент «числовой регулятор» (Digital Numeric Control)
4. Метка-название оси на элементе «график» (Scale Label)
5. Терминал элемента «числовой регулятор» (поз.3) (Digital Numeric Control Terminal)
6. Терминал элемента «числовой регулятор» (поз.10) (Knob Terminal)
7. Константа на панели блок-диаграммы (Numeric Constant)
8. Узел-функция «умножения» (Multiply Function)
9. Пиктограмма ВП (Icon)
10. Элемент «числовой регулятор» (Knob Control)
11. «Легенда» графика (Graph Legend)
12. График (XY Graph)
13. Провод на панели блок-диаграммы
14. Терминал элемента График (поз.12) (XY Graph Terminal)
15. Функция сборки элементов в кластер (Bundle Function)
16. Подпрограмма (Sub VI)
17. Конструкция программирования «Цикл For» (For Loop Structure)

Тулбары

Для окон передней панели и блок-диаграммы вид тулбаров несколько отличается. Тулбар панели блок-диаграммы имеет дополнительные кнопки для отладки ВП







Рис. П3. Тулбар окна передней панели (Front Panel Toolbar)



Рис. П4. Тулбар окна блок-диаграммы (Block Diagram Toolbar)

	запустить ВП
	вид кнопки запуска ВП в процессе работы
	компилятор обнаружил ошибки в коде ВП, нажатие на кнопку откроет список ошибок.
	циклический непрерывный запуск ВП, отмена циклического запуска
	аварийный останов ВП
	приостанов выполнения ВП, переключение пошагового режима выполнения ВП
13pt Application Font	настройка атрибутов отображения текста
	выравнивание и распределение элементов на панели
	показать/спрятать окно контекстно-чувствительной справочной системы (версии LabVIEW 7.0 и выше)

Дополнительные кнопки тулбара окна блок-диаграммы	
	замедленное выполнение ВП с режимом визуализации, показ текущих данных, передаваемых каждым проводом
	пошаговое исполнение с заходом «внутри» конструкции программирования или подпрограммы (SubVI)
	пошаговое исполнение без захода «внутри» конструкции программирования или подпрограммы (SubVI)
	выполнение программы ВП до завершения текущего блока исходного кода, после - останов

Палитры-меню (Palette)

Палитра-меню Controls Palette используется для создания передней панели ВП. Она содержит в иерархической форме все элементы пользовательского интерфейса (числовые регуляторы и индикаторы, логические, графики, кнопки и т.п.).

Палитра-меню элементов блок-диаграммы (Functions Palette) используется для создания блок-диаграммы ВП. Содержит в иерархической форме все встроенные функции и подпрограммы (SubVI), распределенные по функциональному назначению (арифметика, работа со строками, массивами, кластерами, файловый ввод-вывод и т.п.).

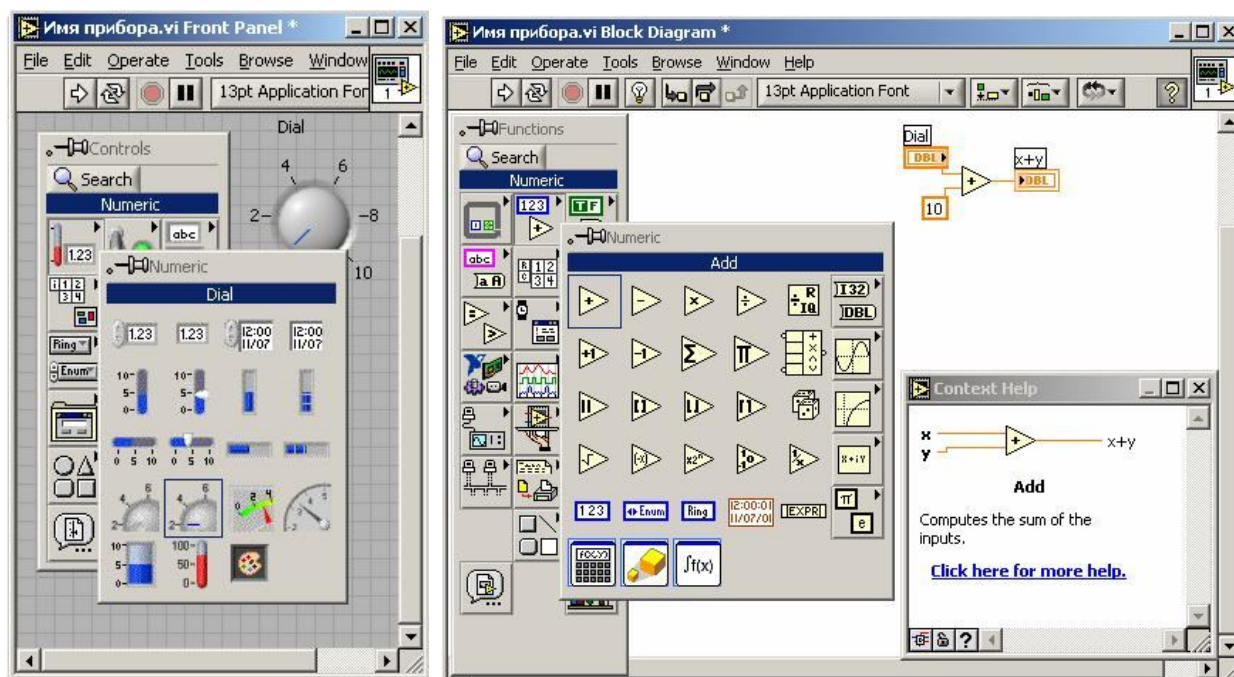


Рис. П5. Палитра-меню *Controls Palette* на окне передней панели (слева) и палитра-меню *Functions Palette* окне блок-диаграммы (справа).

Для отображения палитр на экране используются пункты меню *Window>>Show Controls Palette* окна передней панели ВП и *Window>>Show Functions Palette* окна блок-диаграммы ВП. Также палитры-меню могут быть вызваны щелчком правой кнопки мыши на свободном месте экрана передней панели или блок-диаграммы ВП. Меню *Tools>>Options* позволяет настроить LabVIEW так, чтобы палитры-меню отображались на экране автоматически при запуске LabVIEW.

Палитра-меню инструментов (Tools Palette).



Содержит инструменты, используемые для разработки ВП. Отобразить палитру можно с помощью пункта меню *Window>>Show Tools Palette*, либо комбинацией клавиш Shift+правая кнопка мыши.

Рис. Пб. Палитра-меню инструментов.

	«указательный палец» (Operate Value) – изменение значений регуляторов, констант, нажатие кнопок. Этот инструмент автоматически включается во время запуска ВП.
	«стрелка» (Position/Size/Select) – основной инструмент редактирования передней панели и блок диаграммы ВП. Позволяет менять расположение, размеры элементов, группировать их вместе.
	«текст» (Edit Text) – позволяет менять текстовые поля в любом месте рабочей области LabVIEW, например, менять названия элементов, настраивать оси координат графиков, создавать свободные текстовые комментарии на передней панели и блок-диаграмме ВП и т.п.
	«катушка» (Connect Wire) – используется для прокладки и подключения проводов на блок диаграмме, а также при создании коннектора подпрограммы (SubVI) – указывает соответствие контакта коннектора ВП и элемента передней панели ВП.
	«всплывающее меню свойств» (Object Shortcut Menu) – аналогичен правой кнопке мыши. Отображает всплывающее меню свойств объекта или палитры-меню элементов передней панели или блок-диаграммы ВП.
	«ладонь» (Scroll Window) – перемещает рабочее поле передней панели или блок-диаграммы в любом направлении относительно окна программы. Используется в случае больших размеров передней панели/блок-диаграммы или недостаточном разрешении видеосистемы компьютера.
	«точка останова» (Set/Clear Breakpoint) – используется при отладке ВП. Назначает/снимает точку останова для любого узла ВП. Точкой останова можно управлять из всплывающего меню свойств узла.
	«пробник» (Probe Data) – используется для просмотра значений данных, передаваемых по любому проводу блок-диаграммы. Пробником можно управлять из всплывающего меню свойств нужного провода.
	«пипетка» (Get Color) – запоминает цвет выбранного объекта.
	«палитра цветов» и «кисть» (Set Color) – позволяет выбрать цвет фона и текста, а также раскрасить выбранный объект.

Меню

Большинство пунктов меню являются стандартными, как и для большинства Windows-приложений. Отметим лишь некоторые пункты меню, характерные для LabVIEW.

Раздел меню	Пункт меню	Горячие клавиши	Назначение
File			
	New VI	Ctrl+N	Создать новый ВП
	Save All		Сохранить все ВП (включая всю иерархию ВП)
	Save with Options		Сохранить с опциями (например, защитить паролем или записать в библиотеку ВП)
	VI Properties	Ctrl+I	Настроить свойства ВП
Edit			
	Undo	Ctrl+Z	Отменить последнее действие (количество шагов «отката» можно настроить в меню <i>Tools>>Options</i>)
	Customize Control		Редактировать внешний вид элемента передней панели, позволяет создавать собственные элементы передней панели
	Remove Broken Wires	Ctrl+B	Удалить с блок-диаграммы все «плохие», разорванные провода
	Create SubVI		Создать подпрограмму на основе выделенного на блок-диаграмме кода
	Run-Time		Создание и редактирование меню ВП, которое будет отображаться и работать во время выполнения ВП
Operate			
	Run	Ctrl+R	Запуск ВП из оболочки LabVIEW
	Stop	Ctrl+ S	Останов ВП
Tools			
	Measurement & Automation Explorer		Запуск интерактивного драйвера оборудования National Instruments (MAX)
	Instrumentation		Подключение драйверов приборов к LabVIEW
	Data Acquisition		Интерактивный помощник создания приложений для работы с многофункциональными измерительно-управляющими платами National Instruments
	Build Application or Shared Library (DLL)		Компиляция исходного кода, создание EXE-файла или библиотеки динамической компоновки DLL
	Edit VI Library		Редактировать библиотеку ВП.
	Options		Настройки LabVIEW.

Примечание:

Расположение пунктов меню может несколько отличаться в разных версиях LabVIEW.

Browse			
	Show VI Hierarchy		Показать полную иерархию ВП (ВП верхнего уровня и все связанные подпрограммы)
Window			
	Show Block Diagram/Show Front Panel	Ctrl+E	Открыть блок-диаграмму/переднюю панель текущего ВП (удобно при одновременном редактировании нескольких ВП)
	Show Controls Palette (в меню окна передней панели)		Открыть палитру-меню элементов построения передней панели ВП
	Show Functions Palette (в меню окна блок-диаграммы)		Открыть палитру-меню построения блок-диаграммы ВП
	Show Tools Palette		Открыть палитру инструментов
	Show Error List	Ctrl+L	Показать список предупреждений и ошибок компилятора LabVIEW
Help			
	Show Context Help	Ctrl+H	Показать/спрятать окно контекстно-справочной системы
	VI, Function, & How – To Help	Ctrl+?	Открыть справочник по функциям LabVIEW
	Search The LabVIEW Bookshelf		Поисковая система по электронной документации LabVIEW
	Find Examples		Поиск примеров ВП по тематике задачи или по имени
	Explain Error		Расшифровать числовое значение кода ошибки

Наиболее часто употребляемые «горячие» клавиши LabVIEW (Hot Keys)

Ctrl+B	Удалить все "плохие" провода с блок-диаграммы
Ctrl+E	Найти и показать окно передней панели или блок-диаграммы ВП
Ctrl+S	Сохранить ВП ²¹
Ctrl+R	Запуск ВП
правая кнопка мыши	1. Щелчок на объекте показывает всплывающее меню свойств объекта. 2. Щелчок на пустом месте экрана отображает палитру-меню элементов пользовательского интерфейса, или палитру-меню конструкций программирования, функций и подпрограмм
Shift+правая кнопка	Показ всплывающего меню для выбора инструмента

Примечание:

* В LabVIEW нет механизма автоматического сохранения программ через заданный промежуток времени.

Различные способы расчета функции $N!$

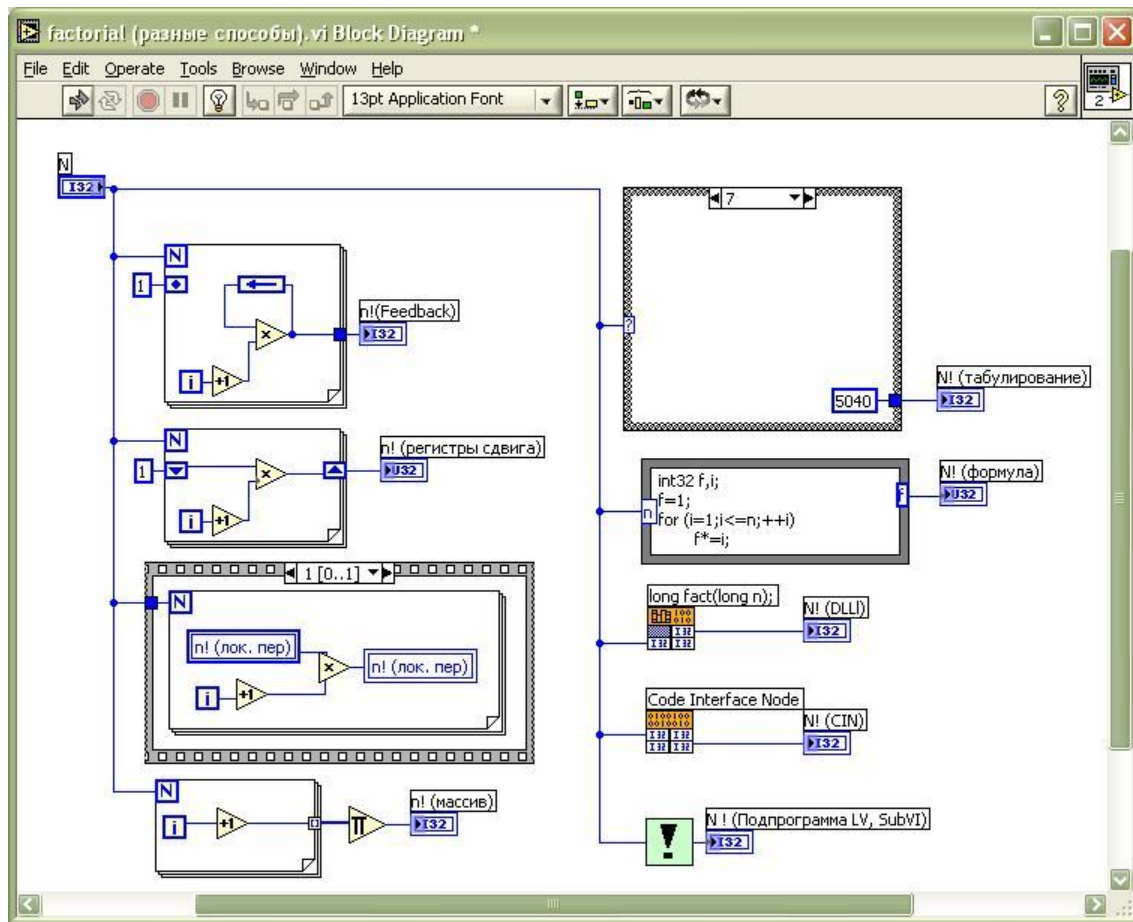


Рис П7. Различные способы расчета значения $N!$

На Рис. П7 приведено девять разных способов определения значения $N!$.

- использование конструкции Feedback Node;
- использование регистра сдвига (Shift Register);
- использование конструкций "последовательность" и "локальные данные";
- формирование массива чисел $[1,2,3...N]$ в цикле с использованием индексующего туннеля, затем перемножение элементов
- табличное задание функции, использование конструкции "условие";
- использование конструкции "формула";
- вызов функции библиотеки динамической компоновки DLL
- интегрирование кода программы на "СИ" в LabVIEW (Code Interface Node);
- использование ранее созданной подпрограммы (SubVI).