Муромский институт (филиал)
федерального государственного бюджетного образовательного учреждения
высшего профессионального образования
«Владимирский государственный университет имени Александра Григорьевича и
Николая Григорьевича Столетовых»

На правах рукописи

МИХЕЕВ КИРИЛЛ ВАЛЕРЬЕВИЧ

РАЗРАБОТКА ВЫЧИСЛИТЕЛЬНЫХ АЛГОРИТМОВ ДЛЯ УСТРОЙСТВ ОБРАБОТКИ И ОТОБРАЖЕНИЯ ИНФОРМАЦИИ РАДИОТЕХНИЧЕСКИХ СИСТЕМ

05.12.04 – «Радиотехника, в том числе системы и устройства телевидения»

Диссертация

на соискание ученой степени кандидата технических наук

Научный руководитель:
д.т.н., профессор Чекушкин В.В.

Муром 2016

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Основой интеллектуальных радиотехнических систем (РТС) и радиоэлектронных комплексов являются высокопроизводительные специализированные вычислительные устройства, которые обеспечивают реализацию различных задач по цифровой обработке информации, обнаружению, определению координат воздушных объектов, управлению, контролю, диагностике, экспертным оценкам, тренировке операторов и т.д. [1 -3]. При решении таких задач требуется моделировать структуры подобных систем, использовать более совершенные численные методы, ориентированные на современные аппаратные средства вычислительных систем, а также разрабатывать и улучшать алгоритмы обработки информации различного назначения. Повышение эффективности связано с обеспечением предельных оптимальных соотношений по точностным характеристикам, быстродействию, программно-аппаратурным затратам, а так же реализуемости выбранных алгоритмов обработки на современных программно-аппаратных платформах.

Существующие подходы к решению отмеченных задач, нашедшие отражение в научных работах отечественных и зарубежных исследователей ориентированы в первую очередь на математическую строну проблемы, и, в большинстве своем, отражают различные математические методы реализации функциональных зависимостей. Однако, в последнее время достижения информационных технологий и вычислительной техники предоставляют исследователям и инженерам-практикам с одной стороны использовать в разрабатываемой аппаратуре различные сложные алгоритмы и методы обработки информации, обеспечивающие работу систем в реальном масштабе времени. С другой стороны, современная элементная база цифровой обработки информации отличается большим разнообразием, в которую входят компьютеры, микроЭВМ, специализированные вычислители, программируемые логические интегральные схемы (ПЛИС) [4-11], что требует от разработчиков адаптировать алгоритмы обработки под используемую элементную базу.

Теоретические основы численных методов и методов аппроксимации были заложены крупнейшими учёными своего времени, а именно, И. Ньютоном, Л. Эйлером, Н.И. Лобачевским, К.Ф. Гауссом, которые построили математическую базу теорий, она активно развивается и в настоящее время. Основоположником теории аппроксимации функций является великий русский математик П.Л. Чебышев. В качестве приближающих функций выбирались чаще всего алгебраические и тригонометрические многочлены, кроме этого Чебышевым был предложен метод наилучшего приближения, который используется и до настоящего времени. На дальнейшее развитие этой теории оказало влияние открытие, сделанное в конце XIX века немецким математиком Карлом Вейерштрассом, которым была доказана принципиальная возможность приближения произвольной непрерывной функции с любой заданной степенью точности алгебраическим многочленом, Дальнейшее развитие теории было продолжено другими известными учеными – Б. Тейлором, Ж. Фурье, Ш. Эрмитом, М. Сэвиджем и другими. С появлением первых аналоговых и, впоследствии, цифровых вычислительных устройств, автоматизирующих процессы обработки различного рода информации, потребовали от исследователей разработки прикладных методов аппроксимации, которые были внедрены в существующие в то время вычислительные устройства отечественными и зарубежными учеными, именно: В.Д. Байковым, В.Б. Смоловым, Б. Волдером, П. Безье, Р.У. Хэммингом и другими. Важный этап в развитии теории, связанный с именами Ш.Ж. Балле Пуссена, Д. Джексона, С.Н. Бернштейна, составили исследования связи между скоростью убывания погрешности приближения функции выбранными тем или иным способом многочленами различных степеней. В этих работах приведённая погрешность была равна порядка 0,1 %, что недостаточно для современных радиотехнических систем. В то время в силу неразвитости вычислительной техники, применяемые алгоритмы формировались на основе упрощения имеющихся подходов в теории аппроксимации и приближений. Появление высокопроизводительных вычислительных систем обработки информации потребовало от ученых и исследователей разработки новых прикладных численных методов и методов аппроксимации, позволивших с одной стороны реализовать существенно сложные алгоритмы обработки, с другой стороны, на порядок и более повысить точность представления функций. Решением этих задач в конце прошлого начале нынешнего века в нашей стране и за рубежом занимались ученые: С.В. Конягин, B. Jacob, Ж.П. Обэн, Б.С. Кашин, Ю.Ф. Опадчий, Е.В Чумакова, С.С. Кукушкин, В.Н. Захаров, В.А. Ларионов, Э.А. Акчурин, В.В. Чекушкин и другие. В работах вышеперечисленных ученых алгоритмы автоматического поиска полиномов наилучшего приближения и алгоритмы аппроксимации стандартных функций, арифметических и векторных операций были реализованы без взаимной компенсации составляющих погрешностей, что не сокращало значения результирующей погрешности. Возникает необходимость разработки новых алгоритмов, которые позволяют сократить разрядные сетки операндов специализированных вычислителей на 2-5 двоичных разрядов и уменьшить значения погрешности в 2-3 раза. В настоящее время мы переживаем резкий скачок производительности специализированных вычислительных систем, а так же появление новых структур вычислителей, построенных на различных принципах обработки информации, которые требуют новых алгоритмов и подходов обработки. Методы и подходы, полученные учеными за предыдущие годы для решения существующих задач аппроксимации, численных методов и моделирования траекторий не позволяют обеспечить предельные точностные характеристики, высокую загрузку устройств обработки, а также эффективные вычисления. Имеющиеся в настоящее время результаты методов аппроксимации функциональных зависимостей и численные методы должны быть адаптированы под существующую новую и перспективную элементную базу вычислительных устройств, более того разработка новых алгоритмов и модификация существующих позволит в разы повысить точность аппроксимации функциональных зависимостей, либо на порядок увеличить производительность вычислителей, а так же приблизить формируемые траектории движения объектов к реальным, что в конечном итоге существенно улучшит тактико-технические характеристики РТС. Кроме этого, необходима разработка алгоритмов, обеспечивающих работу вычислительных устройств использующих двоичное представление чисел в 64 и более разрядов, модификация методов приближений под решение таких задач отсутствует в известных источниках и работах ведущих ученых.

Существующие численные методы решения типовых задач, которые входят в вычислительные процедуры основных алгоритмов работы РТС, не являются наилучшими в современных вычислителях, поскольку ориентированы на другие их структуры. Программно-аппаратурная реализация вычислительных устройств основывается на известных численных методах, ряд типовых задач по измерению и преобразованию координат, формированию физических эталонов для систем контроля, машинной графике, воспроизведению траекторий, вычислению функций, калибровки измерительных систем, которые приводятся в научно-технической литературе могут быть значительно улучшены [12].

Решение отмеченных задач позволит использовать методы аппроксимации функциональных зависимостей под конкретную программно-аппаратную платформу, а разработанные программы автоматизировать этот процесс при ограничениях: разрядные сетки операндов, быстродействие систем обработки, эффективность и сложность вычислительного процесса.

Новые подходы к построению траекторий движения воздушных объектов (ВО) позволят приблизить полученные модели к реальным траекториям движения воздушных судов, обеспечив простой и удобный алгоритм реализации в вычислительных системах, тем самым повысив качество формируемой тренажной информации и высокие тактико-технические характеристики всей системы в целом.

Цели и задачи диссертационной работы

Целью работы является разработка алгоритмов аппроксимации различных функциональных зависимостей и моделирования траекторий движения воздушных объектов для устройств цифровой обработки и отображения информации радиотехнических систем, обеспечивающих высокое быстродействие, необходимые точностные характеристики и минимальные программно - аппаратурные затраты.

Поставленная цель достигается решением следующих **задач**:

- существенного упрощения реализации сложных функциональных зависимостей путем их представления в виде суперпозиции более простых функций и разработки соответствующих полиномиальных преобразований Чебышева, обеспечивающих повышение точности аппроксимации и минимизации программно - аппаратурных затрат;

- моделирования реализуемого в системах цифровой обработки радиотехнической информации вычислительного процесса для обеспечения сокращения вычислительных затрат, значительного уменьшения погрешности результата за счёт взаимного поглощения и взаимной компенсации составляющих погрешностей;

- разработки структуры вычислителя, построенного на универсальном микропроцессоре или ПЛИС, ориентированного на наиболее рациональную реализацию конкретной вычислительной задачи с устранением избыточной точности;

- разработки метода воспроизведения траекторий ВО из плавно сопрягаемых сегментов при различных законах изменений линейной скорости с обеспечением контроля перегрузок в трехмерном пространстве, обеспечивающего реалистичное задание тренажной информации.

Методология и методы исследования

При проведении исследований в диссертационной работе использованы численные методы чебышевской аппроксимации функциональных зависимостей, вариационного исчисления, теории оптимизации, моделирования. Реализация и анализ полученных решений осуществлялись с применением методов вычислительной математики, математического моделирования, различных программных продуктов.

Научная новизна исследования:

- разработаны методы и алгоритмы поиска полиномов наилучшего приближения различных степеней для аппроксимации функциональных зависимостей, повышающих точность представления типовых функций и минимизацию программно-аппаратурных затрат в цифровых вычислителях и синтезаторах частот;

- созданы алгоритмы аппроксимации стандартных функций на основе взаимной компенсации составляющих погрешностей при уменьшении разрядов операндов до 3-6;

- для гибридных алгоритмов преобразования координат, ортогональных составляющих сигналов в амплитуду и фазу увеличено быстродействие в 2 раза. Рациональное использование предлагаемых алгоритмов позволяет обеспечить формирование от 1 до 32 и более значащих двоичных разрядов операндов, избегая невостребованной избыточной точности результата;

- разработан новый метод воспроизведения траекторий ВО из плавно сопрягаемых сегментов кривых на основе параметрических уравнений Безье при различных законах изменений линейной скорости с контролем перегрузок.

Основные положения, выносимые на защиту:

- алгоритмы автоматического поиска полиномов наилучшего приближения для решения задачи восстановления значения функции, отличающиеся тем, что фиксация оптимального значения одной из переменных одновременно обеспечивает уменьшение диапазонов последующего поиска оптимальных значений оставшихся переменных;

- алгоритмы аппроксимации стандартных функций, арифметических и векторных операций с диапазоном представления от 3 до 64 двоичных разрядов результата с устранением избыточной точности, отличающиеся тем, что разрядные сетки операндов сокращены на 2-5 двоичных разрядов без снижения класса точности измерительного прибора;

- алгоритмы автоматического поиска и ускоренные алгоритмы преобразования координат, ортогональных составляющих сигналов в амплитуду и фазу, реализации преобразования комплексного сигнала из алгебраического представления в экспоненциальное с обеспечением выигрыша по быстродействию до 2-х раз;

- метод воспроизведения траекторий ВО из плавно сопрягаемых сегментов, полученных на основе параметрических уравнений кривых Безье при различных законах изменений линейной скорости с контролем перегрузок.

Результаты работы внедрены в учебный процесс кафедры САПР Муромского института ФГБОУ ВО «Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых», а также на предприятии АО «МЗ РИП» (АО Концерн ВКО «АЛМАЗ-АНТЕЙ», г. Муром).

Практическая значимость полученных результатов:

- для решения широкого класса прикладных задач с диапазоном приведённых погрешностей выходных данных до долей процента разработаны численные методы воспроизведения стандартных функций с исключением избыточной точности и уменьшения амплитуд паразитных гармонических составляющих радиосигналов. Обеспечено повышение быстродействия системы цифровой обработки в 2 раза;

- получены алгоритмы, обеспечивающие существенное сокращение числа итерационных циклов при калибровке измерительных каналов с нестабильными параметрами и разрядных сеток операндов специализированных вычислителей на 2-5 двоичных разрядов.

- метод формирования траекторий движения ВО с контролем перегрузок, адекватный реальному движению воздушных судов, позволяющий повысить качество тренажной информации операторов;

- прикладное программное обеспечение, автоматизирующее процесс поиска полиномов наилучшего приближения для различных функциональных зависимостей, оптимизирующее полиномы под различные специализированные вычислительные устройства и обеспечивающее построение траекторий движения ВО.

Разработаны и получены свидетельства о государственной регистрации трех программ для ЭВМ: программа поиска полиномов наилучшего приближения для воспроизведения функциональных зависимостей, программа поиска полиномов наилучшего приближения для воспроизведения функциональных зависимостей с взаимной компенсацией составляющих погрешностей результата, программа поиска метода наилучшего приближения функции $R = \sqrt{A^2 + B^2}$.

Апробация результатов

Результаты исследований получены автором при выполнении гранта РФФИ №14-0700293 «Методы, математические модели оптимальной реализации вычислительных процессов в высокопроизводительных технических системах». Научные положения и выводы диссертации используются в АО «Муромский завод радиоизмерительных приборов».

Результаты работы докладывались и обсуждались на научно-технических конференциях и семинарах различного уровня:

1. IV,V,VI,VII,VIII Всероссийские научные Зворыкинские чтения «Наука и образование в развитии промышленной, социальной и экономической сфере регионов России», Муром, 2012-2016 гг;

2. V,VI,VII Всероссийских молодежных научных конференциях «Научный потенциал молодежи – будущее России». Муром, 2013-2015 гг.

3. I-ой Всероссийской научно-технической конференции «Расплетинские чтения» в ГСКБ Алмаз-Антей, г. Москва 2014 г.;

4. Третьей Всероссийской научно-практической конференции АО «Муромский завод радиоизмерительных приборов» «Радиолокационная техника: устройства, станции, системы РЛС-2015», Муром, 9-10 июня 2015 г.

5. I Всероссийской научно-технической конференции «Математическое моделирование и инженерные расчеты» в научно-образовательном центре воздушно-космической обороны «Алмаз-Антей» им. академика В.П. Ефремова, г. Москва, 2016 г.

Публикации по теме работы

Результаты исследований по теме диссертационной работы опубликованы в 15 научных работах, в том числе в 3 статьях, рекомендованных ВАК РФ, и 2 статьях, входящих в международную базу цитирования «Web of Science», 7 тезисах докладов на научно-технических конференциях международного и всероссийского уровней. Получены 3 свидетельства о государственной регистрации программ для ЭВМ.

Объем и структура работы

Диссертация состоит из введения, четырёх глав, заключения, списка используемой литературы и приложений. Общий объём работы составляет 150 страницы машинописного текста, включая 41 рисунок, 12 таблиц, а также 14 страниц приложения. Библиография содержит 104 наименования, в том числе 15 работ автора.

Работы состоит из четырех глав, заключения и списка литературы.

В первой главе рассматриваются основные принципы построения ИИС, применяемых в РТС. По известным работам отечественных и зарубежных ученых производится анализ типовых задач, решаемых ИИС при испытаниях, контроле и обработке информации в радиоэлектронных системах, выявляются недостатки существующих систем, определяются пути улучшения качества работы.

Вторая глава посвящена улучшению методов поиска полиномов наилучшего приближения для воспроизведения стандартных функций, физических эталонов, реализации градуировочных характеристик датчиков и измерительных систем при их калибровке. Рассмотрен более эффективный метод адаптации вычислительного процесса, вытекающий из стратегии максимальной идентичности графиков воспроизводимой функции и приближающего полинома. Разработаны компьютерные программы и получены аппроксимации функций и градуировочных характеристик для калибровки измерительных систем в виде полиномов наилучшего равномерного приближения степени n с ограниченным числом членов $m < (n+1)$, интерполирующих и экстраполирующих значения эталонных сигналов. Разработаны быстродействующие алгоритмы получения полиномов наилучшего приближения первой и второй степеней для аппроксимации функций и градуировочных характеристик. Сложная задача многомерной упрощения целевой функции с большим числом переменных сводится к последовательному решению ряда простых задач. Особенность алгоритмов состоит в том, что фиксация оптимального значения одной из переменных одновременно обеспечивает уменьшение диапазонов последующего поиска оптимальных значений оставшихся переменных. Стратегия уменьшения интервала неопределенности оптимального значения при одномерной оптимизации определяется скоростью изменения погрешности между соседними узлами аппроксимации.

Третья глава посвящена разработке улучшенных вычислительных алгоритмов воспроизведения стандартных функций, созданию алгоритмов вычисления тригонометрических функций $\sin(x)$, $\mathrm{tg}(x)$, $\mathrm{arctg}(x)$, $\arcsin(x)$, решению типовых задач с устранения избыточной точности путем обеспечения дискретного приращения 2… 3… и более значащих двоичных цифр результата при фиксированном возрастании сложности алгоритма не более чем на 2-3 операции в диапазоне представления выходных данных 3...32 двоичными разрядами. Разработаны быстродействующие алгоритмы воспроизведения функций $Z = \sqrt{A^2 + B^2}$, $\beta = \mathrm{arctg}(A / B)$ с оптимизацией вычислительного процесса по точностным характеристикам, быстродействию и программно-аппаратным затратам на примерах измерений амплитуд сигналов с неизвестной начальной фазой, напряжений переменного тока и фазовых углов. Разработана программа аппроксимации элементарных функций и программа анализа алгоритмов преобразования квадратурных составляющих сигнала в амплитуду и фазу.

Четвертая глава посвящена разработке методов наглядного воспроизведения траекторий ВО из плавно сопрягаемых сегментов в системе координат зоны обзора измерительной системы и преобразованием безразмерного аргумента параметрических уравнений кривых Безье при различных законах изменений линейной скорости в аргумент реального линейно–нарастающего интервала времени воспроизведения траектории. Обеспечен контроль перегрузок. При комплексировании по полной траектории движения с помощью сегментов, выполненных в виде кривых Безье 3-го и 1-го порядков, обеспечен плавный переход для широко распространенного сопряжения от кубической кривой Безье к отрезку прямой и обратно с выполнением условия равенства первой и второй производных. В среде Delphi были разработаны два независимых приложения: Air Situation Designer – для визуального монтажа воздушной обстановки и сохранения результатов работы в файл, а также Air Situation Server – для обслуживания асинхронных запросов о состоянии движения воздушных объектов, поступающих от сторонних приложений. Также была разработана программа имитации воздушной обстановки.

В заключении изложены основные результаты работы.

Глава 1 Направления совершенствования методов, применяемых в автоматизированных радиотехнических системах контроля

1.1 Направления повышения эффективности современных автоматизированных систем контроля

Системы автоматизированного контроля поддерживают работоспособность радиоэлектронной аппаратуры при ее эксплуатации, поэтому большое значение имеет встраивание таких систем в разрабатываемую и эксплуатируемую аппаратуру, обеспечивая сбор, преобразование и обработку данных от различных датчиков информации [5]. Обеспечение оптимальных условий контролепригодности упрощает решение анализа работоспособности аппаратуры как на этапах разработки, отладки опытных образцов, так и в условиях серийного производства и эксплуатации [11]. Аппаратура встроенного контроля, ее программное обеспечение, должны использоваться на всех циклах разработки, изготовления, испытаний, сертификации и эксплуатации радиотехнических систем (РТС). Именно в процессе наладки и испытаний опытного образца непосредственно разработчик совершенствует свой опыт и используемые средства встроенного контроля, производит оценку их эффективности.

При эксплуатации разработанной аппаратуры наиболее естественно использовать тестовые наборы, полученные на этапе ее проектирования и проверять работоспособность системы при помощи тех же имитационных сигналов и с такими же временными параметрами, которые были апробированы на этапах проверки макетов и опытных образцов. Задачи контроля значительно упрощаются при правильной технической постановке вопроса, связанного с назначением системы, возможностью ее иерархического построения, ясностью движения и обработки потоков информации, балансе времени, т.е. рациональном разбиении контролируемой аппаратуры на функционально законченные составные части.

Разработано большое число методов контроля цифровых устройств и цифровизированных РТС. Но в имеющихся работах еще недостаточно внимания уделяется практическим задачам разработки и оптимизации численных методов генерации тестовых воздействий, калибровки измерительных систем, синтеза универсальных встроенных систем автоматического контроля.

Для проведения испытаний, контроля и сертификации радиоэлектронных комплексов используются ИИС [12]. Они представляют собой совокупность функционально-объединенных измерительных, вычислительных и других вспомогательных технических средств для получения измерительной информации, её преобразования и обработки с целью представления потребителю в требуемом виде, либо автоматического осуществления логических функций измерений, контроля, диагностики, идентификации и т. п. Назначение любой ИИС, необходимые функциональные возможности, технические характеристики, её структура (рис. 1.1) определяются объектом исследования, испытаний, контроля, для которых данная система создаётся. При всесторонней проверке объектов необходимо осуществлять съём информации с большого числа датчиков, как например, для контроля сложных РТС необходимо в режиме реального времени получать информацию с нескольких сотен датчиков. Поэтому для временного разделения проверки параметров с большого числа каналов одной измерительной системы с целью уменьшения числа преобразователей, измерительных устройств используют многоканальные адресные коммутаторы сигналов, переключение которых осуществляется через устройство управления. Затем сигналы поступают в преобразователи, в которых впоследствии они через измерительные устройства реализуются в системах визуализации и отображения информации.

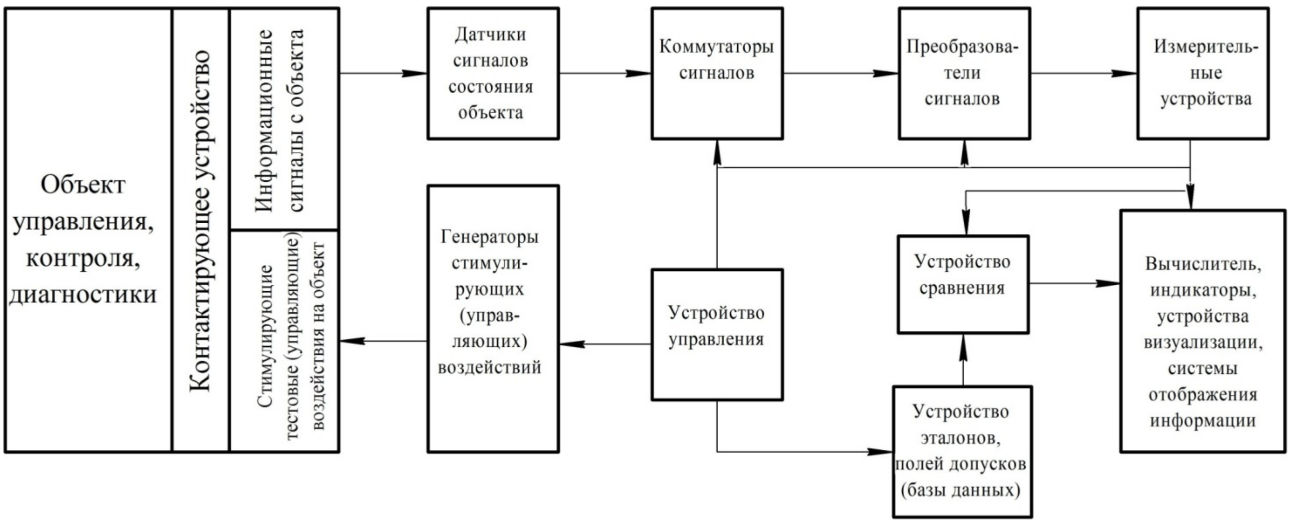


Рисунок 1.1 Структурная схема информационно-измерительной системы

В зависимости от выполняемых функций ИИС реализуются в виде:

- измерительных систем (ИС);
- систем автоматического контроля (САК);
- систем распознавания (идентификации) образов (СРО);
- телеизмерительных систем (ТИС).

ИС – это совокупность функциональных объединенных мер, измерительных приборов, измерительных преобразователей, ЭВМ и других технических средств, размещенных в разных точках контролируемого пространства с целью измерений одной или нескольких физических величин, свойственных этому пространству.

В САК и СРО измерительная система входит как подсистема. Информация, которая характеризует объект измерения и используется для процессов контроля, управления диагностики, воспринимается первичными измерительными преобразователями – датчиками и обрабатывается в ИИС по определённым алгоритмам. На выходе системы получается количественная информация, отражающая состояние объекта. Датчики информируют о состоянии объекта путём определённого взаимодействия и преобразования реакции на это взаимодействие в информационные сигналы согласно заданным алгоритмам.

ТИС – совокупность устройств на приемных и передающих сторонах и каналах связи для автоматического измерения одного или ряда параметров на расстоянии. Они могут включать в себя первичные измерительные преобразователи, блоки обработки и отображения информации, контрольные пункты, преобразователи кодов и сигналов, каналы связи и другие устройства.

Главная подсистема ИИС – это измерительное устройство, обеспечивающее заданную точность измерений, диагностирования. На выходе измерительного устройства формируется информация о техническом состоянии объекта управления, контроля. Эта информация путём преобразования и различных способов индикации, отображения может быть представлена оператору и (или) использована для дальнейшей обработки. Важным элементом такой обработки является идентификация, операция сравнения представленной информации с полем допусков для вынесения решения о виде технического состояния контролируемого или диагностируемого изделия, формирования управляющих воздействий. В связи с этим в измерительную систему может быть включено прогнозирующее устройство, которое позволяет определять состояние изделия в будущем посредством обработки о текущем и прошлом состоянии объекта контроля. Генератор тестовых сигналов необходим для формирования стимулирующих воздействий на проверяемые объекты с целью обеспечения всех вариантов заданной программы его функционирования, получения, регистрации контрольных сигналов с его выходов.

В результате процесса контроля, диагностирования выносится решение о виде технического состояния, работоспособно или неработоспособно контролируемое и (или) диагностируемое изделие. Ошибки диагностирования могут быть допущены в основном из-за неработоспособности средств контроля, диагностики и большой погрешности измерений. Работа всех устройств контролируется устройством управления, которое, например, для заданного вида, этапа контроля синхронно задаёт вид стимулирующего воздействия, назначает номер проверяемого канала и для него обеспечивает выборку соответствующего эталона измеряемого сигнала или поля допусков для подачи его на устройство сравнения и (или) отображения информации.

Для визуального отображения, вывода, регистрации и документирования результатов измерений используются самые разнообразные элементы, устройства и целые системы отображения информации, обеспечивающие превращение сигналов, например, в световые, удобные для визуального наблюдения, механические для перемещения указателя отсчётной шкалы и т.д.

Одним из путей развития РТС различного назначения является повышение эффективности их функционирования за счёт разработки новых и совершенствования известных методов проектирования, испытания, контроля и сертификации РТС. Эксплуатация сложной радиотехнической аппаратуры невозможна без встроенного автоматического функционально-диагностического контроля, введения систем имитации и тренировки операторов РТС. В настоящее время эти задачи решают интеллектуальные ИИС. Основой таких систем являются высокопроизводительные специализированные вычислительные устройства, обрабатывающие значительные потоки измерительной и тестовой информации, создающие имитационные воздействия в реальном масштабе времени. Введение подобных ИИС в эксплуатируемые и вновь разрабатываемые РТС требует совершенствования математического, алгоритмического, методического и программного обеспечения, в частности, совершенствование численных методов, разработки программно-аппаратных средств воспроизведения различных функциональных зависимостей, совершенствование методов имитации тестовых воздействий.

Особенно актуальны отмеченные свойства применяемых в настоящее время ИИС контроля в современных ИС, которые должны на современной отечественной элементной базе при создании тестовых воздействий формировать значения элементарных функций с необходимой степенью точности, реализовать арифметические и векторные операции, преобразовывать координаты и формировать траектории движения воздушных объектов с учетом ограничений на физические параметры движения.

Примером такой РТС может служить система контроля воздушного пространства, структурная схема которой приведена на рис.1.2. В такой системе блок формирования тренажерной и имитационной информации обеспечивает моделирование сложной воздушно-помеховой обстановки в зоне ответственности измерительной системы на участке местности с воспроизведением траекторий движения воздушных объектов [13]. При этом должно быть предусмотрено формирование активных и пассивных помех, отражений от метеообразований, местных предметов, привязанных к трассам движения воздушных объектов. Комплекс обеспечивает обучение и тренировку операторов систем управления и контроля воздушного пространства, проведение функционально-диагностического контроля ИИС, полунатурное моделирование процессов обнаружения, измерения, координат и формирования трасс движения воздушных целей.

Рисунок 1.2 Структурная схема ИИС контроля воздушного пространства

При контроле, диагностике ИИС осуществляется автоматическая обработка информации с комплексной оценкой технических характеристик и диагностикой составных частей в виде наборов подсистем, функциональных устройств и узлов, использованием двойственного набора параметров, являющихся диагностическими, техническими и перекрывающих информационное поле, характеризующее состояние системы в целом и ее составляющих подсистем. Управление режимами работы ИИС осуществляется с рабочего места оператора с подсистемой отображения информации с помощью графических интерфейсов со встроенными меню задания режимов работы.

Блок формирования тренажерной и имитационной информации представлен эквивалентной подсистемой формирования тренажерной и имитационной информации. В центральном диагностическом ядре производится формирование сводной информации о состоянии ИИС в виде древовидного графа с вершиной соответствующей обобщенному признаку состояния ИИС с последующим древовидным ветвлением на соответствующих сборках из отдельных подсистем, функциональных узлов и отображением соответствующей информации на подсистеме отображения информации рабочего места оператора. Подсистема формирования тренажёрной и имитационной информации дополнительно формирует как многофункциональный генератор имитационные тестовые сигналы сложной воздушной обстановки в четырёх - координатном пространстве для дополнительного моделирования, контроля и диагностики подсистем надпорогового обнаружения и коммутации сигналов, формирования отметок целей и сопровождения траекторий и рабочего места оператора с подсистемой отображения информации.

Таким образом, можно выделить следующие направления совершенствования встроенных автоматизированных систем контроля современных РТС.

Первым направлением развития ИИС является достижение потенциальных возможностей систем за счет существенного упрощения реализации сложных функциональных зависимостей путём их представления в виде суперпозиции более простых функций, разработки оптимизированных полиномиальных преобразований Чебышева.

Вторым направлением является предварительное моделирование реализуемого в ИИС вычислительного процесса с целью его оптимизации, что позволит сократить вычислительные затраты, значительно уменьшить погрешности получаемых результатов за счёт взаимного поглощения и взаимной компенсации коррелированных составляющих погрешностей.

Третьим направлением совершенствования ИИС является оптимизация структуры вычислителя под решаемые задачи. Поскольку архитектура современных цифровых вычислительных машин (ЦВМ), построенных на базе универсальных микропроцессоров, ориентированных, как правило, на обычное процедурное выполнение вычислительных алгоритмов, что далеко не всегда адекватна структуре решаемых задач. Это приводит к тому, что реальная производительность ЦВМ составляет, как правило, 5-15% от их пиковой производительности при решении широкого класса прикладных задач. С другой стороны, в настоящее время развиваются программируемые логические интегральные схемы (ПЛИС), содержащие десятки и сотни тысяч универсальных логических блоков. Из этих блоков пользователь может создать специализированную вычислительную структуру, которая будет наиболее адекватна конкретному вычислительному алгоритму и будет обладать наибольшей производительностью при наименьших вычислительных затратах. В этом плане представляет интерес улучшение принципов построения структур специализированных вычислительных систем высокой производительности, архитектура которых основывается на гетерогенных принципах и объединяет в едином вычислительном ресурсе как универсальные микропроцессоры, а также ПЛИС, и другие специализированные устройства, ориентированные на наиболее рациональную реализацию конкретной вычислительной задачи с устранением избыточной точности.

И, наконец, четвертое направление – это разработка адекватного реальной воздушной обстановке метода воспроизведения траекторий ВО из плавно сопрягаемых сегментов с преобразованием безразмерного аргумента параметрических уравнений кривых Безье при различных законах изменений линейной скорости в аргумент реального линейно–нарастающего интервала времени воспроизведения траектории, а также обеспечение контроля перегрузок.

1.2 Методы воспроизведения функциональных зависимостей в системах контроля

Задача воспроизведения функциональных зависимостей в системах контроля имеет высокую актуальность, поскольку цифровые автоматы, применяемые в РТС, не имеют возможности оперировать с аналоговыми сигналами и величинами. Так как в большинстве случаев РТС работают в реальном масштабе времени, то основное требование к алгоритмам формирования – это обеспечение достаточного быстродействия при заданной точности. Поскольку современные цифровые процессоры реализуют на аппаратном уровне элементарные арифметические и логические операции, то для вычисления более сложных функций, таких как $\sin(x)$, $\arcsin(x)$, $\sqrt{x}$ и других, используются различные методы и вычислительные алгоритмы аппроксимации. Кроме этого некоторые РТС оснащены сотнями и тысячами датчиков, передающих измерительную информацию одновременно, то в таких системах возникает проблема обработки огромного потока информации в реальном времени. Решением этих проблем является разработка и внедрение упрощенных алгоритмов и методов обработки информации.

Методы, обеспечивающие уменьшение погрешностей и повышение быстродействия цифровых вычислителей активно применяются в различных технологических процессах, связанных с визуализацией методами компьютерной графики радиотехнической информации, воспроизведением геометрических фигур и траекторий воздушных объектов, преобразованием систем координат, амплитуд и фаз сигналов, формированием тестовых и эталонных воздействий в виде гармонических и других элементарных функций и решением других задач.

Неотъемлемым элементом большинства РТС являются системы синтеза частот и сигналов. Один из основных типов подобных устройств, применяемых в настоящее время, являются цифровые вычислительные синтезаторы (ЦВС). Высокое разрешение по частоте и фазе, максимально быстрый переход на другую частоту без разрыва фазы, возможность управления частотой, фазой, а в некоторых случаях и амплитудой по цифровому интерфейсу постоянно расширяют сферу применения таких устройств. В классическом ЦВС (рис.1.3) функциональный преобразователь реализован на основе ПЗУ, хранящего функцию синуса в табличном виде. Для улучшения спектральных характеристик выходного сигнала ЦВС необходимо повышать точность формирования фазы гармонического колебания, что приводит к снижению быстродействия, увеличению программно-аппаратурных затрат устройства. Для решения этой задачи на практике нашли применения методы аппроксимации гармонического сигнала.

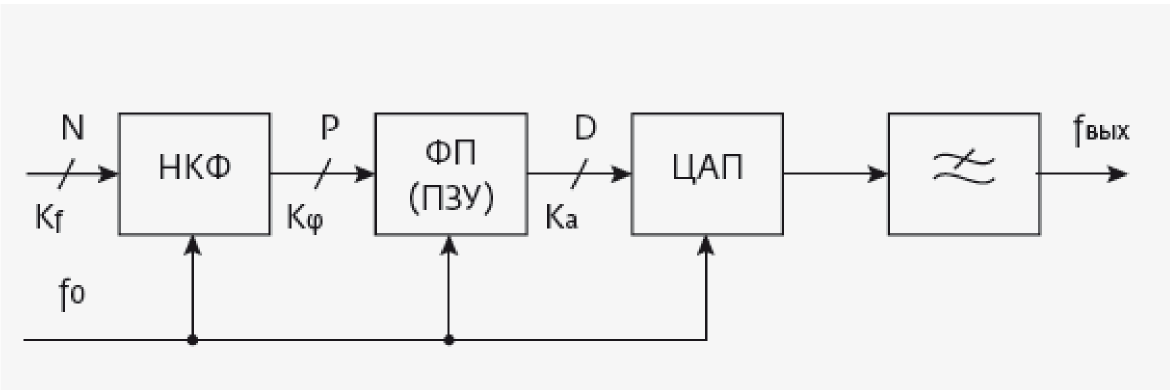


Рисунок 1.3 Функциональная схема синтезатора частот

Под аппроксимацией в контексте, решаемых задач, понимаем задачу нахождения для функции $f(x)$ более простой функции $P(x)$ из заданного класса, близкой в определенном смысле к $f(x)$. В задачах аналитического приближения функций широко применяются интерполяционные формулы, дающие приближенное выражение функции $f(x)$ посредством интерполяционного многочлена $P_n(x)$ (Ньютона, Лагранжа, Гаусса, Бесселя и др.) степени n. Во второй половине XX-го столетия существенное развитие получили приближения на основе сплайн-функций [14].

Теория аппроксимации в современном виде основана П.Л. Чебышевым. В настоящее время развивается теория наилучшего приближения функций алгебраическими или тригонометрическими полиномами. Большой вклад в развитие теории полиномов наилучшего приближения внес С.Н. Бернштейн и его ученики. Наряду с теоретическими достижениями в этой области получены и хорошие практические результаты, в том числе и в области РТС [15].

Полиномиальный метод аппроксимации Чебышева используется во многих научных и прикладных технических задачах: от приближения стандартных математических функций в современных специализированных микропроцессорах до реализации градуировочных характеристик при воспроизведении рабочих эталонов, калибровке датчиков и измерительных систем. Так, например, трудоемкие операции получения эталонных значений априорно неизвестных функциональных зависимостей чаще всего завершаются аппроксимацией с заданной точностью полиномами различной степени [16,17]. Повсеместное распространение полиномиального метода обусловлено его простотой, наглядной геометрической интерпретацией, а главное – низкими вычислительными затратами при расчете значений функции $f(x)$ с помощью полинома

$$L_n\left(x\right) = a_0 + a_1x + \ldots + a_nx^n = \sum_{i=1}^{n} a_ix^n$$

При использовании схемы Горнера полином можем записать в виде

$$L_n\left(x\right) = a_0 + a_1x + \ldots + a_nx^n = \left(\left(\left(a_nx + a_{n-1}\right)x + a_{n-2}\right)x + \ldots a_1\right)x + a_0 \qquad (1.1)$$

Вычислительная сложность расчета полинома степени n составляет $2n$ операций: n умножений и n сложений ($n(\times)$ + $n(+)$). Кроме того в памяти необходимо хранить n+1 констант a_0, …, a_{n+1} и запрашивать их для реализации вычисления полинома.

Ограничение количества операций при воспроизведении функции $f(x)$ достигается сокращением членов ряда, начиная с n+1 степени, исходя из оценки заданного максимального значения погрешности метода δ_{MM} в полиноме наилучшего приближения Чебышева:

$$\delta_{MM} = f(x) - L_n(x) \le f^{n+1}(x)(b-a)^{n+1} \Big/ \Big[(n+1)!2^{2n+1}\Big], \qquad (1.2)$$

где $f^{n+1}(x)$ – производная (n+1)-го порядка на интервале аппроксимации b–a, b – конец интервала.

В соответствии с (1.2) повышение точности может быть достигнуто как уменьшением интервала аппроксимации вычисления, так и повышением степени полинома. При использовании полиномиальных алгоритмов приближения наблюдается возможность сокращения объёма требуемой памяти для хранения алгоритма, за счёт уменьшения требуемой памяти под хранение констант, путём урезания их разрядных сеток до значения, не оказывающего влияние на погрешность результата.

При урезании разрядных сеток a_i полинома, с k десятичных цифр после запятой до $k - l$ цифр, где $l < k$, аппроксимирующий полином $P_p(x)$ приобретает вид

$$P_p(x) = a_{0p} + a_{1p} \cdot x + a_{2p} \cdot x^2 + ... + a_{np} \cdot x^n,$$

где $a_{0p}, a_{1p}, a_{2p}, a_{np}$ – коэффициенты реального полинома с ограниченными разрядными сетками. Поэтому возникает дополнительная погрешность $\delta_к$, обусловленная квантованием констант полинома с усечением разрядных сеток операндов

$$\delta_к = \Delta a_{0p} + \Delta a_{1p} x + \Delta a_{2p} x^2 + ... + \Delta a_{np} x^n,$$

где $\Delta a_{0p}, ..., \Delta a_{np}$ – погрешности усечения констант полинома в реальных условиях ($\Delta a_{0p} = a_0 - a_{0p}$).

Необходимо рассмотреть возможность компенсации погрешности метода, вызванной урезанием разрядных сеток констант, путём варьирования значениями констант. Также решению этой задачи будет способствовать разработка автоматического алгоритма взаимной компенсации составляющих погрешностей метода и задания констант.

При регистрации и измерениях квадратурных составляющих A и B сигналов со случайной начальной фазой используется выражение $R = \sqrt{A^2 + B^2}$, когда определение амплитуды выходного сигнала R не будет зависеть от фазы. Прямое вычисление амплитуды сигнала R сопряжено с расчетом «тяжелых» для процессора двух операций возведения в квадрат и взятия корня. В ИИС вычисление R должно производиться в реальном масштабе времени, поэтому поиск более простых алгоритмов расчета R является весьма актуальной задачей в современных ИИС [18].

Поскольку расчет R осуществляется в цифровом виде, то для преобразования аналоговых сигналов в цифровые широко используется ряд 4 … 8 … 24 разрядных аналого-цифровых преобразователей (АЦП) с шагом изменения порядка двух двоичных разрядов. Такие преобразователи вносят в суммарную погрешность результата измерений свою составляющую $\delta_{АЦП}$ порядка 8% $\ldots 10^{-4}$%. В этих условиях в специализированных вычислителях при различных условиях применения и, как следствие, различных значениях погрешностей вычислений следует использовать наиболее рациональные методы и вычислительные алгоритмы с целью сокращения программно-аппаратных затрат и времени вычислений.

1.3 Пути формирования тренажной информации о движении воздушных объектов в РТС

При тренировке операторов и контроле работоспособности ИИС автоматически назначаются и непрерывно формируются трассы движения ВО, состоящие в ручном вводе координат опорных точек, образующих очерчивающую траекторию движения в виде ломаной линии, с указанием скоростей полета в этих точках. От ИИС требуется при последовательном накоплении массива информации об опорных точках автоматически формировать уравнения движения ВО по трем координатам $x(t_p)$, $y(t_p)$, $z(t_p)$ и скорости $V(t)$.

Формирование траектории движения в виде отрезков прямых, соединяющих заданные точки и постоянной скорости движения ВО между соседними точками является наиболее простым и интуитивно понятным способом формирования траектории. Однако при этом при переходе с одного отрезка траекторий на другой происходит резкое изменение курса и скорости ВО, что не соответствует реальным траекториям, кроме этого, подобная модель не может формировать ускоренное движение ВО.

В связи с этим необходимо, чтобы процесс формирования траектории обеспечивал задание сложных вариантов воспроизведения траектории с гибкой, постепенной деформацией формы и кривизны. Траектория движения объекта должна образовываться только набором чередующихся круговых, параболических и прямолинейных сегментов. Она должна обладать нулевым кручением и, таким образом, должна быть более адаптирована для имитации плоских маневров ВО. Движение вдоль каждого сегмента в общем случае должно быть равноускоренным либо равномерным. Оператор задает лишь значения мгновенных скоростей в каждой узловой точке опорной ломаной линии, которая интерпретируется как скорость в точке выхода объекта на переходную кривую. Требуется обеспечить точное аналитическое воспроизведение траектории в параметрической функции времени только для сегментов в виде отрезков прямых и дуг окружностей.

Реальная имитация траектории движения должна учитывать кинематику маневра ВО, когда объект (его пилот) испытывает мгновенное воздействие перегрузки вследствие скачка центробежной силы $F_ц = m\upsilon^2/R$, (m и υ – соответственно масса и скорость ВО), который неизбежно возникает при переходе с траектории движения по прямой с радиусом кривизны $R \to \infty$ на траекторию движения с конечным радиусом $R_{окр}$. Также, имитируемая траектория должная формировать координаты движения ВО в трехмерном пространстве.

Для реализации отмеченных требований можно использовать способ описания кривой Безье, задаваемой графически и аналитически с наглядным, интуитивным представлением отдельных сегментов ее воспроизведения [15]. При этом параметрическое задание уравнений кривой по прямоугольным координатам формируется как функция безразмерного параметра $t \in [0;1]$. Но способ задает лишь геометрическую форму кривой и не обеспечивает физическое воспроизведение траектории движения ВО с учетом его кинематики - скоростей, ускорений, непосредственно привязанных к каждой точке всей траектории в заданные моменты времени. Кроме этого безразмерный параметр $t \in [0;1]$ не привязан к реальному времени t_p движения по траектории. Для использования этого метода в реальных системах имитации необходимо обеспечить контроль и корректировку динамических параметров движения в виде перегрузок, превышающих заданные предельные значения, например, для летчика больше 8-ми ускорений свободного падения $g = 9{,}8\ \text{м}/\text{с}^2$. А также, необходимо реализовать комплексирование сложной траектории из разнообразных сопрягаемых участков.

Предложены методы реализации воспроизведения траекторий движения объектов с помощью кубических кривых Безье, приведенные в работах [19,20]. Данным работам присущи недостатки. При комплексировании по полной траектории движения с помощью сегментов, выполненных в виде кривых Безье 3-го и 1-го порядков, не обеспечен плавный переход для широко распространенного сопряжения от кубической кривой Безье к отрезку прямой и обратно, поскольку было принято во внимание равенство только первых производных при сопряжении без учета второй производной. И в то же время необходимо производить переход с одного сегмента кривой Безье на другой с плавным изменением радиуса кривизны, когда для сопрягаемых сегментов из кубической параболы и отрезка прямой или двух кубических парабол соответственно их пять смежных вершин лежат на одной прямой или составляют выпуклый многоугольник. В указанных работах при расчете кинематики движения функция скорости задавалась через пройденный путь *S*, что вело к усложнению расчетов. Однако, более наглядным является задание функции скорости от времени при аппроксимации значения пути в функции безразмерного аргумента кривой Безье при помощи полиномов различной степени, в которых в качестве узлов интерполяции использовались только начальные и конечные интервальные значения и их производные, что приводило к большим значениям погрешности примерно в середине интервала аппроксимации [21-23].

При вычислении прямолинейных сегментах траектории, минимального радиуса кривизны траектории с номинальным (комфортным) уровнем перегрузки, необходимо рассчитать коэффициент $K\left(\phi_n\right)$.

$$K\left(\phi_n\right)=\begin{cases}\dfrac{1,017456}{\sqrt{tg\dfrac{\phi_n}{2}}}\cdot\left(1+\dfrac{1}{3\cdot tg^2\dfrac{\phi_n}{2}}\right) & \text{при} \quad \phi_n \geq \dfrac{\pi}{2} \\ \left(0,272166+\dfrac{1,014303}{tg\dfrac{\phi_n}{2}}\right) & \text{при} \quad \phi_n < \dfrac{\pi}{2}\end{cases} \tag{1.3}$$

Как видно из (1.3) здесь необходимо выполнять и вычисление тангенса, извлекать квадратный корень.

Для круговых сегментов траектории, при расчёте скорости, возникает необходимость дополнительно вычислять синус, косинус

$$\vartheta(t)=\frac{\left(\varepsilon\cdot\left(t-T_н\right)+\omega\right)^2}{\sqrt{x^2+y^2+z^2}}\cdot\left[\cos\left(\psi(t)\right)\cdot\left(x\cdot A_x+y\cdot A_y+z\cdot A_z\right)-\sin\left(\psi(t)\right)\cdot\left(x\cdot B_x+y\cdot B_y+z\cdot B_z\right)\right]$$

Для параболических сегментов, при вычислении максимального перемещения по параболе появляется необходимость вычислять арксинус в выражении

$$\psi_m=2\cdot\arcsin\left[3\cdot tg\frac{\phi_n}{2}\cdot\sqrt{\frac{tg^2\dfrac{\phi_n}{2}+\dfrac{1}{9}}{\left(tg^2\dfrac{\phi_n}{2}-\dfrac{7}{72}\right)^2+36\cdot tg^2\dfrac{\phi_n}{2}\cdot\left(tg^2\dfrac{\phi_n}{2}+\dfrac{31}{144}\right)^2}}\right] \tag{1.4}$$

Таким образом очевидно, что для оптимизации вычислительных процессов при имитации траекторий движения ВО необходимо рассмотреть способы наилучшей, с точки зрения точностных и программно-аппаратных затрат, реализации вычисления именно этих математических функций. А именно синус, косинус, тангенс, арксинус, арктангенс, корень квадратный.

От того, насколько эффективно реализуются вычислительные алгоритмы, зависит эффективность работы вычислительных устройств в целом. В свою очередь все вычислительные процессы строятся на элементарных математических функциях и реализуются алгоритмически за счёт предоставляемых АЛУ микропроцессоров набором стандартных операций, реализуемых в жесткой логике, как сложение, вычитание, умножение и т.п.

1.4 Направления практической реализации разрабатываемых алгоритмов

Одной из тенденций построения вычислительных систем реального времени является создание интегрированной вычислительной среды на базе ПЛИС. Структура ПЛИС позволяет гибко адаптироваться к заданному алгоритму обработки информации, причём сама адаптация может выполняться непосредственно во время работы устройства. Однако при разработке на ПЛИС различных систем обработки информации часто встаёт вопрос о вычислении значений различных функциональных зависимостей, как тригонометрических, извлечения корня квадратного и других. К сожалению, с помощью стандартных библиотек таких популярных САПР радиоэлектронной аппаратуры как MAX PLUS II и QUARTUS II нельзя решить эту задачу. Этому есть свои объективные причины. Во-первых, существует достаточно большое количество алгоритмов, отличающихся как по точности и скорости вычисления, так и по аппаратным затратам на их практическую реализацию. Во-вторых, для каждого алгоритма существует некоторый оптимальный, с точки зрения точности и сложности реализации, диапазон изменения аргумента. Выбор того или иного алгоритма всегда индивидуален и в конечном счёте определяется конкретными требованиями к разрабатываемой системе.

Рассмотренные структуры измерительных систем управления (контроля и диагностирования), осуществляющих сбор и обработку информации в различных конфигурациях чаще всего целесообразно реализовать с помощью ЦВМ, специализированных процессоров обработки информации, универсальных модулей сопряжения с датчиками сигналов и исполнительных устройств.

Перспективным считается подход, состоящий в органичном сочетании универсальных и специализированных вычислительных средств. В то же время разработка специализированных процессоров технически и экономически обоснована при их достаточно широком применении. Поэтому поставим задачу расширения областей применения разрабатываемых методов повышения точностных характеристик и быстродействия вычислительных устройств радиотехнических систем и на другие системы, например, технологические, решающие аналогичные задачи.

Необходимо исследовать численные методы решения вычислительных задач с учётом новой элементной базы, программного обеспечения, разработать алгоритмы, обладающие более широкими функциональными возможностями и отвечающие требованиям эффективной реализации. Эффективная реализация предполагает повышение производительности, точностных характеристик, контролепригодности вычислительных устройств систем управления и контроля при минимальных программно - аппаратурных затратах, поскольку мобильные передвижные РТС проектируются с ограничениями на массогабаритные характеристики и стоимость.

На основе анализа и исследования современных методов компьютерной математики с представлением сложных функциональных зависимостей суперпозицией более простых функций, предлагается сделать ряд разработок, обеспечивающих поиск и выработку более оптимальных наборов решений для типовых вычислительных задач. Совершенствование методов аппроксимации и интерполяции планируется производить с использованием итерационных рекурсивных методов, сплайн - аппроксимации, кривых Безье, гибридных алгоритмов; предполагается обобщить и усовершенствовать современные концепции и алгоритмы. В исследованиях, путем компьютерного моделирования, проводятся серии экспериментов на основе вновь разработанных собственных специализированных приложений. Решение поставленной задачи будет осуществляться с использованием методов вычислительной математики и математического моделирования на ЭВМ. С этой целью разрабатываются программы многомерной оптимизации полиномов наилучшего приближения с использованием сред программирования Delphi, Builder, C++, MathCAD, LabView и т.д.

Выводы к главе 1:

1. Поставлена актуальная задача существенного упрощения реализации сложных функциональных зависимостей путем их представления в виде суперпозиции более простых функций и разработка оптимизированных полиномиальных преобразований Чебышева.

2. На основе отечественных и зарубежных научно-технических публикаций проведен анализ, который свидетельствует о необходимости моделирования реализуемого в ИИС вычислительного процесса, что позволит обеспечить сокращение вычислительных затрат, значительное уменьшение погрешности результата за счёт взаимного поглощения и взаимной компенсации фактически коррелированных, детерминированных составляющих погрешностей, которые имеют одну физическую природу.

3. Требует решения задача оптимизации структуры вычислителя под решаемые задачи, улучшения принципов построения структур специализированных вычислительных систем высокой производительности, архитектура которых основывается на гетерогенных принципах и объединяет в едином вычислительном ресурсе как универсальном микропроцессоре, а также ПЛИС и других специализированных устройствах, ориентированных на наиболее рациональную реализацию конкретной вычислительной задачи с устранением избыточной точности.

4. Необходимо разработать метод наглядного воспроизведения траекторий ВО из плавно сопрягаемых сегментов в системе координат зоны обзора ИИС и преобразованием безразмерного аргумента параметрических уравнений кривых Безье при различных законах изменений линейной скорости в аргумент реального

линейно–нарастающего интервала времени воспроизведения траектории с обеспечением контроля перегрузок в трехмерном пространстве.

Сведем недостатки существующих алгоритмов и направления разработки новых алгоритмов в таблицу 1.1.

Таблица 1.1 Сравнительная характеристика существующих подходов и вновь разрабатываемых алгоритмов.

Решаемые задачи	Недостатки существующих	Отличие разрабатываемых
Методы воспроизведения функциональных зависимостей	1. Сложные функциональные зависимости. 2. Отсутствие адаптации под современные аппаратные средства. 3.Большой объем памяти для хранения констант.	1. Комбинация простых зависимостей с изменяемой под аппаратурную реализацию точностью. 2. Компенсация различных погрешностей. 3. Разработка метода под конкретный специализированный вычислитель.
Формирование траекторий	1. Моделируют не все виды движений ВО. 2. Не обеспечивают плавность траекторий. 3. Не обеспечивают контроль параметров движения.	1. Обеспечивают моделирование равномерного, неравномерного и равноускоренного движений ВО. 2. Обеспечивают контроль скорости, ускорений и максимальных перегрузок. 3. Формирует траекторию движения ВО в виде плавного сопряжения сегментов на основе параметрических уравнений Безье.
Программное обеспечение, реализующее методы воспроизведения функциональных зависимостей и моделирование траекторий движения ВО.	1. Сложность реализации существующих подходов. 2. Отсутствие адаптации под современные специализированные вычислители. 3. Отсутствует возможность изменять параметры, методы, алгоритмы обработки.	1. Простота применяемых алгоритмов воспроизведения функциональных зависимостей и моделирования траекторий движения ВО. 2. Ориентация на используемые вычислительные средства обработки информации. 3. Широкий спектр изменяемых методов, параметров и т.д.

Глава 2 Разработка методов поиска полиномов наилучшего приближения при аппроксимации функциональных зависимостей

Математический аппарат интерполирования различных функциональных зависимостей с помощью многочленов является важнейшим аппаратом численного анализа для повышения эффективности, как проектирования, так и эксплуатации вычислительных и измерительных систем. Наибольшее применение для приближения стандартных функций, калибровки интеллектуальных датчиков, воспроизведения их функций преобразования нашли полиномы Чебышева [24].

Целью данной главы является разработка алгоритмов поиска полиномов наилучшего приближения в соответствии со стратегией максимальной идентичности графиков аппроксимируемой функции и приближающего полинома.

2.1 Совершенствование методов поиска полиномов наилучшего приближения

В отличие от классического чебышевского альтернанса, когда в (1.1) для полного многочлена степени n с n+1 членами имеется n+1 узел аппроксимации, в котором значения функции и интерполянта совпадают, сократим число вычислительных операций A и обращений к постоянному запоминающему устройству m, уменьшив число узлов интерполяции и исключив отдельные константы a_i, члены ряда $a_i x^i$ или проведя их специальную группировку. При этом увеличение погрешностей метода δ_{MM} по сравнению с погрешностями для полинома степени n, у которого n+1 член, должно быть незначительным. В то же время обеспечим при m оставшихся константах a_i и m+1 узлах аппроксимации, как и в полиноме наилучшего приближения, принцип чередования не менее чем m+1 раз знака погрешности δ_{MM} воспроизводимой полиномом функции, с наименьшим отклонением ее от нуля [16].

Ограничение количества операций при воспроизведении функции $f(x)$ достигается также сокращением числа членов ряда в (1.1), начиная с n+1 степени, исходя из оценки заданного максимального значения погрешности метода δ_{MM} в полиноме наилучшего приближения Чебышева (1.2). При переходе от полинома Чебышева степени n с погрешностью $\delta_{nч}(x)$ к полиному степени n+1 с погрешностью $\delta_{nч}(x)$ в соответствии с (1.2) отношение погрешностей составит

$$\delta_{nч}(x) \big/ \delta_{(n+1)ч}(x) = f^{[n+1]}(x) \cdot 4(n+2) \big/ [f^{[n+2]}(x) \cdot (b-a)] \tag{2.1}$$

Если на первом этапе поиска оптимального алгоритма воспроизведения функции исключить отношение производных $f^{n+1}(x)$, $f^{n+2}(x)$ в (1.1) при увеличении степени многочлена n, то для повышения точности, например, при вычислении $\sin x$, эффективно повышать степень полинома, т. е. при n=4; 5; 6 в соответствии с (2.1) отношение погрешностей $\delta_{4ч}(x)/\delta_{5ч}(x)$ и $\delta_{5ч}(x)/\delta_{6ч}(x)$ составит соответственно 28 и 32. Увеличение степени полинома на единицу обеспечивает примерно пятизначное приращение M-двоичных цифр результата при увеличении вычислительных операций N на 2 и обращении к памяти P за дополнительной константой на 1.

В соответствии с (1.1) эффективным альтернативным средством повышения точности может быть и уменьшение интервала аппроксимации b–a в (1.2) с разбивкой его на подинтервалы равной или неравной длины [24]. При уменьшении интервала b–a в k раз погрешность должна уменьшиться в k^{n+1} раз, а при n=4, k=2 можно получить ее уменьшение в 2^5 раз. С введением двух подинтервалов аппроксимации количество операций вычисления полинома возрастет на 2 из-за необходимости определения подинтервала нахождения аргумента: извлечения константы фиксации границ подинтервалов и сравнения текущего аргумента функции с этим значением. Имеются и другие потенциальные способы более рационального применения полиномиальных методов, связанные с исключением и (или) более сложной перегруппировкой членов степенного ряда в (1.1) [16].

Упрощенный алгоритм поиска оптимального полинома для указанных условий (рис. 2.1) после задания начальных данных в виде требуемой точности приближения включает этапы:

- приближенное определение полинома с наименьшей степенью n, обеспечивающего заданную максимальную погрешность δ_{MM} воспроизведения функции и расчет констант a_i;

- исключение неэффективных, наименее влияющих на погрешность δ_{MM} членов ряда $a_i x^i$, констант a_i (например, когда заданы $a_0 = 0$, $a_1 = 1$ и т. д. и используют только четные (нечетные) члены ряда с полной или частичной группировкой по схеме Горнера) в соответствии с максимальной идентичностью графиков функции $f(x)$ или преобразовательной характеристики ИИС и приближающего полинома $L_n(x)$;

- поиск усеченного полинома наилучшего приближения $L_n(x)$ с $m < (n+1)$ оставшимися константами a_i, в котором на интервале аппроксимации получается по крайней мере $m+1$ точек, в которых погрешности δ_{MM} принимают равные максимальные значения $+(\delta_{MM} \pm \Delta\delta_{MM})$ и $-(\delta_{MM} \pm \Delta\delta_{MM})$ с учетом неустранимых погрешностей $\Delta\delta_{м}$ (см. рис. 2.1);

- уточнение степени полинома n и количества констант m в соответствии с заданными погрешностями результата δ_p и полученными значениями погрешностей δ_{MM} сохранение этой степени или увеличение (уменьшение) ее на единицу, уточнение числа констант a_i для получения заданной погрешности результата δ_p с исключением невостребованной избыточной точности определения преобразовательной характеристики или функции $f(x)$ полиномом $L_n(x)$ с достаточным дискретным числом значащих цифр представления результата M (чтобы погрешности квантования констант на данном этапе не влияли на погрешность результата) и соответствующим ему минимальным числом вычислительных операций M и (или) суммарного числа операций $M+P$ воспроизведения полинома;

- выполнение на заключительном этапе усечения, симметрирования, взаимопоглощения составляющих погрешности δ_p на выходе системы в виде суммы погрешностей $\delta_м$, дискретного квантованного представления данных δ_{a_i} набора констант a_i, измеряемой физической величины x, промежуточных преобразований $\delta_{пр}$ с ограниченным числом разрядов в форматах представления данных специализированного вычислительного комплекса (см. рис. 2.1) [16,17].

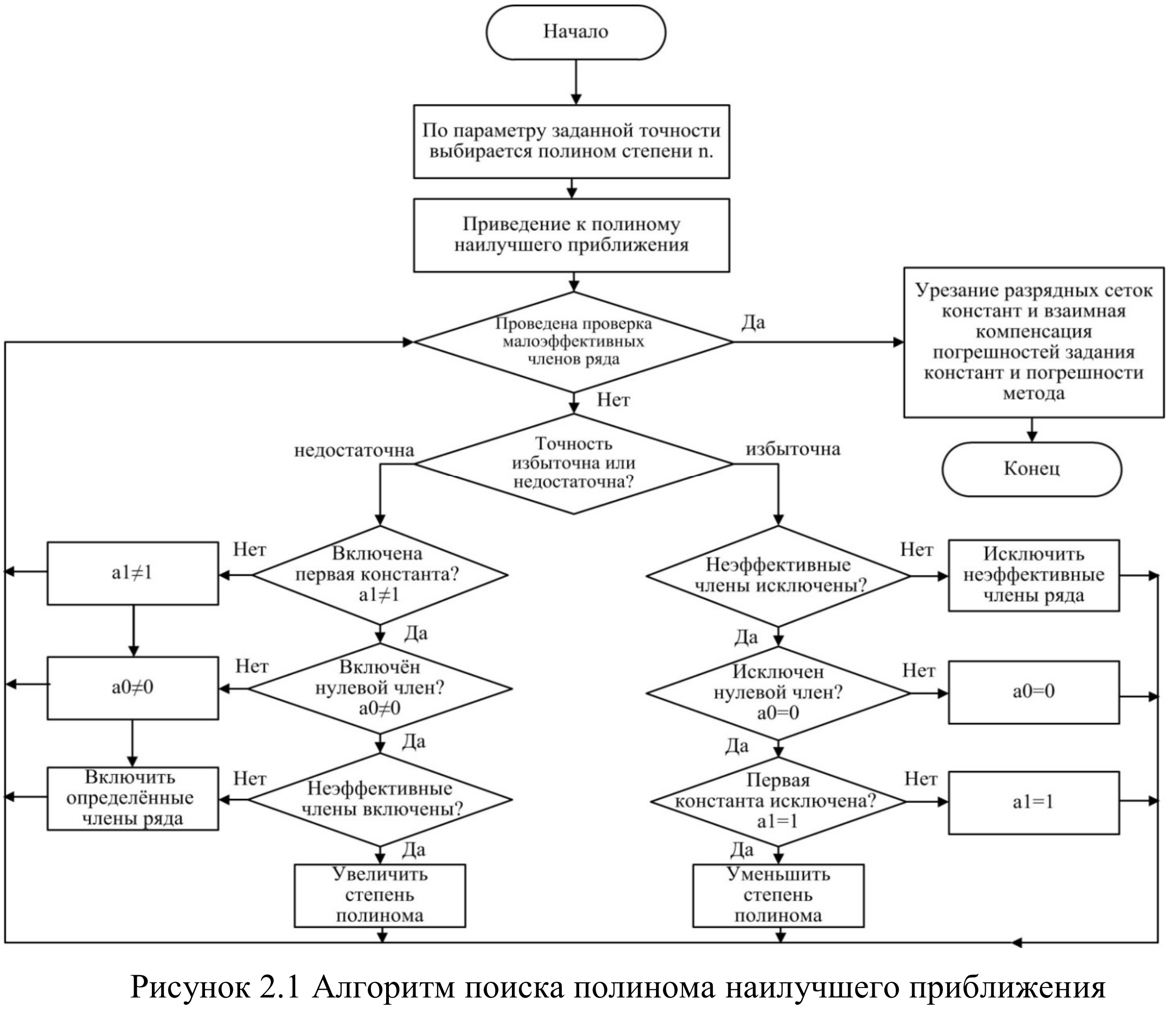


Рисунок 2.1 Алгоритм поиска полинома наилучшего приближения

При уменьшении количества констант полинома требуется меньше вычислительных операций, узлов аппроксимации и физических эталонов при калибровке измерительного канала, чем в (1.1).

В соответствии с рис 2.1 рассмотрим методы поиска полиномов наилучшего приближения. Этот поиск для полинома степени n представляет собой многомерную задачу, поскольку для полинома наилучшего приближения степени n надо найти $n+1$ узлов аппроксимации. Теоретические и практические аспекты одномерной и многомерной оптимизации рассмотрены в [2-4].

При поиске полиномов наилучшего приближения стандартных функций практически не являются критичными временные затраты и количество эталонных значений функции в узлах аппроксимации [24]. В тоже время при проведении калибровки измерительных систем количество экспериментальных данных ограничено, также критично и время калибровки. При поиске оптимальных узлов аппроксимации полинома Чебышева используются различные итерационные методы. На данный момент не существует универсальных методов поиска полиномов наилучшего приближения. Общим недостатком известных методов является отсутствие универсальной теоретической базы, в соответствии с которой можно было бы выбирать оптимальное начальное приближение ($n+1$ узлов аппроксимации), последующие критерии быстрого сужения подинтервалов поиска, непрерывного уменьшением дискрета их изменения, расчет максимального времени поиска полинома наилучшего приближения. В тоже время в технических приложениях наибольшее применение нашли полиномы первой, второй и третьей степени. Для таких полиномов при наличии нулевого члена число узлов аппроксимации соответственно равно 2, 3 и 4. В общем случае, в отличие от классического чебышевского альтернанса предполагают, что в (1.1) для полного многочлена степени n с $n+1$ членами имеется $n+1$ узлов аппроксимации, в которых значения функции и интерполянта совпадают. Однако в ряде случаев можно уменьшить число узлов аппроксимации и исключить отдельные константы a_i, члены ряда $a_i x^i$ или произвести их специальную группировку [25, 26]. Уменьшение числа узлов аппроксимации значительно упрощает методы поиска полиномов наилучшего приближения, делает этот процесс более детерминированным и позволяет выявить общие закономерности, которые могут быть эффективно использованы и для упрощения трудоемкого поиска полиномов с более высокими степенями.

2.2 Быстродействующие алгоритмы поиска полиномов наилучшего приближения для воспроизведения функциональных зависимостей

2.2.1 Методы поиска полиномов наилучшего приближения первой степени. Рассмотрим методы поиска полиномов наилучшего приближения первой степени в соответствии с методами одномерной оптимизации. Принцип получения полинома наилучшего приближения, например, для функции sin(x) следующий: выбираются крайние точки начала и конца интервала аппроксимируемой функции a=0 и b=1,571, через эти точки проводится прямая, которая соответствует полиному Ньютона аппроксимирующему эту функцию с коэффициентом $a_1 = \frac{f(b) - f(a)}{b - a}$. При a_0=0 получим полином $L_H(x) = 0,6366x$ с максимальным значением погрешности метода аппроксимации, определяемым отрезком cc' (рис. 2.2 а). Для монотонно изменяющейся на заданном интервале функции sin(x) значение максимальной абсолютной погрешности метода аппроксимации для полинома $L_H(x) = 0,6366x$ достигается в некоторой точке x_M.

Чтобы получить полином наилучшего приближения на основе полинома Ньютона $L_H(x) = 0,6366x$ и необходимо определить расстояние между точками c и c' (рис.2.2 а)) и в полином Ньютона ввести коэффициент a_0, значение которого равно половине отрезка (c,c'). Для графического определения точки x_M с максимальным значением погрешности целесообразно использовать быстродействующие методы последовательного одномерного поиска экстремума: половинного деления, золотого сечения [26]. При использовании метода половинного деления подинтервал нахождения точки x_M погрешности на каждой итерации сокращается в два раза. Этот метод позволяет исключить из дальнейшего рассмотрения на каждой итерации в точности половину текущего интервала неопределенности. Алгоритм состоит в анализе величин функции в трех точках, которые равномерно распределены на текущем интервале. Число итераций будет определяться выражением

$$N = 1 + \mathrm{ent}\log_2 \frac{b-a}{\Delta x}, \qquad (2.2)$$

где: b-a – величина интервала интерполяции, Δx – некоторый минимальный дискрет изменения аргумента, определяемый максимальным приращением функции при заданном приращении аргумента и относительной погрешностью определения максимальной погрешности метода аппроксимации; ent – выделение целой части числа.

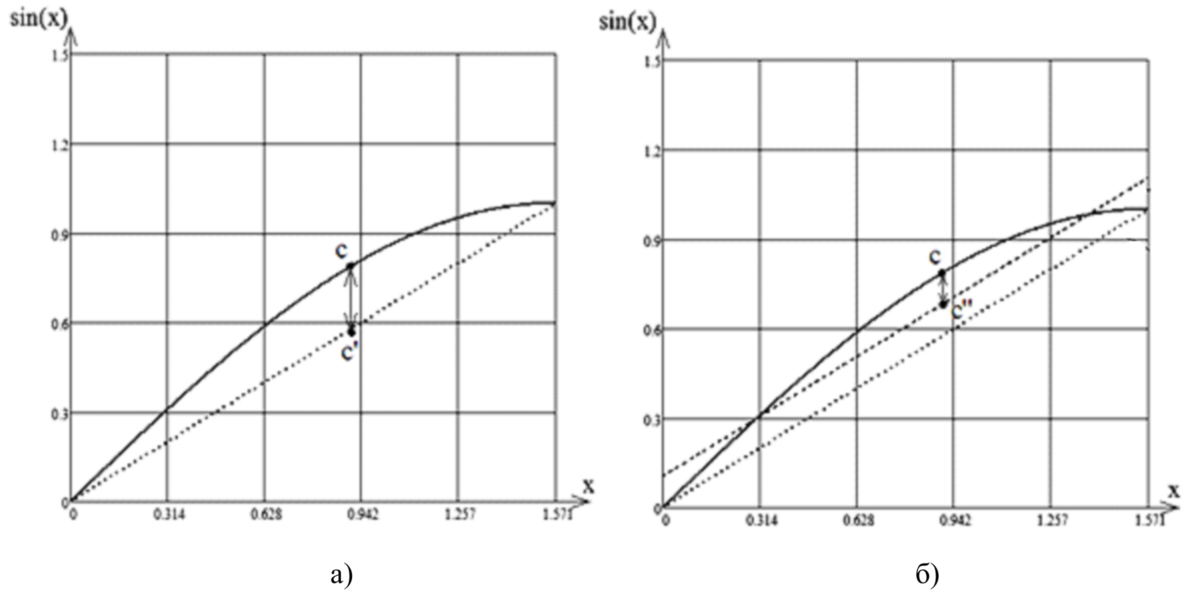


Рисунок 2.2 Совмещенные графики функции $\sin(x)$ и аппроксимирующих полиномов-пунктирная линия

Допустимо задавать относительную погрешность определения погрешности метода в пределах (1-5)%.

Как отмечено выше, за счет смещения аппроксимирующей прямой можно существенно повысить точность аппроксимации. Рассмотрим применение этого метода. Для калибровки датчиков в измерительных системах в ряде случаев недостаточно иметь только аппроксимирующую функцию. Необходимо знать и узлы аппроксимации (физические эталоны), значения аппроксимируемой функции, в которых будут использованы для уточнения коэффициентов в (2.2) при периодических калибровках измерительных каналов с нестабильными параметрами. Для решения поставленной задачи необходимо определить, где значения функции и приближающего полинома равны. Можно и одновременно целенаправленно искать оптимальные узлы аппроксимации и по ним определять коэффициенты в (1.1).

При $a_0 = 0$ по сравнению с полиномами Ньютона 1, 2 и 3-й степеней в (1.1) целесообразно соответственно примерно в 1,45, 1,24 и 1,09 раза уменьшить погрешность при $f^{n+1}(x) = \text{const}$ если последний узел аппроксимации перенести из конца интервала на некоторое расстояние внутрь интервала (рис. 2.3). В этом случае при вычислении полинома сокращаются две операции-сложения и выборки из памяти коэффициента a_0. Определение одного оптимального значения узла аппроксимации в функции одной переменной с минимальным значением погрешности полинома является задачей одномерной оптимизации.

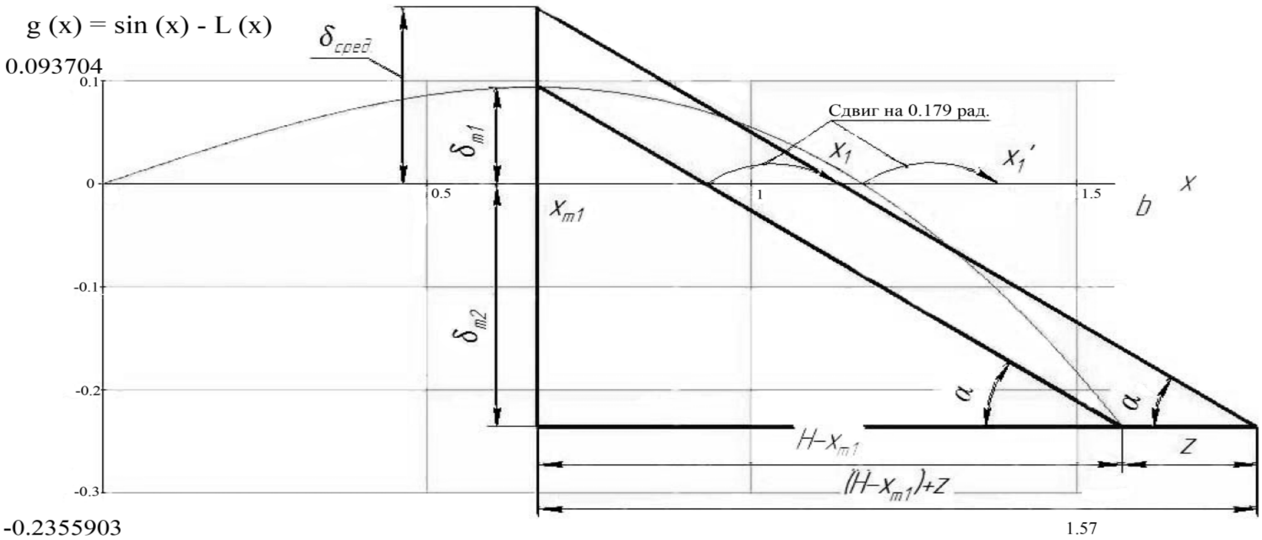


Рисунок 2.3 Расчет сдвига узла аппроксимации функции sin(x) по начальному приближению

Итерационные процессы, основанные на использовании фиксированного приращения шага изменения положения узлов аппроксимации для ряда функциональных зависимостей, не обеспечивают эффективный поиск полинома. В связи с этим разработан быстродействующий алгоритм, в котором шаг изменения положений аргумента между узлами аппроксимации на каждой итерации изменяется пропорционально скорости изменения текущего значения погрешности аппроксимации между ее соседними локальными экстремальными значениями [27]. Данные значения определяются в значительной степени расстоянием между этими прилегающими узлами аппроксимации. Рассмотрим алгоритм поиска полинома наилучшего приближения для этого случая при $f^{n+1}(x) \neq \text{const}$ на примере функции sin(x) на интервале аппроксимации H при $x \in [0;1,571]$ с одним начальным значением узла x_1 аппроксимации $0,75H = 0,75 \cdot 1,571 = 1,17$ рад. В этом случае для $L(x) = 0,787x$ получен график погрешности $g(x) = \sin(x) - L(x)$ аппроксимации функции sin(x) (рис. 2.3). Внутри подинтервала изменения аргумента $0 - 0,75H$ методом половинного деления определяется точка $x_{m1} = 0,657$ рад с локальным максимальным абсолютным значением погрешности $|\delta_{m1}| = 0,0937$. В конце интервала просто фиксируется второе локальное максимальное абсолютное значение погрешности $|\delta_{m2}| = 0,2356$. Следует отметить, что система MathCAD на рис. 2.3 автоматически фиксирует максимальные значения погрешностей. Для быстрого итерационного поиска узла аппроксимации полинома наилучшего приближения аппроксимируем текущее значение погрешности между ее максимальными значениями полиномом Ньютона первой степени с коэффициентом, определяемым тангенсом угла $tg\alpha = (|\delta_{m1}| + |\delta_{m2}|)/(H - x_{m1}) = 0,329/0,914 = 0,359$ в прямоугольном треугольнике. По вертикали величина катета треугольника равна сумме абсолютных значений локальных погрешностей $|\delta_{m1}| + |\delta_{m2}| = 0,0937 + 0,2356 = 0,3293$, другой катет равен $H - x_{m1} = 0,914$ рад. Величину смещения узла аппроксимации от значения $0,75 \cdot H = 1,17$ рад в сторону большего значения погрешности определим из того, что в соответствии с теоремой Чебышева, после первой и последующих итераций локальные максимальные абсолютные значения погрешностей δ_{m1} и δ_{m2} должны быть равны между собой по абсолютной величине и определяться средним значением абсолютных значений погрешностей как внутри интервала $[0;0,75H]$, так и на втором конце интервала. В этом случае получим среднее значение погрешности $\delta_{сред.} = (|\delta_{m1}| + |\delta_{m2}|)/2 = (0,0937 + 0,2356)/2 = 0,1646$. Таким образом, новая длина катета будет равна $\left|\delta_{m2}\right| + \left|\delta_{сред.}\right| = 0,235 + 0,165 = 0,4$. Из этого значения, в соответствии с $tg\alpha$, вычисляется второй катет $0,4 / 0,366 = 1,093$ и по нему определяется смещение узла аппроксимации x_1 в точку $x_1^{/}$ на величину $1,093 - 0,914 = 0,179$ рад (рис 2.3). По уточненному значению узла аппроксимации в точке $x_1^{/} = 1,35$ вычисляется новый полином $L(x) = 0,723 \cdot x$ и воспроизводится график погрешности после 1-й итерации. Среднее значение погрешности будет примерно равно $\delta_{сред.} = (\left|\delta_{m1}\right| + \left|\delta_{m2}\right|)/2 = (0,1394 + 0,1351) / 2 = 0,1373$, то есть практически соответствовать точному значению [27]. Поэтому в данном случае необходимо узел аппроксимации смещать влево на гораздо меньшую величину $0,862 - 0,8566 = 0,0054$ рад. В тоже время 2-ю итерацию можно и не проводить, поскольку уже и после 1-й итерации относительная погрешность определения максимального значения погрешности по сравнению с точным значением равна $(0,139 - 0,137) / 0,137 = 0,002 / 0,137 = 0,0145$ (не превышает 1,5%).

Указанный алгоритм был проверен для функции $\sqrt{x}$ с большим изменением производных на интервале аппроксимации $x \in [0\,;1]$. За 5 итераций получен полином наилучшего приближения $L(x) = 1,204 \cdot x$ с относительной погрешностью менее 2% относительно максимальной погрешности метода аппроксимации полиномов наилучшего приближения.

2.2.2 Методы поиска полиномов наилучшего приближения второй степени. Сведем сложную задачу многомерной оптимизации целевой функции с большим числом переменных к последовательному решению ряда простых задач одномерной оптимизации. Необходимо поставить условие, чтобы фиксация оптимального значения одной из переменных одновременно обеспечивала бы уменьшение диапазонов последующего поиска оптимальных значений оставшихся переменных. При этом стратегию уменьшения интервала неопределенности оптимального значения первой переменной возьмем такой же, как и в случае одномерной оптимизации в подразделе 2.2.1. Т.е. определим ее, исходя из скорости изменения погрешностей между соседними узлами аппроксимации.

Рассмотрим ускоренный метод поиска полинома наилучшего приближения второй степени на примере функции $\sqrt{x}$ с интервалом изменения $x \in [0;1]$, поскольку у этой функции значения производных на границах интервала аппроксимации значительно отличаются. В качестве первого приближения используем полином Ньютона, где узлы аппроксимации равноудалены друг от друга и соответствуют значениям аргумента: 0; 0,5; 1. Вычислив значения функции $\sqrt{x}$ в этих точках, составим и решим методом Гаусса систему уравнений:

$$\begin{cases} a_0 0 + a_1 0 + a_2 0^2 = \sqrt{0} \\ a_0 0{,}5 + a_1 0{,}5 + a_2 0{,}5^2 = \sqrt{0{,}5} \\ a_0 1 + a_1 1 + a_2 1^2 = \sqrt{1} \end{cases} \qquad (2.3)$$

Полученные коэффициенты a_0, a_1, a_2 подставим в (1.1) и получим аппроксимирующий полином $L_H = a_0 + 1{,}8284271x - 0{,}8284271x^2$. На рис. 2.4 представлены графики погрешности первого приближения для функции $\sqrt{x}$ и совмещенные графики функции и приближающего полинома. Максимальные погрешности δ_{m1} и δ_{m2} значительно отличаются друг от друга по абсолютной величине [16]. С этой целью разобьем решение задачи трехмерной оптимизации при поиске полинома наилучшего приближения на 2 этапа: сначала производится поиск только положения оптимального среднего узла x_1 с фиксацией двух других узлов на концах интервала аппроксимации, а затем с итерационным перемещением этих узлов внутрь интервала. Поиск узла аппроксимации x_1 производится таким же методом, как и поиск одного оптимального узла для полинома первой степени. Сравниваются значения максимальных локальных погрешностей ($|\delta_{m1}| = 0{,}1422$; $|\delta_{m2}| = 0{,}0393$) на подинтервалах [0; 0,5] и [0,5 1], затем производится расчет смещения узла аппроксимации x_1 в сторону большего значения погрешности на величину 0,19.

Величина смещения пропорциональна отношению абсолютного значения разности между средним значением максимальных абсолютных погрешностей $\delta_{сред.} = (|\delta_{m1}| + |\delta_{m2}|)/2 = (0,1421 + 0,0392)/2 = 0,0907$ и абсолютным значением любой из этих погрешностей, например, $|\delta_{m1}| = 0,1421$ к величине интервала аргумента между максимальными значениями погрешностей $x_{m2} - x_{m1} = 0,77 - 0,09 = 0,68$.

После первой, второй итераций относительная погрешность отклонения абсолютных максимальных погрешностей от их среднего значения составляет соответственно 5,8%, 2,6%. После третьей итерации для полинома $L(x) = 0 + 2,193x - 1,193x^2$ относительная погрешность отклонения между $|\delta_{m1}|$ и $|\delta_{m2}|$ станет меньше 1%.

После примерного определения среднего узла аппроксимации (точки равновесия) за 2-3 итерации производится определение примерного положения и двух крайних узлов [27]. При этом выравниваются по два абсолютных максимальных значения погрешностей в левой и правой частях интервала относительно друг друга без изменения положения среднего узла. Следует отметить, что определение положения среднего узла уже было в значительной степени предопределено и обеспечило сужение интервалов поиска двух других узлов.

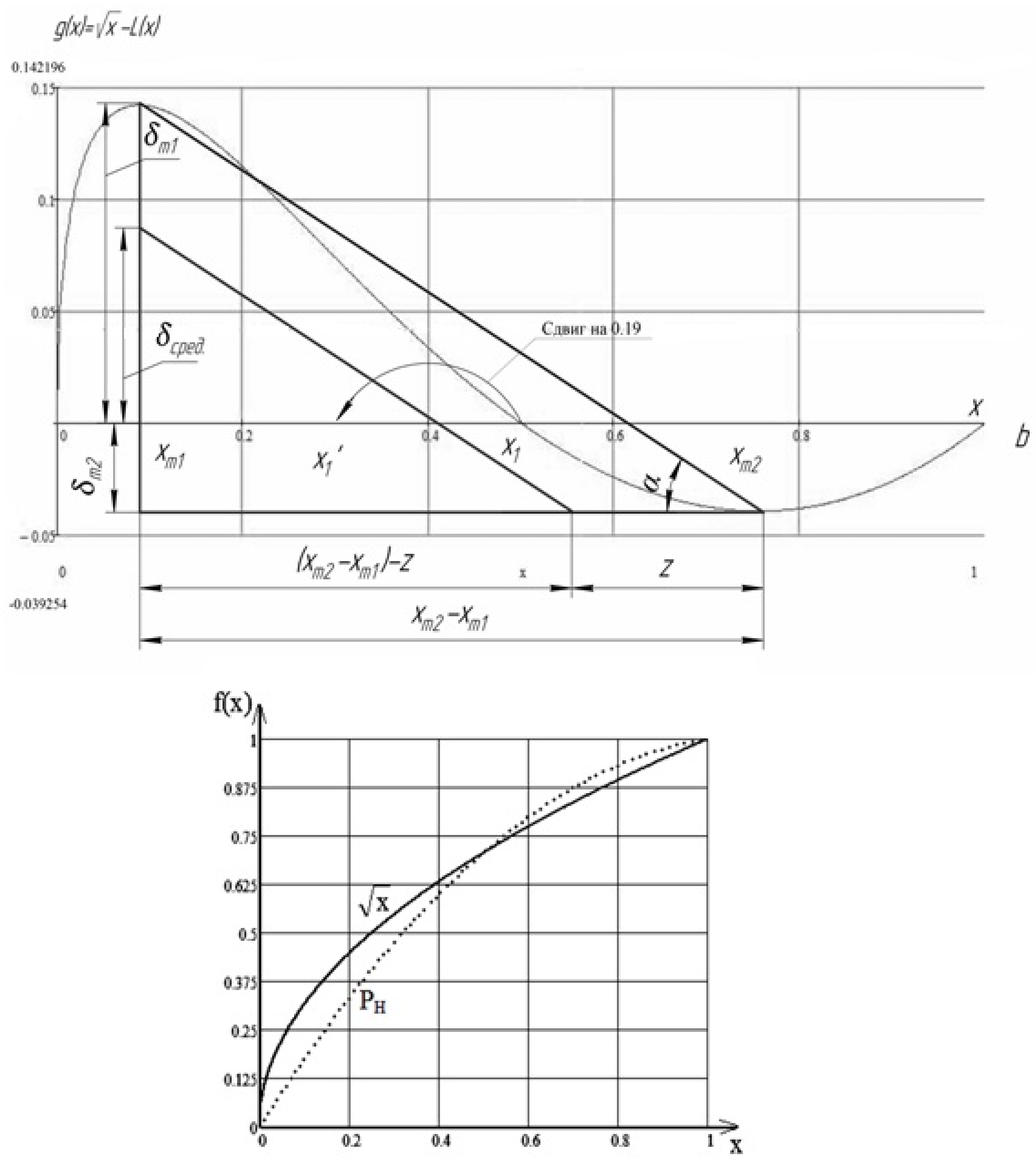


Рисунок 2.4 График погрешности полинома Ньютона и совмещенные графики функции и приближающего полинома

Так как изменение положения одного узла влечет за собой некоторое изменение всего графика аппроксимирующего полинома, в том числе и в противоположной части интервала, то для завершения итерационного процесса проверяется условие: все абсолютные значения максимальных погрешностей на подинтервалах не должны отличаться более чем на 1-5% от их среднего значения.

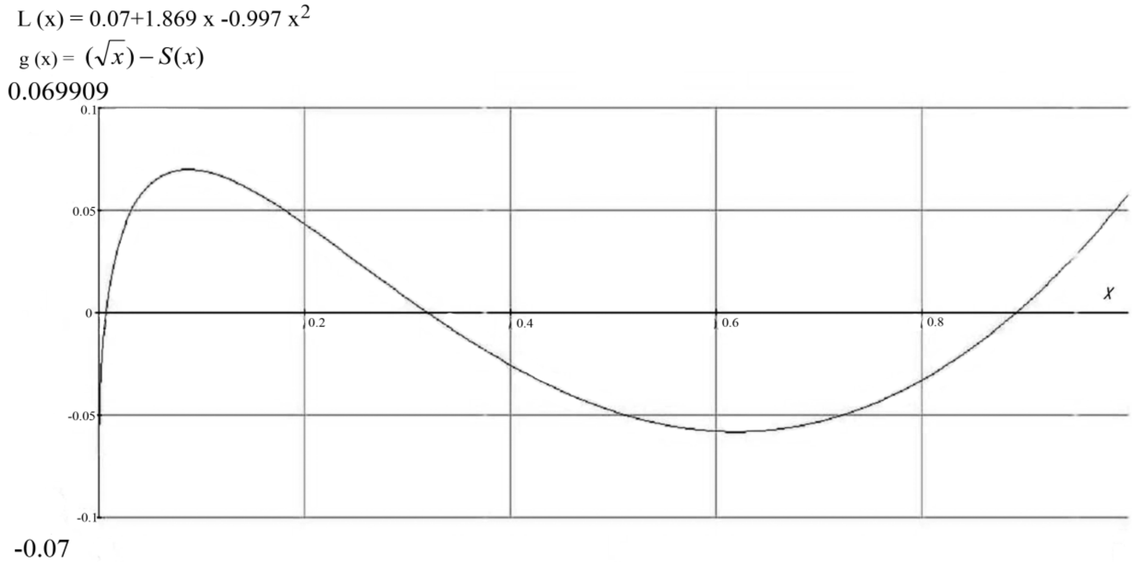


Рисунок 2.5 График погрешности полинома наилучшего приближения

Прикладное значение предложенных алгоритмов заключается в многократном сокращении числа итерационных циклов при калибровке измерительных каналов с нестабильными параметрами. Результат работы предложенного алгоритма – нахождение полинома наилучшего приближения для функции квадратного корня за 10–12 итераций (рис. 2.5), тогда как при использовании алгоритмов, описанных в литературе, затраты будут в 3 раза больше [28].

2.2.3 Поиск полиномов наилучшего приближения третьей и высших степеней. В общем случае полином степени n имеет вид

$$P(x) = a_0 + a_1 x + a_2 x^2 + ... + a_n x_n, \qquad (2.4)$$

где $a_0,...,a_n$ – коэффициенты.

Получение коэффициентов уравнения (2.4) осуществляется следующим образом: составляется система n+1 линейных алгебраических уравнений вида:

$$\begin{cases} P(x_0) = a_0 + a_1 x_0 + a_2 x_0^2 + ... + a_n x_0^n \\ P(x_1) = a_0 + a_1 x_1 + a_2 x_1^2 + ... + a_n x_1^n \\ ... \\ P(x_n) = a_0 + a_1 x_n + a_2 x_n^2 + ... + a_n x_n^n \end{cases} \qquad (2.5)$$

Затем одним из известных способов [4] решается система (2.5), и вычисляются значения коэффициентов $a_0,...,a_n$.

Рассмотрим получение полинома четвертой степени наилучшего приближения для функции $\sqrt{x}$ на интервале $x \in [0\,;1]$. Полинома четвертой степени имеет вид

$$P_H(x) = a_0 + a_1 x_0 + a_2 x^2 + a_3 x^3 + a_4 x^4 \tag{2.6}$$

Чтобы составить систему линейных алгебраических уравнений (2.5), необходимо задать пять узлов аппроксимации $x_0,...,x_4$. В этих узлах значения исходной и аппроксимируемой функцией должны совпадать. Для упрощения вычисления начального приближения возьмем случай, когда все узлы равноудалены друг от друга. Получаем значения: x_0 = 0; x_1 =0,25; x_2 = 0,5; x_3= 0,75; x_4 = 1. По этим узлам аппроксимации, построим систему линейных алгебраических уравнений вида

$$\begin{cases} \sqrt{0} = a_0 + a_1 0 + a_2 0^2 + a_{1n} 0^3 + a_n 0^4 \\ \sqrt{0,25} = a_0 + a_1 0,25 + a_2 0,25^2 + a_{1n} 0,25^3 + a_n 0,25^4 \\ \sqrt{0,5} = a_0 + a_1 0,5 + a_2 0,5^2 + a_{1n} 0,5^3 + a_n 0,5^4 \\ \sqrt{0,75} = a_0 + a_1 0,75 + a_2 0,75^2 + a_{1n} 0,75^3 + a_n 0,75^4 \\ \sqrt{1} = a_0 + a_1 1 + a_2 1^2 + a_{1n} 1^3 + a_n 1^4 \end{cases} \tag{2.7}$$

Для решения полученной системы (2.7) воспользуемся методом Гаусса [29]. Результатом решения этой системы являются коэффициенты функции

$$P_4 = 3,13352x \text{ - } 5,92483x^2 + 6,15356x^3 - 2,36225x^4 \tag{2.8}$$

На рис. 2.6 представлен график абсолютной погрешности этого полинома.

Задачей оптимизации является изменение положения точек аппроксимации x_0, ..., x_n таким образом, чтобы график погрешности полученного полинома стал постоянным по амплитуде. На рис. 2.7 представлена схематическая реализация данного подхода. Стрелками обозначены необходимые перемещения узлов аппроксимации для получения полинома наилучшего приближения. Точка x_0 смещается в точку x'_0 – точку максимума погрешности, x_1 – в x'_1 – точку минимума погрешности и так далее. На основе вновь полученных узлов составляется система (2.7), рассчитываются коэффициенты $a_0,...,a_n$, составляется новый полином P_4 и узлы аппроксимации смещаются в точки экстремумов ошибки аппроксимации и т.д. Итерационный процесс заканчивается в случае, если максимальные ошибки аппроксимации не превышают по абсолютной величине среднего арифметического ошибок на 1 – 5%. На рис. 2.7 пунктирной линией приведен график погрешности полинома наилучшего приближения.

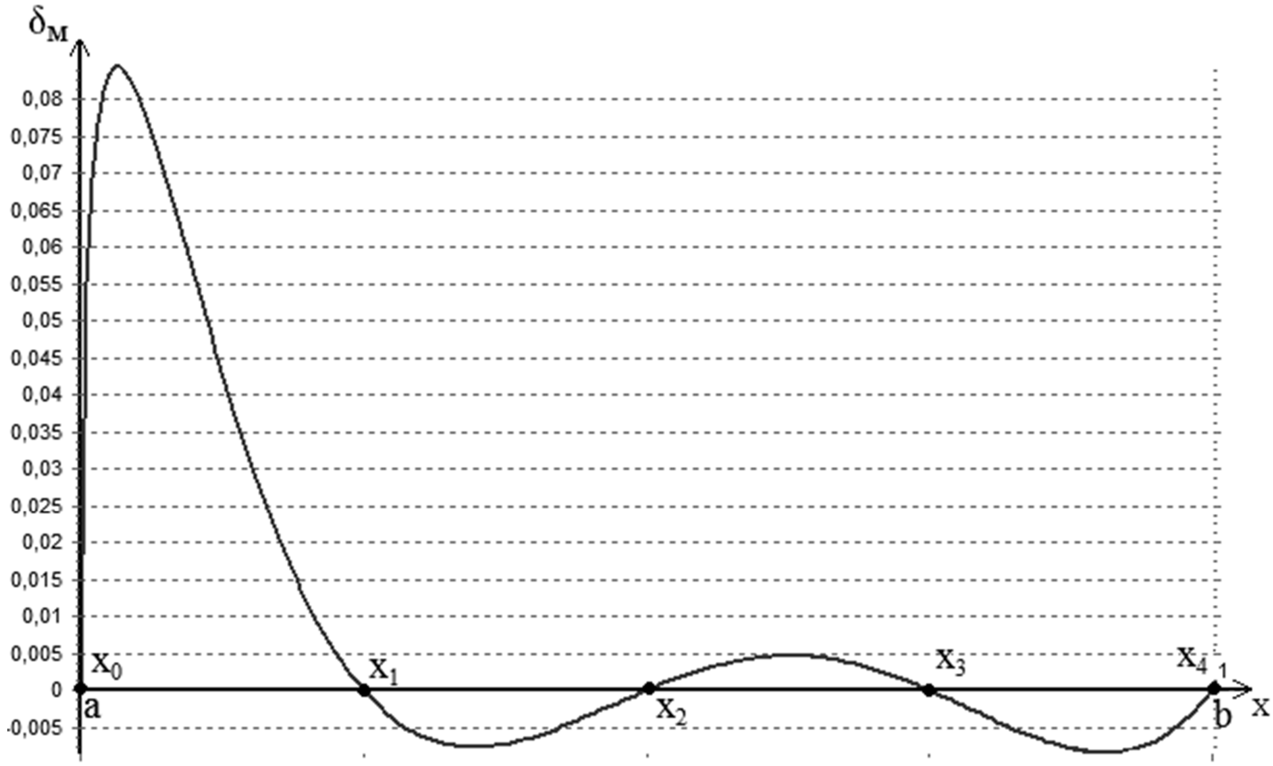


Рисунок 2.6 График погрешности исходного полинома (2.11)

Таким образом, исходными данными для расчета коэффициентов полинома наилучшего приближения являются:

1. Вид аппроксимируемой функции.
2. Порядок полинома n.
3. Нижняя и верхняя границы интервала аппроксимации $x \in [a, b]$.
4. Начальное положение точек аппроксимации.

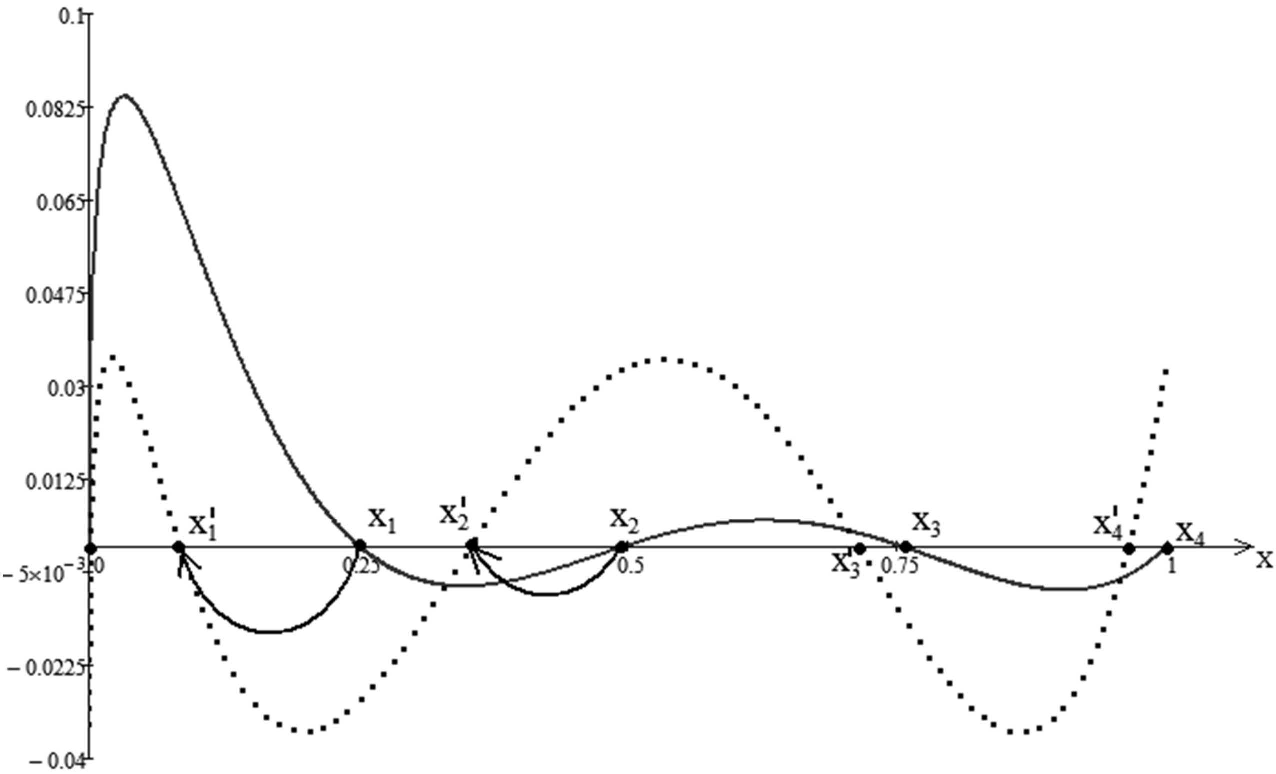


Рисунок 2.7 Графики погрешности исходного и оптимизированного полинома

2.3 Алгоритм взаимной компенсации составляющих погрешности метода и погрешности задания констант приближающего полинома

Другой подход для повышения точности аппроксимации основан на компенсации погрешностей аппроксимации и усечения количества разрядов коэффициентов полинома. При вычислении полинома в реальных условиях возникает дополнительная погрешность $\delta_к$ воспроизведения функции, обусловленная квантованием констант полинома с усечением разрядных сеток операндов вида

$$\delta_к = \Delta a_{0p} + \Delta a_{1p}x + \Delta a_{2p}x^2 + \ldots + \Delta a_{np}x^n, \tag{2.9}$$

где $\Delta a_{0p}, \ldots, \Delta a_{np}$ – погрешности усечения констант полинома в реальных условиях $\Delta a_{0p} = a_0 - a_{0p}$, $\Delta a_{1p} = a_1 - a_{1p}$.

По сравнению с (1.1) уменьшение числа членов, констант полинома требует меньшего числа операций, узлов аппроксимации и физических эталонов, как при вычислении функций, так и при калибровке измерительного канала системы. При компьютерном моделировании обеспечивается возможность получить полиномы наилучшего приближения, например, с 15-ю десятичными цифрами результата. В то же время уже с тремя десятичными цифрами фактически можно получить десятиразрядное представление результата в двоичной системе счисления. Такое представление обеспечивает возможность получить приведенную относительную погрешность результата порядка 0,1%. В измерительном канале с аналоговым датчиком, работающим на 8-ми разрядном АЦП принципиально возможно уменьшить приведенную относительную погрешность системы до значений $2^{-9}\cdot 100\,\%...2^{-8}\cdot 100\,\%$. АЦП, работающие с датчиками и системами имеют примерно равную 2-м дискретам приращения разрядности. Специализированные вычислители (СВ), реализуемые на ПЛИС также можно проектировать с переменной разрядностью. При меньшем числе разрядов существенно снижаются программно-аппаратурные затраты на реализацию ПЛИС и повышается их быстродействие. Поэтому актуально уменьшение разрядных сеток операндов при сохранении заданных точностных характеристик систем. В связи с этим на втором этапе производится такое сокращение разрядных сеток СВ, чтобы погрешность результата не увеличивалась бы более чем на 5-10% от погрешности δ_{MM}.

Обеспечим уменьшение погрешности δ_p на выходе системы, определяемой как сумма погрешностей метода $\delta_м$, квантованного представления δ_{a_i} набора констант a_i, промежуточных преобразований $\delta_{пр}$ с ограниченным числом разрядов в форматах данных СВ. При возрастании значений аргумента график погрешности $\delta_м$ полинома наилучшего приближения в (1.1) представляет знакочередующуюся функцию (рис.2.8). Его формой можно управлять, меняя узлы аппроксимации или коэффициенты в полиноме. Значения погрешностей отрезания констант a_i определяют степень наклона линейной функции $a_1 \cdot x$, растяжения парабол $a_i \cdot x^i$ и монотонно возрастают по абсолютной величине с ростом аргумента x. В то же время, в совокупности графики комбинаций кривых разного знака, например, $y = \Delta a_3 x^3 - \Delta a_1 x^1$, функции $\delta_k = \Delta a_{0p} + \Delta a_{1p} \cdot x + \Delta a_{2p} \cdot x^2 + ... + \Delta a_{np} \cdot x^n$ при определенных соотношениях квантованного представления δ_{ai} набора констант a_i могут напоминать графики погрешностей аппроксимации полиномом наилучшего приближения с противоположными знаками. Из анализа графиков совместного влияния составляющих погрешностей на погрешность δ_p и разрабатываются алгоритмы взаимной компенсации. Данные вопросы были частично рассмотрены в [85,89,91], однако, не был разработан алгоритм взаимной компенсации, не рассмотрена реализация данного алгоритма.

Компенсацию можно реализовать и методом перебора всех взаимных комбинаций отклонений констант в пределах определенного числа последних отбрасываемых цифр с фиксацией наименьшего значения δ_p. Число комбинаций $j = s^m$, где $m \langle n -$ количество коэффициентов полинома, а s – число перебираемых вариантов для одной константы a_i (рис.2.9).

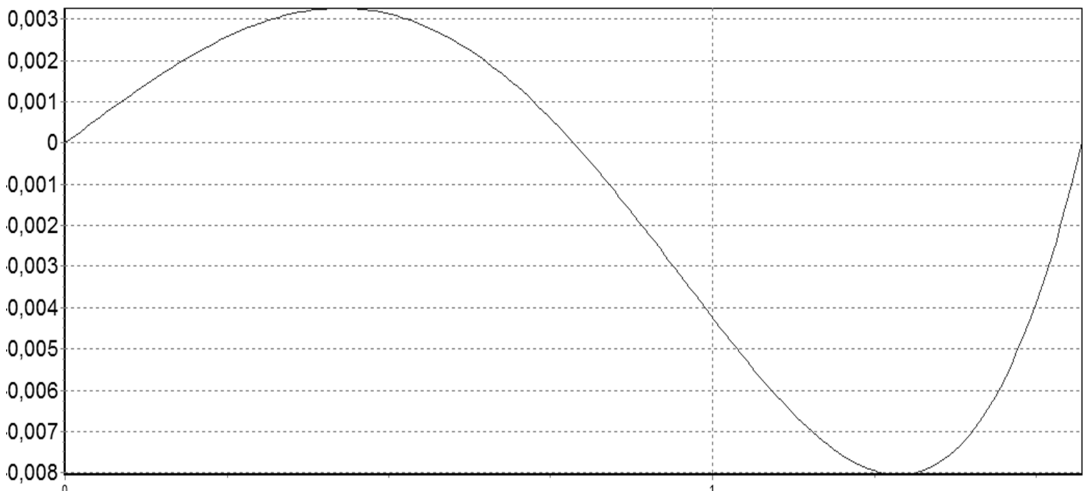


Рисунок 2.8 График погрешности полинома наилучшего приближения 3-й степени

Рисунок 2.9 Алгоритм минимизации числа разрядов для представления коэффициентов полинома

В соответствии с вышеизложенным, количество разрядов представления констант в десятичной системе счисления можно задать равным 15-ти или более. Такое количество разрядов для ряда применений является избыточным. Поэтому представим алгоритм, который позволял бы уменьшить количество разрядов после запятой при допустимом увеличении погрешности результата.

В качестве входных данных для взаимной компенсации составляющих погрешностей используем полином наилучшего приближения. После определения входных данных и последовательного урезания числа цифр коэффициентов с проверкой относительного приращения погрешности, осуществляется урезание критической разрядной цифры и определение числа итераций подбора коэффициентов. Впоследствии, запускается главный цикл перебора массива пробных коэффициентов. По выходу из цикла имеем набор компенсированных коэффициентов. Выполняется этот цикл методом прямого перебора по всем возможным комбинациям значений коэффициентов. Затем определяется максимальная погрешность для очередного набора коэффициентов и сравнивается с предыдущим значением. По окончанию происходит вывод параметров полинома и графика погрешности.

Для функции sin(x) документированы результаты моделирования в виде полинома: $P(x)$=0,99999662955348 x – 0,166648339090245 x^3 + +0,00830637908917905 x^5 – 0,000183651094361543 x^7 и график погрешности $\delta_м$ на интервале $x \in [0; \pi/2]$ со значением $\delta_m = 5{,}9 \cdot 10^{-7}$

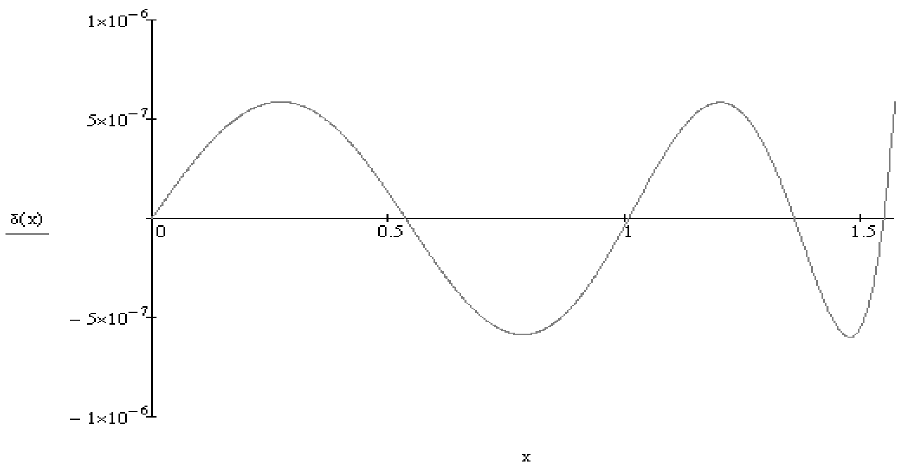


Рисунок 2.10 График погрешности приближения sin(x) полиномом
$P(x)$= (0,999996629 + (-0,166648338 + (0,008306378 – 0,000183650$\cdot x^2$)$\cdot x^2$)$\cdot x^2$)$\cdot x$

Так при приближении функции sin(x), на интервале $x \in [0;\pi/2]$ полиномом $P(x)$= (0,999996629 + (-0,166648338 + (0,008306378 – 0,000183650$\cdot x^2$)$\cdot x^2$)$\cdot x^2$)$\cdot x$ имеем график погрешности представленный на рис. 2.10. Погрешность метода при данном формате констант составляет 5,9$\cdot 10^{-7}$.

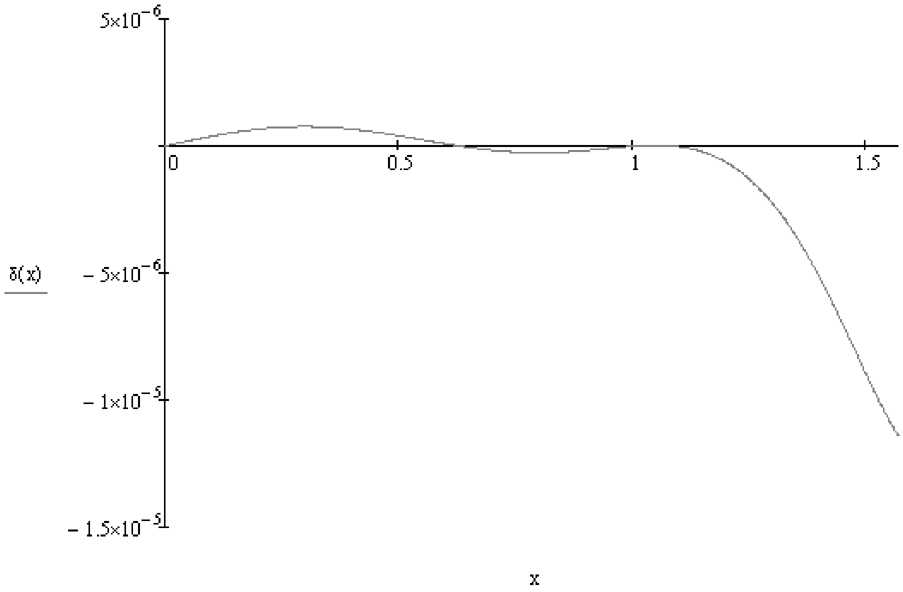


Рисунок 2.11 График погрешности приближения sin(x) полиномом

$P(x)$= (0.999996 + (-0,166648 + (0,008306 – 0,000183$\cdot x^2$)$\cdot x^2$)$\cdot x^2$)$\cdot x$

При урезании формата представления констант до 6 знаков после запятой то погрешность задания констант начинает резко влиять на погрешность результата. Так погрешность результата становится равна $1{,}14\cdot 10^{-5}$ (рис.2.11). Однако можно скомпенсировать взаимную погрешность и подстроить коэффициенты таким образом, как это представлено на рис.2.12. В этом случае погрешность результата составляет $9{,}13\cdot 10^{-7}$.

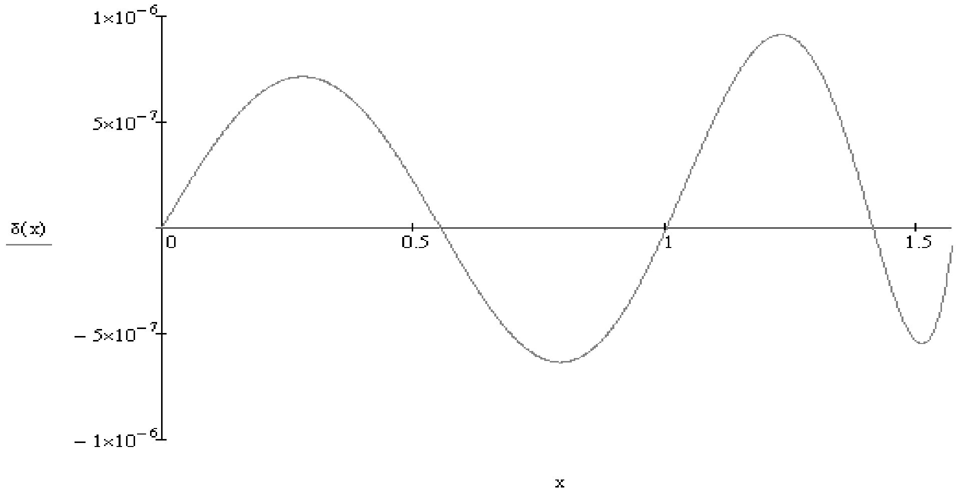


Рисунок 2.12 График погрешности приближения sin(x) полиномом

$P(x)$= (0,999996 + (-0,166646 + (0,008304 – 0,000183$\cdot x^2$)$\cdot x^2$)$\cdot x^2$)$\cdot x$

В соответствии с алгоритмом (рис.2.1) проведен поиск полиномов наилучшего приближения для часто используемой в измерительных системах функциональной зависимости sin(x) (таблица 2.1).

Таблица 2.1 Полиномы наилучшего приближения для функции sin(x)

Схема аппр-и	Интервал	Формула полинома	Максимальная погрешность	Количество	
				операций	констант
$P10$	[0; 90°]	$x(0,999999976705128+(-0,166666477121386+(0,00833290116196682+$ $+(-0,000198009808158351+2,59065692308463\cdot 10^{-6}x^2)x^2)x^2)x^2)$	$3,4\cdot 10^{-9}$	15	5
$P11$	[0; 90°]	$x(0,99999997+(-0,16666647+(0,083329+(-0,000198+2,59\cdot 10^{-6}x^2)x^2)x^2)x^2$	$2,1\cdot 10^{-7}$	15	5
$P12$	[0; 90°]	$x(0,99999998+(-0,16666649+(0,00833291+(-0,00019801+2,59\cdot 10^{-6}x^2)x^2)x^2)x^2$	$5\cdot 10^{-9}$	15	5

Для полинома наилучшего приближения функции sin(x) схемы аппроксимации Р10 максимальное значение погрешности $\delta_{мм}=3,4\cdot 10^{-9}$, а погрешности усечения констант фактически исключены, поскольку операнды представлены 17-ю разрядными десятичными цифрами. Простое усечение разрядных цифр до 8 в схеме увеличивает значение погрешности результата с $\delta_{мм}=3,4\cdot 10^{-9}$ до $\delta_{мм}=2,1\cdot 10^{-7}$, т. е. погрешность увеличивается в $2,1\cdot 10^{-7}/3,4\cdot 10^{-9}=62$ раза. Проведение взаимной компенсации погрешностей с 8-ю десятичными цифрами после запятой обеспечивает уменьшение погрешности в $2,1\cdot 10^{-7}/5\cdot 10^{-9}=42$ раза. Таким образом, прикладная значимость проведенной взаимной компенсации погрешностей в данном случае состоит в возможности сокращения разрядных сеток операндов специализированного вычислительного комплекса примерно на 3–5 двоичных разрядов без снижения класса точности измерительного прибора [16].

Для сокращения вычислительных операций, констант предлагаются и другие методы группирования членов полинома. Например, для вычисления функции cos x на интервале от 0^0 до 90^0 можно воспользоваться формулой аппроксимации полиномом 8-й степени с числом вычислительных операций равном 13-ти и максимальной погрешности метода $\delta_{мм} = 1{,}2 \cdot 10^{-4}$ (рис. 2.13)

$$f_{аппрq}(x) = 1 - \frac{1}{2} \cdot x^2 + \left(\frac{835}{2^{12}} \cdot x^2 - \frac{53}{2^{14}} \cdot x^4 \right)^2 = 1 + \left[-\frac{1}{2} + x^2 \cdot \left(\frac{835}{2^{12}} - \frac{53}{2^{14}} \cdot x^2 \right)^2 \right] \cdot x^2. \quad (2.10)$$

Алгоритм имеет более рациональную группировку членов полинома, чем в обычной схеме Горнера. Использование свернутой в (2.10) формулы $(c_1x^2+c_2x^4)^2$ при уже вычисленном значении x^2 обеспечивает с учетом извлечения из памяти констант уменьшение числа операций только при вычислении этого выражения с 7 при раскрытой формуле $(c_1x^2+c_2x^4)^2$ на 4. В свою очередь последующий вынос за общие скобки множителя x^2 позволяет уменьшить общее число операций в (2.13) до 12-ти. Но предлагаемый алгоритм в представленном виде уступает «обычным» оптимизированным полиномам Чебышева вычисления функции cos (x) и тем более sin (x), полученным в соответствии с программами поиска полиномов наилучшего приближения и полиному наилучшего приближения для функции косинуса (2.11) по точностным характеристикам

$$P(x)=0{,}9999933 + (-0{,}4999127 + (0{,}0414881 - 0{,}0012713 \cdot x^2)\, x^2)\, x^2) \quad (2.11)$$

Так для (2.11) при меньшем числе операций равном 11 значение погрешности меньше в $1{,}2 \cdot 10^{-4}/6{,}35 \cdot 10^{-6}=18{,}9$ раз. В тоже время для выражения (2.11) значение погрешности метода можно уменьшить в несколько раз путем частичного превращения его в полином наилучшего приближения. Отметим, что для полиномов (3-9)-й степеней при a_1=1 уменьшается число узлов интерполяции, констант полинома и (или) физических эталонов для калибровки ИИС.

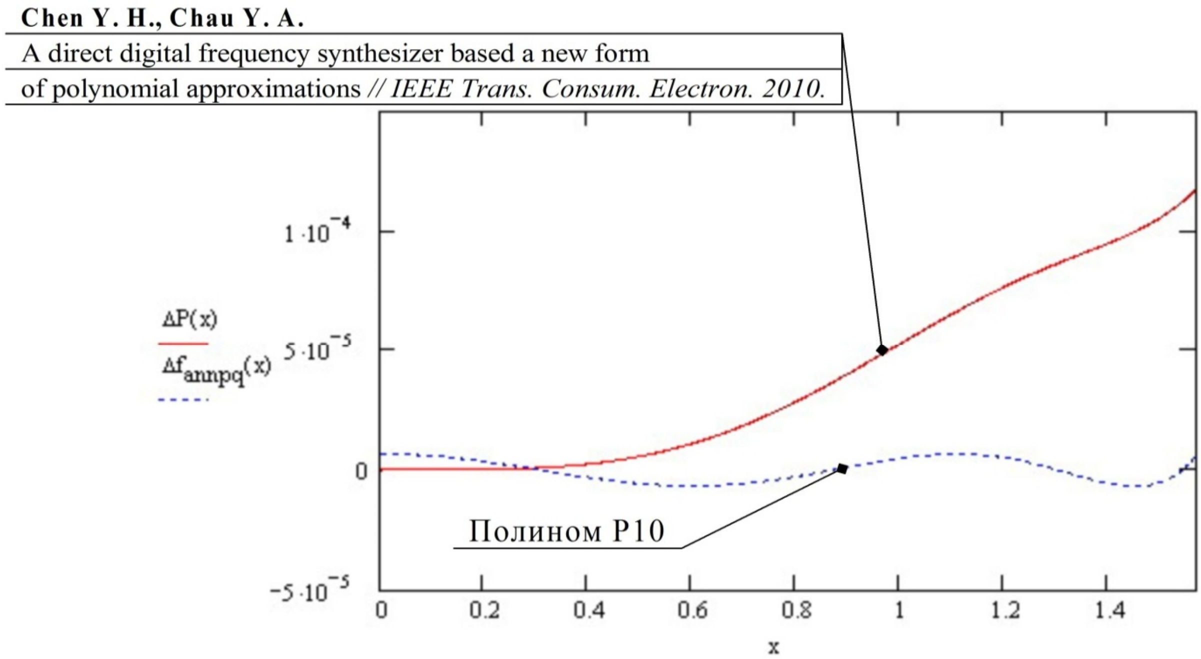


Рисунок 2.13 Графики погрешностей полиномиальных приближений (2.10) и (2.11)

2.4 Разработка программ нахождения полинома наилучшего приближения и компенсации ошибок аппроксимации

Методы поиска полиномов наилучшего приближения реализованы в средах программирования Builder C++. Исходными входными данными для поиска полинома наилучшего приближения являются:

- вид аппроксимируемой функции;
- порядок полинома – n;
- нижняя и верхняя границы интервала аппроксимации - $x \in [a, b]$;
- используемые члены – четные, нечетные, все;
- наличие экстраполяции - есть или нет.

Выходными данными являются коэффициенты полинома, значение максимальной погрешность, а также график изменения погрешности аппроксимации в интервале $[a, b]$.

На рис. 2.14 показана декомпозиция функциональной диаграммы разрабатываемой программы в среде Delphi. Декомпозиция функциональной диаграммы показывает, что функционирование программы состоит из двух основных блоков, это расчет полинома Ньютона и по данным расчетов полинома Ньютона, нахождение полинома наилучшего приближения. В соответствии с этой схемой на выходе получается полином наилучшего приближения.

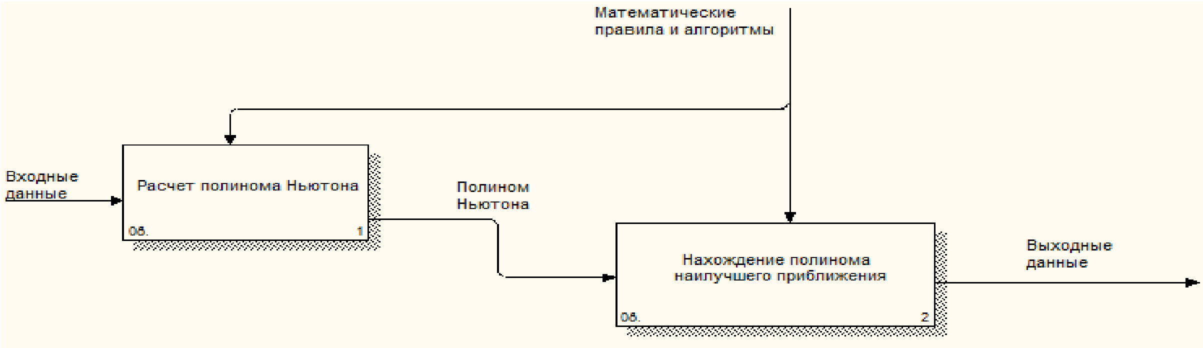


Рисунок 2.14 Декомпозиция функциональной диаграммы разрабатываемой программы

На рис. 2.15 показана общая блок-схема алгоритма для нахождения полинома наилучшего приближения. В качестве входных данных используются: вид аппроксимируемой функции, порядок полинома, интервал аппроксимации [30].

В качестве начального приближения используется полином Ньютона, так как все узловые точки равноудалены друг от друга. После получения значений узловых точек, составляется система линейных алгебраических уравнений и находится ее решение методом Гаусса [31]. После нахождения полинома начального приближения программа переходит к процессу поиска полинома наилучшего приближения, расширенная блок-схема подпрограммы приведена на рис. 2.16.

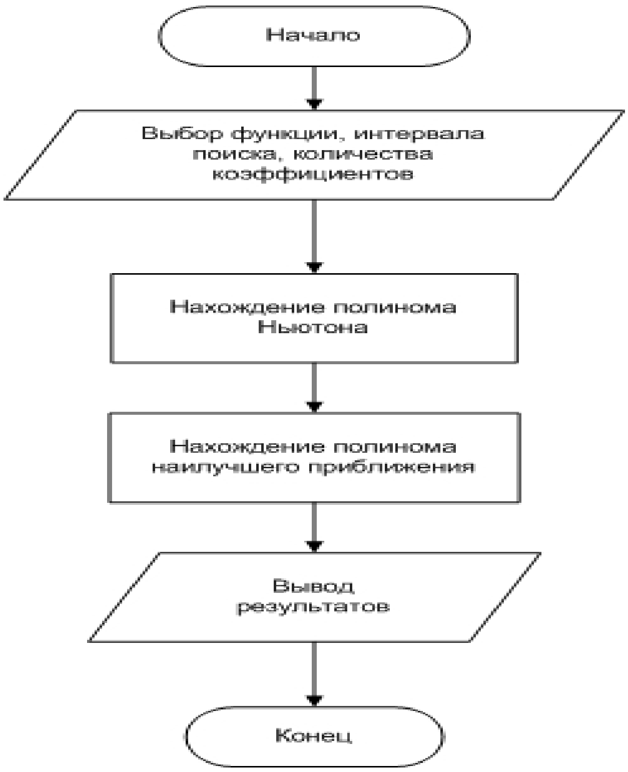


Рисунок 2.15 Общая блок схема алгоритма для нахождения полинома наилучшего приближения

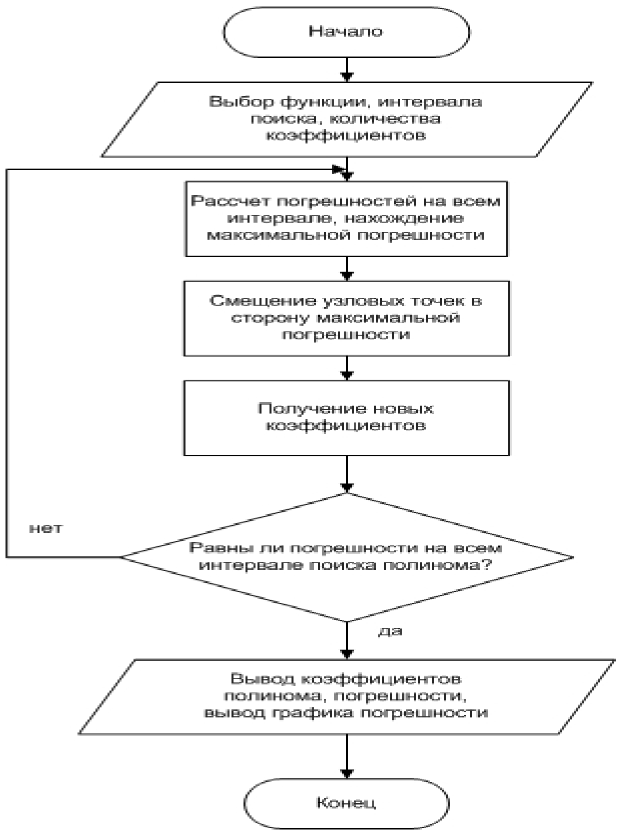


Рисунок 2.16 Процесс поиска полинома наилучшего приближения

Определение оптимальных точек производится итерационно за счет смещения узлов аппроксимации на каждой итерации в точки экстремумов ошибки аппроксимации. На время реализации алгоритма влияет: количество точек, на которое будет разбит интервал; точность вычисления, характеристики информационно-вычислительной системы; степень полинома.

Выходными данными работы программы являются [32]:

- коэффициенты полинома наилучшего приближения;

- максимальная погрешность результата;

- график погрешности полинома наилучшего приближения на всем интервале аппроксимации;

- сформированное из полученных коэффициентов выражение полинома.

Вторая программа разработана в средах программирования MathCAD, Delphi, Builder C++ и служит для поиска полиномов наилучшего приближения с учетом компенсации погрешностей. Блок-схема программы приведена на рис.2.9.

Перед проведением эксперимента вводится количество членов полинома, задается аппроксимируемая функция, начало и конец интервала, определяется наличие, например, только членов полинома с нечетной степенью. Под надписью “Экстраполяция” указателю “Есть” соответствует присутствие члена полинома a_0. В следующем за этой надписью окне задается число цифр после запятой. Командой “Пуск” задается полином Ньютона, а командой “Улучш.” этот полином превращается в полином наилучшего приближения. Предусмотрена последующая компенсация составляющих погрешностей результата при урезании разрядных сеток.

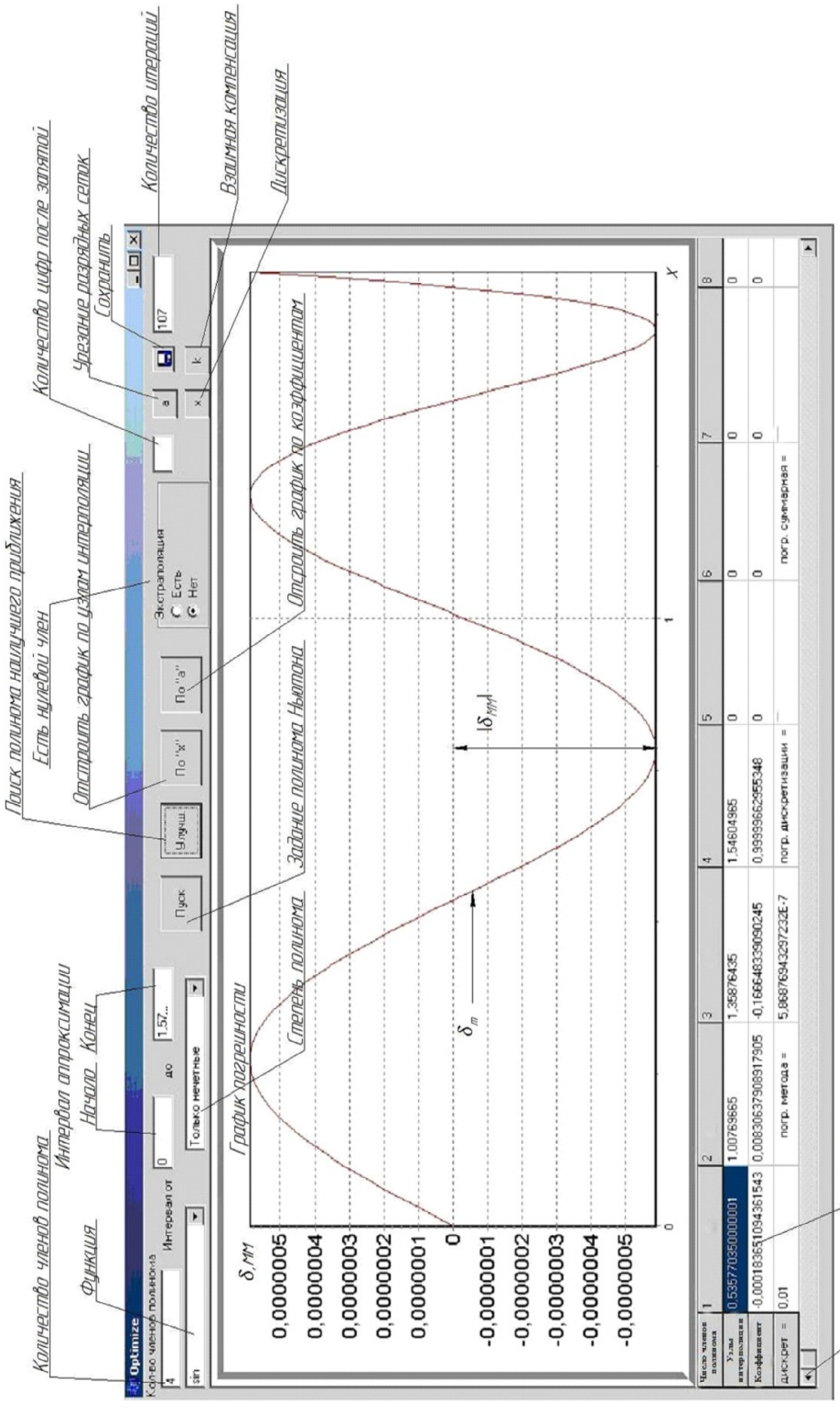


Рисунок 2.17 Окно программы поиска полинома наилучшего приближения для воспроизведения функциональных зависимостей с взаимной компенсацией составляющих погрешностей результата

Выводы к главе 2

1. Разработаны алгоритмы получения полиномов наилучшего приближения различных степеней для аппроксимации различных функций и градуировочных характеристик. Использование этих методов позволяет уменьшить погрешность формирования функции sin(x) на интервале $x \in [0^0;90^0]$, с точностью δ=0,105, что в 2 раза выше, чем у полинома Ньютона. Для полиномов второй степени функции квадратного корня поиск полинома наилучшего приближения осуществляется за 10-12 итераций, что в 3 раза лучше по сравнению с ранее известными методами.

2. Разработаны алгоритмы нахождения полиномов аппроксимации, позволяющие уменьшить погрешности результата путем взаимной компенсации погрешностей численного метода решения задачи, квантования констант аппроксимирующего полинома в цифровом вычислительном устройстве радиотехнической системы, а также симметрирования составляющих погрешностей дискретизации и округления. Как показали результаты моделирования, использование этих алгоритмов позволяет сократить разрядные сетки операндов специализированного вычислительного комплекса примерно на 3–5 двоичных разрядов без снижения класса точности измерительного прибора, при этом погрешность уменьшается в 2-30 раз.

3. В средах программирования Builder, C++, MathCAD разработано прикладное программное обеспечение, позволяющее находить полиномы наилучшего приближения и полиномы с учетом взаимной компенсации погрешностей. Программное обеспечение позволило найти коэффициенты полиномов наилучшего приближения разной степени, а так же подробно проанализировать точность полученных полиномом при различных условиях для разных типов функций.

Глава 3 Оптимизация методов вычисления стандартных функций и векторных операций, применяемых в РТС

На практике при реализации алгоритмов аппроксимации элементарных функций наиболее важным является уменьшение вычислительных затрат при различном количестве разрядов представления результатов вычислений с сохранением достаточной точности аппроксимации. Для реализации этого подхода при аппроксимации функций можно использовать полиномы высоких степеней, отбрасывать слагаемые, слабо влияющие на точность аппроксимации и в результате получить более быстрые и точные алгоритмы аппроксимации. В этой главе рассматриваются методы поиска полиномов наилучшего приближения при аппроксимации функций синус, тангенс, арктангенс и арксинус, исходя из предельных соотношений по точностным характеристикам, времени вычислений и программно-аппаратным затратам.

Целевая функция оптимизации специализированных алгоритмов воспроизведения функциональных зависимостей с устранением излишней точности E соответствует отношению последовательного дискретного приращения максимального числа значащих двоичных разрядов операндов выходных данных G при минимальном возрастании сложности вычислительных алгоритмов и соответственно времени их реализации C [33]. Оценка эффективности алгоритма наглядно иллюстрируется максимизируемым выигрышем G при ограниченных затратах C, не превосходящих некоторой величины C* или минимизируемыми затратами, при условии, что выигрыш от применения алгоритма не менее заданного G*:

$$E = G \to \max \left| C \le C^* \right., \; E = G \to \min \left| C \ge C^* \right., \; E = G/(A+m), \tag{3.1}$$

где G – число значащих двоичных цифр результата или их приращения ΔG от некоторых начальных условий, $C=A+m$ – число выполненных операций или их приращение $\Delta(A+m)$, A обозначает количество математических операций, m – число извлекаемых из памяти констант. Исходя из вышесказанного, создадим оптимизированные алгоритмы воспроизведения стандартных функций в диапазоне от 3…64 значащих цифр результата.

3.1 Улучшение методов аппроксимации элементарных функций полиномиальными методами

3.1.1 Аппроксимация синусоидальной функции. Полином наилучшего приближения функции sin (x) первой степени $y_{ч1} = a_0 + a_1$ имеет вид

$$y_{ч1} = 0{,}105 + 0{,}636 \cdot x \qquad (3.2)$$

Как видно из рис. 3.1 а) на интервале $x \in [0; \pi / 2]$ максимальное значение погрешности при использовании полинома (3.2) составляет $\delta = 0{,}105 = 2^{-3{,}2}$. Этот полином получен путем решения задачи вариационного исчисления, когда при поиске минимального значения погрешности не закреплены левое (x_1) и правое (x_2) положения узлов интерполяции. В общем случае для реализации алгоритма вычисления полинома из ПЗУ необходимо выбрать две константы a_0, a_1 и выполнить операцию алгебраического умножения и сложения, то есть всего осуществить 4 операции.

При разложении функции sin(x) в ряд Тейлора и сохранению линейного члена ряда y=x, максимальное значение погрешности будет при $x = \pi / 2$ и составит $\delta_{1T} = |1 - 1{,}57| = 0{,}57$. Интересно отметить, что меньшее значение погрешности можно получить, если ограничиться на интервале $x \in [0; \pi / 2]$ только значением $y_0 = a_0 = 0{,}5$ с погрешностью $\delta = |0{,}5|$ или $y_{T1C} = -0{,}285 + x$ с погрешностью 0,285 и $a_1 = 1$, или использовать полином $y = a_1 \cdot x$ с $a_0 = 0$, с обеспечением поиска оптимального коэффициента $a_i = 0{,}724$ на правом подвижном конце (рис. 3.1 б).

Таким образом, при значениях $\delta \in [0{,}5; 0{,}15]$ применение полинома (3.2) неэффективно, а наиболее целесообразно использовать полином $y = 0{,}7246 \cdot x$ со значением погрешности $\delta_1 = 0{,}137 \cong 2^{-3}$ (рис. 3.1 б), когда только при двух (а фактически одной) вычислительных операциях получается около 3-х значащих двоичных цифр результата.

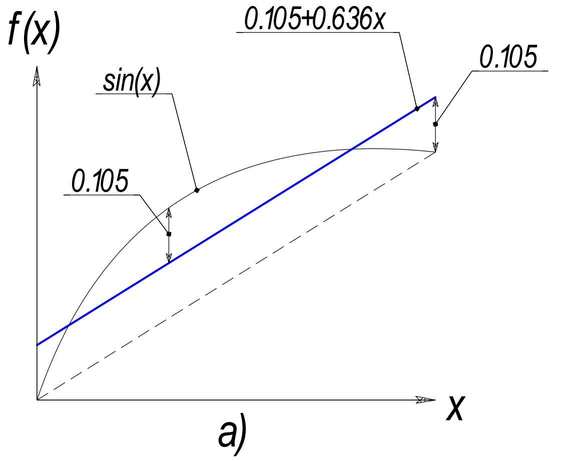


Рисунок 3.1 Полиномы первой степени наилучшего приближения функции sin(x) с нулевым членом (а) и без нулевого члена (б)

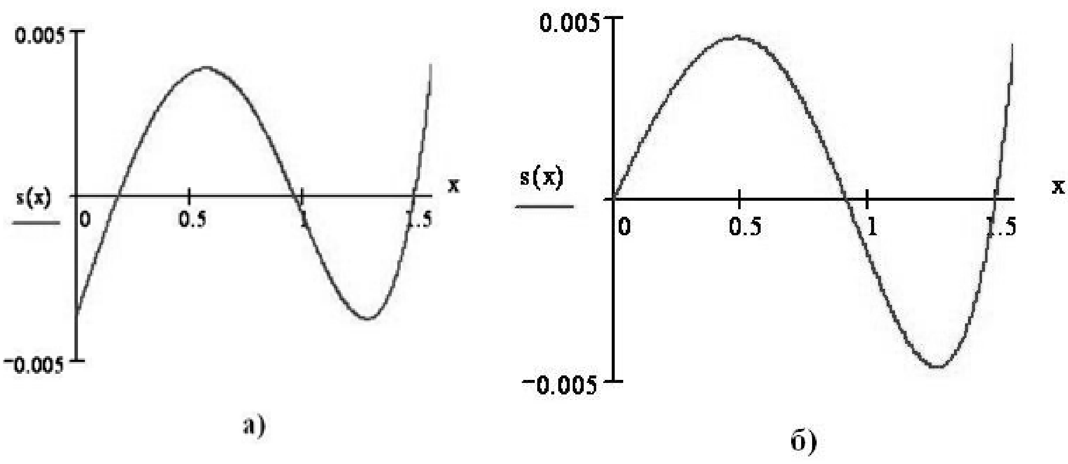
Для реализации оптимального алгоритма вычисления полинома было использовано компьютерное моделирование, которое позволяет устранить избыточную точность, обеспечить последовательное дискретное приращение точности не менее 1..3 значащих двоичных цифр при последовательном возрастании сложности вычислительного алгоритма не более чем на 1..2 вычислительных операции. Компьютерное моделирование с целью поиска полиномов с наиболее низким произведением $\delta_M(A+m)$ подтвердило, что разложение в ряд при использовании для нечетной функции sin(x) комбинаций членов с четными степенями неэффективно [26]. Установлено, что для интервала значений погрешностей $\delta \in [0{,}15; 0{,}01]$ наиболее эффективно применение полинома

$$y = x - 0{,}14966 \cdot x^3, \qquad (3.3)$$

поскольку исключаются операции выборки из ПЗУ коэффициентов $a_0 = 0$, $a_1 = 1$. Разработанная компьютерная программа моделирования полиномов (3.2) и (3.3) показала, что наибольшее значения погрешности для полинома (3.3) составило $\delta = 0{,}01 \cong 2^{-6}$ при пяти вычислительных операциях по сравнению с применением полинома (3.2). Уменьшение значения погрешности составило $0{,}105 / 0{,}01 = 10{,}5$ раз. На одну вычислительную операцию получено эффективное приращение более трех значащих двоичных цифр результата.

Использование в (3.3) схемы Горнера предполагает дополнительную выборку константы 1, и поэтому общее сокращение числа операций не обеспечивает. В то же время применение полинома $y = (0{,}9855 - 0{,}1426 \cdot x^2) \cdot x$ со схемой Горнера, с возможностью варьирования и оптимизации константы $a_1 \neq 1$ при увеличении числа операций только на одну по сравнению с (3.3) позволяет уменьшить значение погрешностей еще в $0{,}01 / 0{,}0046 = 2{,}17$ раза. В связи с этим для интервала $\delta \in [0{,}01; 0{,}05]$ наиболее эффективно применение полинома $y = (0{,}9855 - 0{,}1426 \cdot x^2) \cdot x$.

Эффективное сокращение числа операций при незначительном увеличении значений погрешности для трех полиномов третьей степени: $y = a_0 + a_1 \cdot x + a_3 \cdot x^3$ при $a_0 \neq 0$, $a_0 = 0$ и $a_1 = 1$ проиллюстрировано на рис. 3.2, где для всех полиномов получены единственно возможные наименьшие отклоняющиеся от нуля симметричные значения погрешностей.



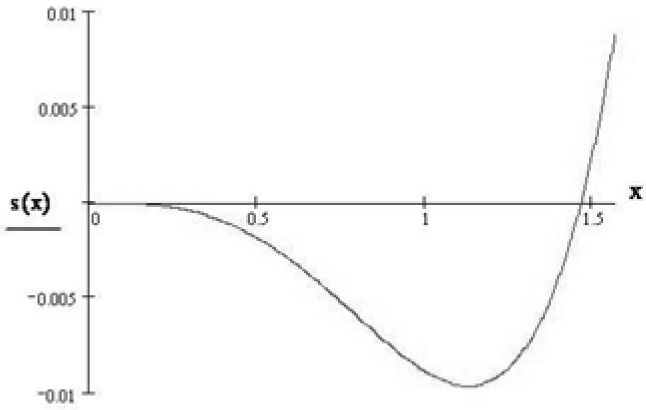


Рисунок 3.2 Графики погрешности по абсолютной величине полиномов 3-й степени а) при $a_0 \neq 0$, б) при $a_0 = 0$, в) при $a_1 = 1$.

В соответствии с рассмотренными принципами, примерами для обеспечения оптимальных соотношений по точностным характеристикам для диапазона $\delta_M \in [50\%...10^{-6}\%]$, числу вычислительных операций, обращений к ПЗУ, программно-аппаратным затратам путем компьютерного моделирования получен набор полиномов для приближения функции $\sin(x)$ при $x \in [0; \pi/2]$, который приведен в таблице 3.1. В таблице 3.1 № обозначает значение степени полинома. Для полиномов 1-й и 3-й степени вычисления функции $\sin(x)$ по схемам аппроксимации с $a_0 = 0$, $a_1 \neq 1$ при погрешностях $\delta_{MM} = 0{,}137$ и $\delta_{MM} = 0{,}006$ необходимо реализовать соответственно 2 операции, хранение в памяти 1 константы и 6 операций с хранением в памяти 2-х констант. Уменьшению погрешности по сравнению с полиномом 1-й степени в $0{,}137/0{,}006 = 22{,}8$ раз при увеличении числа операций на 4 и констант в памяти на одну примерно соответствует приращение $4{,}25/4 = 1{,}06$двоичных разрядных цифр на 1 операцию и т.д. Для полинома 9-й степени, $a_0 = 0$ $a_1 = 1$ уменьшение погрешности по сравнению с полиномом 7-й степени равно $7 \cdot 10^{-7} / 4 \cdot 10^{-9} = 175$. После стартового приближения функции для полинома 1-й степени при $a_0 = 0$ $a_1 \neq 1$ примерно с тремя двоичными цифрами результата на две двоичные операции в дальнейшем при увеличении степени полинома на 2 получаем хорошие нарастающие приращения числа разрядных цифр на одну операцию: 1,06 разрядных цифр (с полинома 1-й степени на полином 3-й степени), 2 с полинома 3-й степени на полином 5-й степени, 2,3 с полинома 5-й степени на полином 7-й степени и 2,4 двоичных цифры для полинома 9-й степени на одну операцию [34]. Для полинома 1-й степени число операций равно 2, для полинома 3-й степени 6 и в последующем число операций увеличивается на 3, и каждый раз число констант в памяти надо увеличивать на 1.

Таблица 3.1 Полиномы вычисления sin (x) на интервале $x \in [0; \pi / 2]$

№	Формулы полиномов		δ_{MM}	$A+m$
0	$a_0 \neq 0$	$P(x) = 0,5$	0,5	1
1	$a_0 \neq 0$ $a_1 = 1$	$P(x) = -0,285 + x$	0,285	2
	$a_0 = 0$ $a_1 \neq 1$	$P(x) = 0,7246 \cdot x$	0,137	2
	$a_0 \neq 0$ $a_1 = 1$	$P(x) = 0,105 + 0,636 \cdot x$	0,105	4
3	$a_0 = 0$ $a_1 = 1$	$P(x) = x - 0,14966 \cdot x^3$	0,01	5
	$a_0 = 0$ $a_1 \neq 1$	$P(x) = x \cdot (0,9857 - 0,1426 \cdot x^2)$	0,006	6
	$a_0 \neq 0$ $a_1 \neq 1$	$P(x) = 0,0035 + x \cdot (0,9794 - 0,1409 \cdot x^2)$	0,005	8
5	$a_0 = 0$ $a_1 = 1$	$P(x) = x + (-0,16607 + 0,00763 \cdot x^2) \cdot x^3$	$1,4 \cdot 10^{-4}$	8
	$a_0 = 0$ $a_1 \neq 1$	$P(x) = x \cdot (0,999659 + (-0,165626 + 0,0075 \cdot x^2) \cdot x^2)$	$8 \cdot 10^{-5}$	9
	$a_0 \neq 0$ $a_1 \neq 1$	$P(x) = 0,000056 + x \cdot (0,999559 + (-0,165581 +$ $+0,007494 \cdot x^2) \cdot x^2)$	$7 \cdot 10^{-5}$	11
7	$a_0 = 0$ $a_1 = 1$	$P(x) = x + (-0,16665438 + (0,0083089 -$ $-0,00018384 \cdot x^2) \cdot x^2) \cdot x^3$	$1,5 \cdot 10^{-6}$	11
	$a_0 = 0$ $a_1 \neq 1$	$P(x) = x \cdot (0,99999692 + (-0,16664889 +$ $+(0,00830678 - 0,00018374 \cdot x^2) \cdot x^2) \cdot x^2)$	$7 \cdot 10^{-7}$	12
	$a_0 \neq 0$ $a_1 \neq 1$	$P(x) = 0,00000054 + x \cdot (0,99999491 + (-0,16664577 +$ $+(0,0083048 - 0,00018334 \cdot x^2) \cdot x^2) \cdot x^2)$	$6 \cdot 10^{-7}$	14

Продолжение таблицы 3.1

№	Формулы полиномов		δ_{MM}	A+m
9	$a_0 = 0$ $a_1 = 1$	$P(x) = x + (\text{-}0{,}166666515 + (0{,}008332906 +$ $(\text{-}0{,}0001979999 + 0{,}00000258769 \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^3$	$1 \cdot 10^{-8}$	14
	$a_0 = 0$ $a_1 \neq 1$	$P(x) = x \cdot (0{,}999999974755 + (\text{-}0{,}166666469385 +$ $+(0{,}008332892062 + (\text{-}0{,}00019800584 +$ $+0{,}00002590091 \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2)$	$3{,}47 \cdot 10^{-9}$	15
	$a_0 \neq 0$ $a_1 \neq 1$	$P(x) = 0{,}00000000282 + x \cdot (0{,}99999996413 +$ $+(\text{-}0{,}16666644834 + (0{,}008332871968 +$ $+(\text{-}0{,}00019799744 + 0{,}00000258882 \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2)$	$3{,}5 \cdot 10^{-9}$	17
11	$a_0 = 0$ $a_1 = 1$	$P(x) = x + (\text{-}0{,}16666666561 + (0{,}008333329213 +$ $+(\text{-}0{,}000198406823 + (0{,}000002751799 \text{ -}$ $-0{,}0000000237861 \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^3$	$6 \cdot 10^{-11}$	17
	$a_0 = 0$ $a_1 \neq 1$	$P(x) = (0{,}999999999874 + (\text{-}0{,}166666665294 +$ $+(0{,}008333329026 + (\text{-}0{,}0001984068315 +$ $+(0{,}0000027518132 \text{ - } 0{,}000000023785 \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2) \cdot x$	$1{,}7 \cdot 10^{-11}$	18
	$a_0 \neq 0$ $a_1 \neq 1$	$P(x) = 0{,}000000000014 + (0{,}999999999822 +$ $+(\text{-}0{,}166666665182 + (0{,}0083333288855 +$ $(\text{-}0{,}0001984067409 + (0{,}0000027517847 \text{ -}$ $-0{,}0000000237815 \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2) \cdot x$	$1{,}45 \cdot 10^{-11}$	19

Таблицу 3.1 следует использовать при программно-аппаратной реализации функции $\sin(x)$ с заданной погрешностью в системах обработки информации с 8, 16, 32 разрядами представления формата данных с фиксированной точкой.

Для часто используемой в технических приложениях функции $\sin(x)$ с погрешностью порядка 0,1% на рис. 3.3 приведен алгоритм вычисления полинома $x + (\text{-}0{,}16607 + 0{,}00763 \cdot x^2) \cdot x^3$, где частично используется схема Горнера для сокращения числа операций умножения.

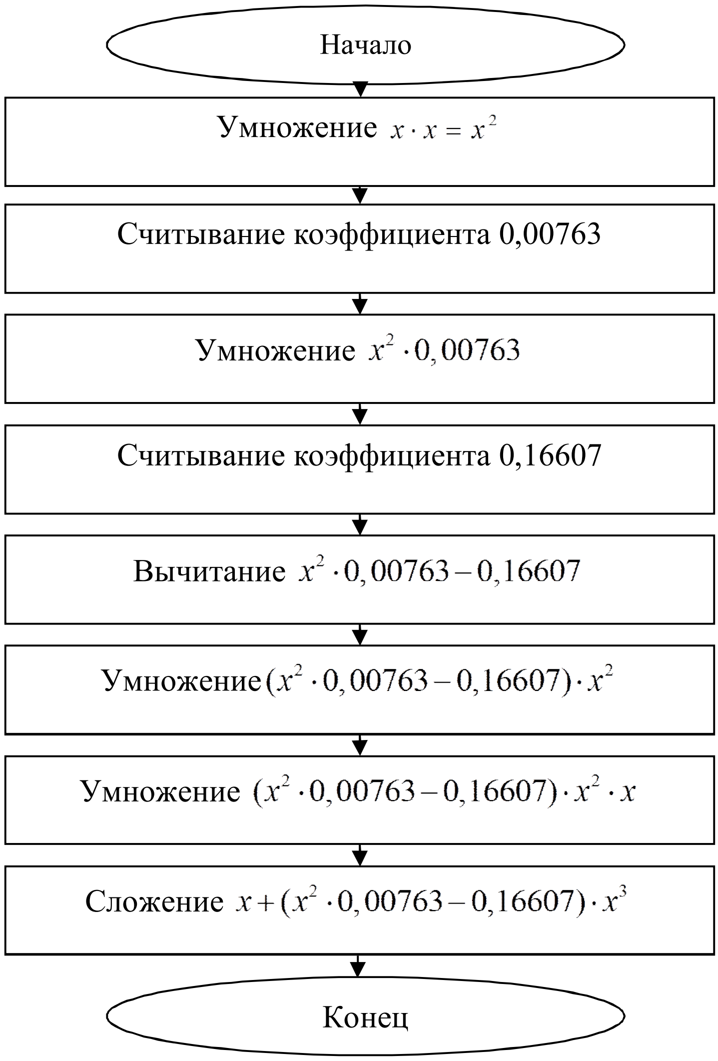


Рисунок 3.3 Алгоритм вычисления функции sin(x) для $x \in [0; \pi / 2]$

$$y_{45} = x + (-0,16607 + 0,00763 \cdot x^2) \cdot x^3$$

Для диапазона аргумента x от 0 до 2π значения функции sin(x) получают путем использования тригонометрических формул приведения к $x \in [0; \pi / 2]$.

Для всех указанных полиномов 3…11 степеней при увеличении числа операций на единицу по сравнению с полиномами, где $a_0 = 0$, $a_1 = 1$ погрешность уменьшилась примерно в два раза. Эффективность применения полиномиального метода вычисления функции sin(x) возрастает при увеличении степени полинома. Так, например отношение значений погрешностей $\delta_5 / \delta_7 = 1,4 \cdot 10^{-4} / 1,5 \cdot 10^{-6} = 93$. При увеличении числа операций с 8-ми до 11 обеспечивается приращение числа значащих двоичных цифр результата более чем на 6.

Для одного полинома 9-й степени (таблица 3.2) приближения sin(x) погрешность $\delta_{MM} = 3{,}47 \cdot 10^{-9}$, а погрешности урезания констант исключены, поскольку они представлены 10-11-ю разрядными десятичными цифрами. Простое урезание разрядных цифр до 8 увеличивает значение погрешности результата с $\delta_p = \delta_{мм} = 3{,}47 \cdot 10^{-9}$ до $\delta_p = 2 \cdot 10^{-7}$, то есть в 58 раз. При проведении компенсации погрешность уменьшается в $2 \cdot 10^{-7} / 4{,}86 \cdot 10^{-9} = 41$ раз с сокращением разрядных сеток операндов на 5,5 двоичных разрядов.

Таблица 3.2 Полиномы вычисления sin(x) с урезанием разрядных сеток

Число цифр	Формулы полиномов 9-й степени на интервале $x \in [0; \pi / 2]$	δ_p
Без компенсации погрешностей		
11	$P(x) = (0{,}99999997658 + (-0{,}16666647635 + (0{,}00833289986 +$ $(-0{,}00019800901 + 0{,}00000259049 \cdot x^2) \cdot x^2 \cdot x^2) \cdot x^2) \cdot x$	$3{,}47 \cdot 10^{-9}$
10	$P(x) = (0{,}9999999765 + (-0{,}1666664763 + (0{,}0083328998 +$ $+(-0{,}0001980090 + 0{,}0000025904 \cdot x^2) \cdot x^2 \cdot x^2) \cdot x^2) \cdot x$	$7{,}56 \cdot 10^{-9}$
9	$P(x) = (0{,}999999976 + (-0{,}166666476 + (0{,}008332899 +$ $+(-0{,}000198009 + 0{,}000002590 \cdot x^2) \cdot x^2 \cdot x^2) \cdot x^2) \cdot x$	$3{,}35 \cdot 10^{-8}$
8	$P(x) = (0{,}99999997 + (-0{,}16666647 + (0{,}00833289 +$ $+(-0{,}00019800 + 0{,}00000259 \cdot x^2) \cdot x^2 \cdot x^2) \cdot x^2) \cdot x$	$1{,}07 \cdot 10^{-7}$
7	$P(x) = (0{,}9999999 + (-0{,}1666664 + (0{,}0083328 +$ $+(-0{,}0001980 + 0{,}0000025 \cdot x^2) \cdot x^2 \cdot x^2) \cdot x^2) \cdot x$	$5{,}83 \cdot 10^{-6}$
С компенсацией погрешностей		
11	$P(x) = (0{,}99999997672 + (-0{,}16666647714 + (0{,}00833290114 +$ $(-0{,}00019800978 + 0{,}00000259065 \cdot x^2) \cdot x^2 \cdot x^2) \cdot x^2) \cdot x$	$3{,}32 \cdot 10^{-9}$
10	$P(x) = (0{,}9999999769 + (-0{,}1666664775 + (0{,}0083329015 +$ $(-0{,}0001980091 + 0{,}0000025907 \cdot x^2) \cdot x^2 \cdot x^2) \cdot x^2) \cdot x$	$3{,}37 \cdot 10^{-9}$
9	$P(x) = (0{,}999999975 + (-0{,}166666474 + (0{,}008332901 +$ $+(-0{,}000198011 + 0{,}000002591 \cdot x^2) \cdot x^2 \cdot x^2) \cdot x^2) \cdot x$	$3{,}75 \cdot 10^{-9}$
8	$P(x) = (0{,}99999998 + (-0{,}16666649 + (0{,}00833291 + (-0{,}00019801 +$ $+0{,}00000259 \cdot x^2) \cdot x^2 \cdot x^2) \cdot x^2) \cdot x$	$4{,}86 \cdot 10^{-9}$
7	$P(x) = (0{,}9999999 + (-0{,}1666661 + (0{,}0083327 + (-0{,}0001981 +$ $+0{,}0000026 \cdot x^2) \cdot x^2 \cdot x^2) \cdot x^2) \cdot x$	$4{,}96 \cdot 10^{-8}$

Введение на интервале $[0; \pi/2]$ двух подинтервалов с двумя полиномами 1-й степени и выше с одинаковыми значениями погрешности метода обеспечивает уменьшение погрешности по сравнению с полиномом $L = 0,724 \cdot x$ в 5,75 раза при увеличении сложности алгоритма на 4 операции, то есть не получаем приращения в одну разрядную цифру на операцию. Два полинома 3-й степени также менее эффективны, чем применение одного полином 5-й степени на всем интервале $[0;\pi/2]$. Таким образом, целесообразно для аппроксимации функции sin(x) в диапазоне погрешностей $10^{-1} - 10^{-11}$ и ниже использовать полиномиальную аппроксимацию на интервале $x \in [0;\pi/2]$. Не эффективно и введение константы a_0 в отличие от исключения константы a_1 $(a_1 = 1)$, например, в полиномах 3-й – 9-й степеней, когда, например, для полинома 3-й степени можно уменьшить дискретное приращение числа операций на одну при значении погрешности в интервале $10^{-3} < \delta_{MM} < 10^{-2}$ с 4-х до 3-х. Для полинома 9-й степени с $a_1 = 1$ по сравнению с полиномом 7-й степени при $a_1 \neq 1$ получаем повышение точности в $7 \cdot 10^{-7} / 1 \cdot 10^{-8} = 70$ раз или почти в 3 двоичные цифры результата на одну операцию. Для полиномов (3-9)-й степеней при $a_1 = 1$ уменьшается число узлов интерполяции, констант полинома и (или) физических эталонов для калибровки измерительной системы. Для уменьшения дискрета приращения числа операций целесообразно исключать константу a_1 (брать $a_1 = 1$). Для 1-го полинома 9-й степени (таблица 3.2) приближения sin(x) погрешность $\delta_{MM} = 3,47 \cdot 10^{-9}$, а погрешности урезания констант исключены, поскольку они представлены 10-11-ю разрядными десятичными цифрами. Простое урезание разрядных цифр до 8 увеличивает значение погрешности результата с $\delta_p = \delta_{мм} = 3,47 \cdot 10^{-9}$ до $\delta_p = 2 \cdot 10^{-7}$, то есть в 58 раз. При проведении компенсации погрешность уменьшается в $2 \cdot 10^{-7} / 4,86 \cdot 10^{-9} = 41$ раз с сокращением разрядных сеток операндов на 5,5 двоичных разрядов [35].

3.1.2 Аппроксимация функции tg(x) полиномами наилучшего приближения. При вычислении более медленно сходящейся функции tg(x), приемлемым является набор полиномов наилучшего приближения с нечетными степенями на интервале $x \in [-\pi/4; \pi/4]$. Применение интервала $x \in [-\pi/2; \pi/2]$ усложняет задачу аппроксимации, т.к. при $x \to \pi/2$ $tg\alpha \to \infty$. Для уменьшения дискретного приращения числа операций до двух, как и для функции sin(x), можно использовать полиномы с $a_1 = 1$. При увеличении степеней полиномов для tg(x) на 2 (таблица 3.3) погрешность соответственно уменьшается примерно в 13,5 раз, что соответствует приращению только 1,25 двоичных цифры на одну операцию.

Таблица 3.3 Полиномы наилучшего приближения tg(x) на интервале $x \in [-\pi/4; \pi/4]$

Число членов	Формула полинома	δ_{MM}	$A+m$
1	$P(x) = 1{,}199 \cdot x$	$5{,}7 \cdot 10^{-2}$	2
2	$P(x) = x \cdot (0{,}9754 + 0{,}4742 \cdot x^2)$	$4{,}1 \cdot 10^{-3}$	6
3	$P(x) = x \cdot (1{,}00250 + (0{,}30341 + 0{,}21863 \cdot x^2) \cdot x^2)$	$2{,}9 \cdot 10^{-4}$	9
4	$P(x) = x \cdot (0{,}999767 + (0{,}338064 + (0{,}107861 +$ $+0{,}101688 \cdot x^2) \cdot x^2) \cdot x^2)$	$2{,}1 \cdot 10^{-5}$	12
5	$P(x) = x \cdot (0{,}9999999 + (0{,}3326998 + (0{,}1387526 +$ $+(0{,}0355390 + 0{,}0473359 \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2)$	$1{,}5 \cdot 10^{-6}$	15
6	$P(x) = x \cdot (0{,}99999825 + (0{,}33340931 + (0{,}13239304 +$ $+(0{,}05888562 + (0{,}00975583 + 0{,}02203145 \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2)$	$1{,}1 \cdot 10^{-7}$	18
7	$P(x) = x \cdot (1{,}0000001446 + (0{,}3333248797 + (0{,}1334756942 +$ $+(0{,}0529201231 + (0{,}0257646276 + (0{,}0013818002 +$ $+0{,}0102552909 \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2)$	$7{,}7 \cdot 10^{-9}$	21
8	$P(x) = x \cdot (0{,}99999998822 + (0{,}33333422186 + (0{,}13331382515 +$ $+(0{,}05415948685 + (0{,}02088553451 + (0{,}01168103091 +$ $+(-0{,}00082688714 + 0{,}00477347094 \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2)$	$5{,}6 \cdot 10^{-10}$	24
9	$P(x) = x \cdot (1{,}000000000942 + (0{,}333333244119 + (0{,}133335806795 +$ $+(0{,}053937201388 + (0{,}022079225067 + (0{,}008040731268 +$ $+(0{,}005501188927 + (-0{,}001067508728 +$ $+0{,}002221239931 \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2)$	$3{,}97 \cdot 10^{-11}$	27
10	$P(x) = x \cdot (0{,}9999999999288 + (0{,}3333333416382 + (0{,}1333330482297 +$ $+(0{,}0539727284831 + (0{,}0218310809196 + (0{,}0090598253379 +$ $+(0{,}0029696421758 + (0{,}0026721376132 + (-0{,}0008014409517 +$ $+0{,}0010291563402 \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2)$	$3{,}02 \cdot 10^{-12}$	30

Таблица 3.4. Полиномы вычисления tg(x) с разбиением на подинтервалы

<table>
<tr><th>№</th><th>Интервал</th><th>Формула полинома</th><th>δ_{MM}</th><th>$A+m$</th></tr>
<tr><td rowspan="2">1</td><td>$[0;0,526]$</td><td>$P(x)=1,1037x-0,0109$</td><td>0,0109</td><td rowspan="2">6</td></tr>
<tr><td>$[0,526;0,785]$</td><td>$P(x)=1,6164\cdot x-0,2802$</td><td>0,0106</td></tr>
<tr><td rowspan="2">2</td><td>$[0;0,544]$</td><td>$P(x)=0,99350\cdot x+0,39482\cdot x^3+0,00042$</td><td>$4,2\cdot 10^{-4}$</td><td rowspan="2">10</td></tr>
<tr><td>$[0,544;0,785]$</td><td>$P(x)=0,76357\cdot x+0,64909\cdot x^3+0,08539$</td><td>$4,1\cdot 10^{-4}$</td></tr>
<tr><td rowspan="2">3</td><td>$[0;0,5513]$</td><td>$P(x)=1,0003315\cdot x+0,3271278\cdot x^3+0,1688683\cdot x^5-$
$-1,57020353740948\cdot 10^{-5}$</td><td>$1,57\cdot 10^{-5}$</td><td rowspan="2">13</td></tr>
<tr><td>$[0,5513;0,785]$</td><td>$P(x)=1,0763362\cdot x+0,172760\cdot x^3+0,3161679\cdot x^5-$
$-0,0235526$</td><td>$1,53\cdot 10^{-5}$</td></tr>
</table>

Для проверки возможности уменьшения числа операций при вычислении функции $tg(x)$ на интервале $\left[0^0;45^0\right]$ введены два подинтервала с двумя полиномами 1-й-5-й степеней с примерно одинаковыми погрешностями δ_{MM} на подинтервалах (таблица 3.4). Обеспечено уменьшение погрешности δ_{MM} по сравнению с полиномом 1-й степени $P(x)=1,199\cdot x$ в $0,057/0,0109=5,22$ раза, для полиномов 3-й и 5-й степеней (таблица 3.4) обеспечено уменьшение погрешности соответственно в $4,1\cdot 10^{-3}/4,2\cdot 10^{-4}=9,76$ и $2,9\cdot 10^{-4}/1,57\cdot 10^{-5}=18,5$ раз при увеличении сложности алгоритма только на 2 операции. Для полинома 5-й степени с двумя подинтервалами по сравнению с полиномом 7-й степени (таблица 3.3) имеем меньшие значение погрешности, при небольшом увеличении затрат на хранение в памяти 2-х констант. Таким образом, применение схемы аппроксимации с двумя полиномами 5-й степени (таблица 3.4) при погрешности $1,57\cdot 10^{-5}$ может оказаться более предпочтительным, чем применение одного полинома 7-й степени с погрешностью $2,1\cdot 10^{-5}$. Обеспечена взаимная компенсация составляющих погрешностей, при которой погрешность уменьшается в 20 раз.

3.1.3 Аппроксимация функции arctg(x) полиномами наилучшего приближения. Для вычисления функции arctg(x) приемлемым является набор полиномов наилучшего приближения с нечетными степенями на интервале $x\in[-1;1]$ (таблица 3.5). При увеличении степеней полиномов на 2 погрешность уменьшается примерно в 8 раз, что соответствует приращению только 1 двоичной цифры на одну операцию.

Таблица 3.5 Полиномы наилучшего приближения arctg(x) на интервале $x \in [-1;1]$

№	Формула полинома	δ_{MM}
1	$P(x) = 0{,}83 \cdot x$	0,047
2	$P(x) = x \cdot (0{,}972 - 0{,}915 \cdot x^2)$	0,0049
3	$P(x) = x \cdot (0{,}9954 - (0{,}2888 + 0{,}0794 \cdot x^2) \cdot x^2)$	0,0006
4	$P(x) = x \cdot (0{,}99921 - (0{,}32115 + 0{,}14621 - 0{,}03895 \cdot x^2) \cdot x^2) \cdot x^2)$	$8{,}14 \cdot 10^{-5}$
5	$P(x) = x \cdot (0{,}999866 - (0{,}330305 + (0{,}180161 -$ $- (0{,}085154 + 0{,}020841 \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2)$	$1{,}14 \cdot 10^{-5}$
6	$P(x) = x \cdot (0{,}9999772 - (0{,}3326226 + (0{,}1935339 -$ $- (0{,}1164265 + 0{,}0526481 - 0{,}0117195 \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2)$	$1{,}66 \cdot 10^{-6}$
7	$P(x) = x \cdot (0{,}99999611 - (0{,}33317372 + (0{,}19807853 -$ $- (0{,}13233518 + (0{,}07962695 - (0{,}03360707 +$ $+ 0{,}00681273 \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2)$	$2{,}47 \cdot 10^{-7}$
8	$P(x) = x \cdot (0{,}999999336 - (0{,}333298667 + (0{,}199466383 -$ $- (0{,}139090075 + (0{,}096431906 - (0{,}055926150 +$ $+ (0{,}021872667 - 0{,}004057274 \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2) \cdot x^2)$	$3{,}75 \cdot 10^{-8}$

Таблица 3.6 Полиномы вычисления arctg(x) с разбиением на подинтервалы

№	Интервал	Формула полинома	δ_{MM}
1	$[0; 0{,}545]$	$P(x) = 0{,}915 \cdot x + 0{,}008$	0,008
	$[0{,}545; 1]$	$P(x) = 0{,}629 \cdot x + 0{,}163$	0,008
2	$[0; 0{,}574]$	$P(x) = 0{,}9929 \cdot x - 0{,}2632 \cdot x^3 + 0{,}0004$	$4{,}5 \cdot 10^{-4}$
	$[0{,}574; 1]$	$P(x) = 0{,}8582 \cdot x - 0{,}1261 \cdot x^3 + 0{,}0527$	$4{,}5 \cdot 10^{-4}$
3	$[0; 0{,}593]$	$P(x) = 0{,}99939 \cdot x - 0{,}32223 \cdot x^3 + 0{,}13383 \cdot x^5 +$ $+ 2{,}9465 \cdot 10^{-5}$	$2{,}94 \cdot 10^{-5}$
	$[0{,}593; 1]$	$P(x) = 0{,}94422 \cdot x - 0{,}22204 \cdot x^3 + 0{,}04487 \cdot x^5 +$ $+ 0{,}018371$	$2{,}94 \cdot 10^{-5}$

При вычислении функции arctg (x) в соответствии на интервале $[-1;1]$ также введены два подинтервала с двумя полиномами 1-й - 5-й степеней с одинаковыми погрешностями δ_{MM} на подинтервалах (таблица 3.6). Обеспечено уменьшение погрешности δ_{MM} по сравнению с полиномом 1-й степени $P(x) = 0{,}83 \cdot x$ в $0{,}047/0{,}008 = 5{,}9$ раза. По сравнению с полиномами 3-й и 5-й степеней (таблица

3.6) обеспечено уменьшение погрешности соответственно в $4{,}95\cdot10^{-3}/4{,}5\cdot10^{-4}=11$ и $6\cdot10^{-4}/2{,}94\cdot10^{-5}=20{,}4$ раз при увеличении сложности алгоритма только на 2 операции. Таким образом, целесообразно введение подинтервалов, поскольку приращение числа разрядных цифр на одну операцию будет больше чем 1.

Для функции arctg(x) также обеспечена возможность сокращения разрядных сеток операндов. Для полинома 9-й степени

$$P(x) = x\,(0{,}9998 + x^2\,(-0{,}3303 + x^2\,(0{,}1801 + x^2\,(-0{,}0851 + 0{,}0208\,x^2))))$$

с простым урезанием $\delta_p = 1{,}532\cdot10^{-5}$, а с компенсацией для полинома

$$P(x) = x\,(0{,}9998 + x^2\,(-0{,}3304 + x^2\,(0{,}1805 + x^2\,(-0{,}0851 + 0{,}0211\,x^2))))$$

$\delta_p = 1{,}8\cdot10^{-6}$, т.е. погрешность уменьшилась в $1{,}532\cdot10^{-5}/1{,}8\cdot10^{-6} = 8$ раз. Для полинома 15-й степени $P(x) = x\,(0{,}9999993 + x^2\,(-0{,}3332986 + x^2\,(0{,}1994663 + x^2\,(-0{,}1390900 + x^2 + (0{,}0964319 + +x^2\,(-0{,}0559261 + x^2\,(0{,}0218726 - 0{,}00450572\,x^2)))))))$ без компенсации погрешность $\delta_p = 1{,}7\cdot10^{-6}$, а с компенсацией для полинома

$$P(x) = x\,(0{,}9999993 + x^2\,(-0{,}3333012 + x^2\,(0{,}1994861 + x^2\,(-0{,}1391981+$$
$$+x^2 + (0{,}0967274 + x^2\,(-0{,}0563533 + x^2\,(0{,}0221851 - 0{,}0041481\,x^2)))))))$$

$\delta_p = 4{,}72\cdot10^{-8}$, т.е. погрешность уменьшилась в $1{,}7\cdot10^{-6}/4{,}72\cdot10^{-8} = 36$ раз.

3.1.4. Аппроксимация функции arcsin(x) полиномами наилучшего приближения. Для вычисления функции arcsin(x) одного полинома на всем интервале задания аргумента $x\in[-1;1]$ для диапазона максимальных значений погрешностей аппроксимации δ_{MM} в диапазоне от 0,15 до 0,0015 обеспечивает незначительное приращение числа значащих цифр результата на 1 операцию при последовательном увеличении степени полинома с 1-й до 9-й и выше. Поэтому для поиска набора полиномов с большим увеличением значений приращения числа двоичных цифр результата на одну операцию исследован интервал [0;0,707]. В соответствии с разработанными алгоритмами поиска полинома наилучшего приближения для функции arcsin(x) в интервале значений аргумента [0;0,707] при часто используемых на практике области значений угла $[0^0;45^0]$ для диапазона погрешностей δ_{MM} от $2{,}09851 \cdot 10^{-2}$ до $6{,}229703 \cdot 10^{-9}$ получены полиномы с нечетными степенями (таблица 3.7). Уменьшение погрешности для: полинома 3-й степени по сравнению с полиномом 1-й степени составит $20{,}98/1{,}6 = 13{,}2$, соответственно: полиномов 5-й степени по сравнению с 3-й степенью $16{,}02/1{,}628 = 9{,}81$, 7-й степени по сравнению с 5-й степенью $16{,}28/1{,}897 = 8{,}582$, 9-й степени по сравнению с 7-й степенью $18{,}97/2{,}392 = 7{,}94$, 11-й степени по сравнению с 9-й степенью $23{,}92/3{,}188 = 7{,}5$, 13-й степени по сравнению с 11-й степенью $31{,}88/4{,}39 = 7{,}26$, 15-й степени по сравнению с 13-й степенью $43{,}9/6{,}23 = 7{,}05$ [36].

В среднем имеем приращение порядка 1-й разрядной двоичной цифры на одну операцию. Произведено исследование возможности уменьшения дискрета приращения числа операций путем фактического исключения константы a_i в полиноме (принимаем $a_1 = 1$, полином №2` в таблице 3.7). В данном случае уменьшение числа операций на 1 обеспечивает уменьшение погрешности по сравнению с полиномом 1-й степени только в $20{,}98/7{,}96 = 2{,}62$ раза.

Таблица 3.7 Полиномы вычисления arcsin(x) на интервале $x \in [0; 0{,}707]$

№	Формулы полиномов		Максимальная погрешность	Количество операций	
				A+m	A
1	$a_0 = 0$ $a_1 \neq 1$	$P(x) = 1{,}08 \cdot x$	$2{,}0910^{-2}$	4	2
2	$a_0 = 0$ $a_1 \neq 1$	$P(x) = x \cdot (0{,}9895 + 0{,}2379 \cdot x^2)$	$1{,}6 \cdot 10^{-3}$	6	4
2`	$a_0 = 0$ $a_1 = 1$	$P(x) = x + 0{,}2379 \cdot x^3$	$7{,}96 \cdot 10^{-3}$	5	3
3	$a_0 = 0$ $a_1 \neq 1$	$P(x) = x \cdot (1{,}0015 + x^2 \cdot (0{,}1453 + 0{,}1454 \cdot x^2))$	$1{,}62 \cdot 10^{-4}$	9	6
4	$a_0 = 0$ $a_1 \neq 1$	$P(x) = x \cdot (0{,}99977 + x^2 \cdot (0{,}17219 + x^2 \cdot (0{,}04023 +$ $+0{,}11817 \cdot x^2)))$	$1{,}89 \cdot 10^{-5}$	12	8

Продолжение таблицы 3.7

№	Формулы полиномов		Максимальная погрешность	Количество операций	
				A+m	A
5	$a_0 = 0$ $a_1 \neq 1$	$P(x) = x \cdot (1{,}000036 + x^2(0{,}165359 + x^2(0{,}088178 +$ $+x^2(\text{-}0{,}007369 + 0{,}110056 \cdot x^2))))$	$2{,}39 \cdot 10^{-6}$	15	10
6	$a_0 = 0$ $a_1 \neq 1$	$P(x) = x \cdot (0{,}9999944 + x^2(0{,}1669578 + x^2(0{,}0707215 +$ $+x^2(0{,}0709638 + x^2(\text{-}0{,}0443618 + 0{,}1109995 \cdot x^2)))))$	$3{,}18 \cdot 10^{-7}$	18	12
7	$a_0 = 0$ $a_1 \neq 1$	$P(x) = x \cdot (1{,}00000089 + x^2(0{,}16660387 +$ $+ x^2(0{,}07626018 + x^2(0{,}03365015 +$ $+x^2 (0{,}07824441 + x^2 (\text{-}0{,}08285871 + 0{,}11811726\, x^2))))))$	$4{,}39 \cdot 10^{-8}$	21	14
8	$a_0 = 0$ $a_1 \neq 1$	$P(x) = x \cdot (0{,}999999855 + x^2(0{,}166679861 + x^2(0{,}074656217 +$ $+ x^2(0{,}048657133 + x^2 (0{,}005951529 + x^2 (0{,}104078691 +$ $+x^2 (\text{-}\ 0{,}128502199 + 0{,}130331661 \cdot x^2)))))))$	$6{,}22 \cdot 10^{-9}$	24	16

Для поиска полиномов с возможным большим увеличением значения приращения числа двоичных цифр результата на одну операцию, интервал [0;0,707] разбит на два подинтервала с двумя полиномами 1-й-7-й степеней с примерно одинаковыми погрешностями δ_{MM} на подинтервалах. (таблица 3.8). Для двух полиномов 1-й степени не обеспечено уменьшение погрешности δ_{MM} по сравнению с полиномом 3-й степени $P(x) = x \cdot (0{,}9895 + 0{,}2379 \cdot x^2)$ с одним интервалом интерполяции и одинаковым количеством операций равном 6. В тоже время, для двух полиномов 7-й степени с общим числом операций равном 16, максимальное значение погрешности составит $5{,}35 \cdot 10^{-7}$.

Таблица 3.8 Полиномы вычисления arcsin(x) с разбиением на подинтервалы

№	Интервал	Формула полинома	δ_{MM}
1	$[0;0{,}475]$	$P(x) = 1{,}042 \cdot x - 0{,}004$	$3{,}98 \cdot 10^{-3}$
	$[0{,}475;0{,}707]$	$P(x) = 1{,}251 \cdot x - 0{,}103$	
2	$[0;0{,}499]$	$P(x) = 0{,}0001 + x \cdot (0{,}9973 + 0{,}1968 \cdot x^2)$	$1{,}66 \cdot 10^{-4}$
	$[0{,}499;0{,}707]$	$P(x) = 0{,}0371 + x \cdot (0{,}8885 + 0{,}3389 \cdot x^2)$	
3	$[0;0{,}511]$	$P(x) = -8{,}92 \cdot 10^{-6} + x \cdot (1{,}00021 + x^2 \cdot (0{,}16238 +$ $+0{,}10272 \cdot x^2))$	$8{,}9 \cdot 10^{-6}$
	$[0{,}511;0{,}707]$	$P(x) = -0{,}01662 + x \cdot (1{,}05802 + x^2 \cdot (0{,}02646 + 0{,}25181 \cdot x^2))$	

Продолжение таблицы 3.8

№	Интервал	Формула полинома	δ_{MM}
4	[0;0,5175]	$P(x) = -5{,}36 \cdot 10^{-7} + x \cdot (0{,}999985 + x^2 \cdot (0{,}167201 +$ $+ x^2 \cdot (0{,}068752 + 0{,}071366 \cdot x^2)))$	$5{,}4 \cdot 10^{-7}$
	[0,5175;0,707]	$P(x) = 0{,}008190 + x \cdot (0{,}967809 + x^2 \cdot (0{,}272555 -$ $- x^2 (0{,}144085 + 0{,}248582 \cdot x^2)))$	

Для полинома 11-й степени (таблица 3.7) максимальное значение погрешности δ_{MM} составит $3{,}188161 \cdot 10^{-7}$ при количестве операций равном 18. Таким образом, разбиение интервала [0;0,707] на подинтервалы становится эффективным при значении максимальной погрешности порядка $5 \cdot 10^{-7}$ и менее.

При вычислении полинома в реальных условиях возникает дополнительная погрешность $\delta_{ЦУ}$ воспроизведения функции, обусловленная квантованием констант полинома с усечением разрядных сеток операндов

$$\delta_p = \delta_{MM} + \delta_{ЦУ},$$

где δ_p - погрешность результата.

Проверена возможность проведения взаимной компенсации составляющих погрешностей для уменьшения разрядных сеток операндов.

Поиск полиномов наилучшего приближения производится до таких значений числа разрядов операндов, когда их сокращение еще не оказывает существенного влияния на результирующую погрешность [16].

Для полинома 15-й степени

$$P(x) = x\,(0{,}999999853 + x^2(0{,}166679859 + x^2\,(0{,}074656216 + x^2\,(0{,}048657131 +$$
$$+ x^2\,(0{,}005951528 + x^2\,(0{,}104078692 + x^2\,(0{,}128502198 + 0{,}130331660\,x^2)))))))$$

без компенсации погрешность $\delta_p = 7{,}83 \cdot 10^{-8}$, а с компенсацией (рис. 3.4 б) для полинома

$$P(x) = x\,(0{,}999999855 + x^2\,(0{,}166679861 + x^2\,(0{,}074656217 + x^2\,(0{,}048657133 +$$
$$+ x^2\,(0{,}005951529 + x^2\,(0{,}104078691 + x^2\,(0{,}128502199 + 0{,}130331661 x^2))))))),$$

$\delta_{\text{р}} = 6{,}2 \cdot 10^{-9}$ т.е. погрешность уменьшилась в $7{,}83 \cdot 10^{-8} / 6{,}2 \cdot 10^{-9} = 12$ раз.

Введение сокращенного интервала для значений угла от 0^0 до 45^0 обеспечивает приращение одной двоичной цифры результата на одну операцию, что примерно в 3 раза больше приращения для значений углов от 0^0 до 90^0. При взаимной компенсации составляющих погрешностей результата обеспечена возможность уменьшения разрядных сеток операндов для полиномов 7-й степени и выше примерно на 2-3 двоичных разряда.

3.2 Разработка программ моделирования алгоритмов аппроксимации

3.2.1 Разработка программы аппроксимации элементарных функций. Прежде, чем приступить к разработке программного продукта, необходимо описать функциональную модель программного обеспечения оптимизации быстродействующих методов вычисления стандартных функций в информационно-вычислительных системах контроля.

CASE-средства поддерживают концептуальное проектирование, позволяют осуществить логическое и физическое проектирование. Среди инструментов CASE-средств широкое применение получили такие программы как ERwin, BPwin, S-Designor, CASE.Аналитик. В данном случае в качестве инструмента CASE-средств для создания функциональной модели выбрана программа Erwin process modeller версии 7.3.

ERwin автоматизирует задачи, связанные с построением моделей, обеспечивая семантическую строгость, необходимую для гарантирования правильности результатов, а также целостность и непротиворечивость модели, которые гарантируются применением методологий IDEF 0 (Business Process), IDEF 3 (Process Flow) и DFD (Data Flow Diagram). Каждая из этих трех нотаций, поддерживаемых в ERwin, позволяет рассмотреть различные стороны деятельности предприятия.

При использовании программы ERwin 7.3 были созданы диаграмма декомпозиции для описания функциональной модели разрабатываемой программы.

Полученные диаграммы (рис. 3.4, 3.5) представляют собой два блока «Определение коэффициентов полинома» и «Урезание разрядных сеток». Слева показаны необходимые входные данные для расчета, справа – выходные.

Исходными входными данными для расчета являются:

- вид аппроксимируемой функции;
- количество членов полинома – n;
- границы интервала аппроксимации - $x \in [a, b]$;
- используемые члены – четные, нечетные, все;
- количество разрядных сеток операндов.

Выходными данными являются скомпенсированные коэффициенты полинома, максимальная погрешность, а также для наглядности требуется выводить график погрешности. Также программа предполагает математические расчеты, требуется руководствоваться математическими правилами и алгоритмами.

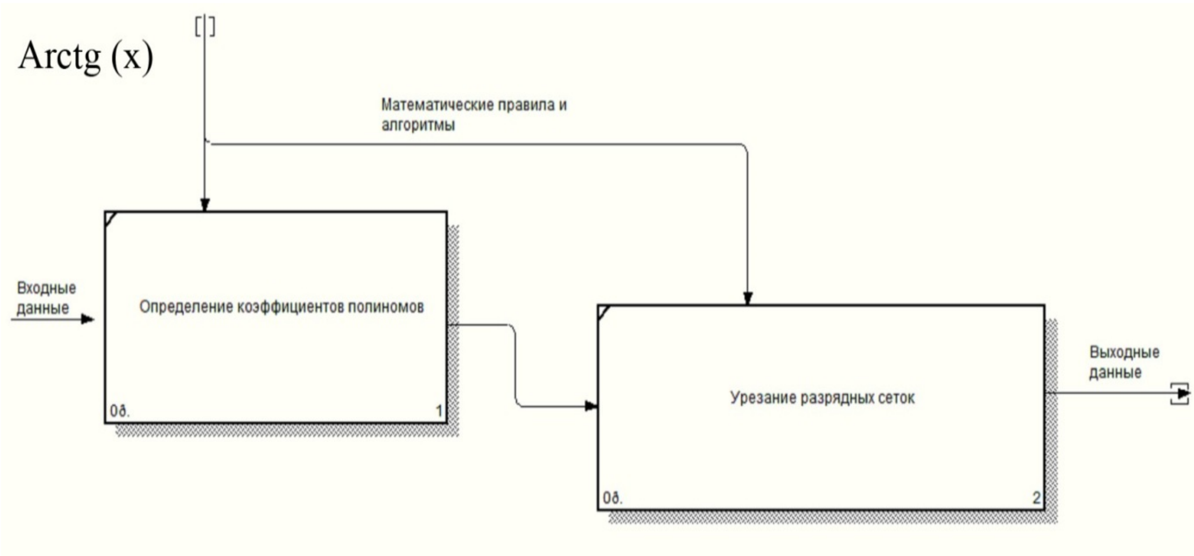


Рисунок 3.4 Декомпозиция функциональной диаграммы для arctg(x)

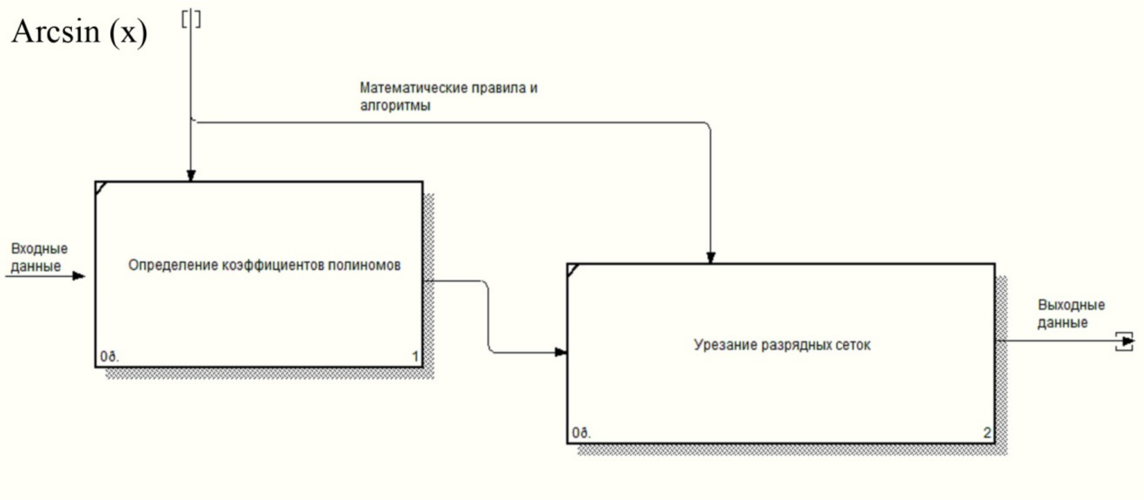


Рисунок 3.5 Декомпозиция функциональной диаграммы для arcsin(x)

3.2.2 Разработка программы поиска алгоритма наилучшего приближения функции $R=\sqrt{A^2+B^2}$ ортогональных составляющих в амплитуду и фазу. Разработаны быстродействующие алгоритмы воспроизведения функций $R=\sqrt{A^2+B^2}$ (рис.3.6), $\beta=\mathrm{arctg}(A/B)$ с оптимизацией вычислительного процесса по точностным характеристикам, быстродействию и программно-аппаратным затратам на примерах измерений амплитуд сигналов с неизвестной начальной фазой, напряжений переменного тока и фазовых углов. Для значений относительной погрешности результата от 4% до 0,26%, по сравнению с алгоритмами с непосредственным вычислением квадратного корня обеспечен выигрыш по быстродействию в 2…2,75 раза. При этом не происходит как переполнения разрядной сетки, так и исчезновения порядка чисел A и B. Для ускоренного нахождения значения $R=\sqrt{A^2+B^2}$ используются приближенные алгоритмы с проверкой условий соотношения A и B с обеспечением последовательного максимального приращения количества значащих цифр представления результата (M) при соответствующем ему минимальном увеличении числа вычислительных операций N и обращений к памяти P [18,35].

Алгоритм с одним условием проверки соотношения обеспечивает максимальную относительную погрешность $\delta_{\text{мо}}$=4,08% при 6-ти операциях N+P: сравнение, умножение, сложение и извлечение констант. Следует отметить, что только вычисление на интервале $x\in[0;1]$ с погрешностью 4,6% полинома наилучшего приближения $\sqrt{x} = 0{,}0459 + x(2{,}866 - x(4{,}172 - 2{,}305x))$ требует реализации 10 операций: 6 N и 4 P. При двух условиях (рис.3.6) сравнения соотношений получены полиномы аппроксимирующие функцию $R=\sqrt{A^2 + B^2}$ с погрешностью $\delta_{мо}$ менее 1,4%. При реализации алгоритма необходимо выполнить 9 N+P операций и хранить в памяти 5 констант. По сравнению с алгоритмом с одним условием при увеличении числа операций на 3 погрешность $\delta_{мо}$ уменьшается в 2,9 раза. При введении 3-х условий (рис.3.7) погрешность $\delta_{мо}$ уменьшилась до 0,5%, что в 2,8 раза меньше чем для алгоритма с двумя условиями. Число операций N+P на самой длинной ветви реализации алгоритма соответствует 12, а в памяти необходимо хранить 8 констант. Для алгоритма с 4-мя условиями погрешность не превышает 0,26% при приближенном воспроизведении функций $R = \sqrt{A^2 + B^2}$ за 15 операций. При этом в памяти необходимо хранить 11 констант. При увеличении числа операций N+P на 3 погрешность уменьшилась почти в 2 раза. Дальнейшее увеличение количества условий усложняет реализацию алгоритма и для систем с более высокими классами точности предпочтительнее использовать непосредственные методы вычисления квадратного корня.

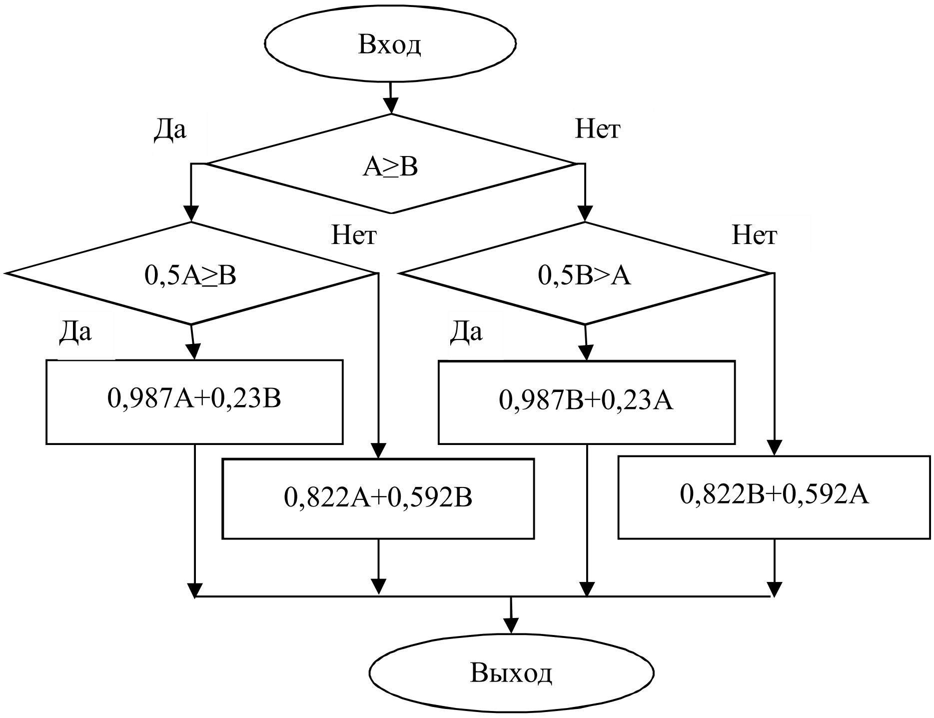


Рисунок 3.6 Структурная схема алгоритма вычисления функции $\mathrm{R} = \sqrt{\mathrm{A}^2 + B^2}$ с двумя условиями

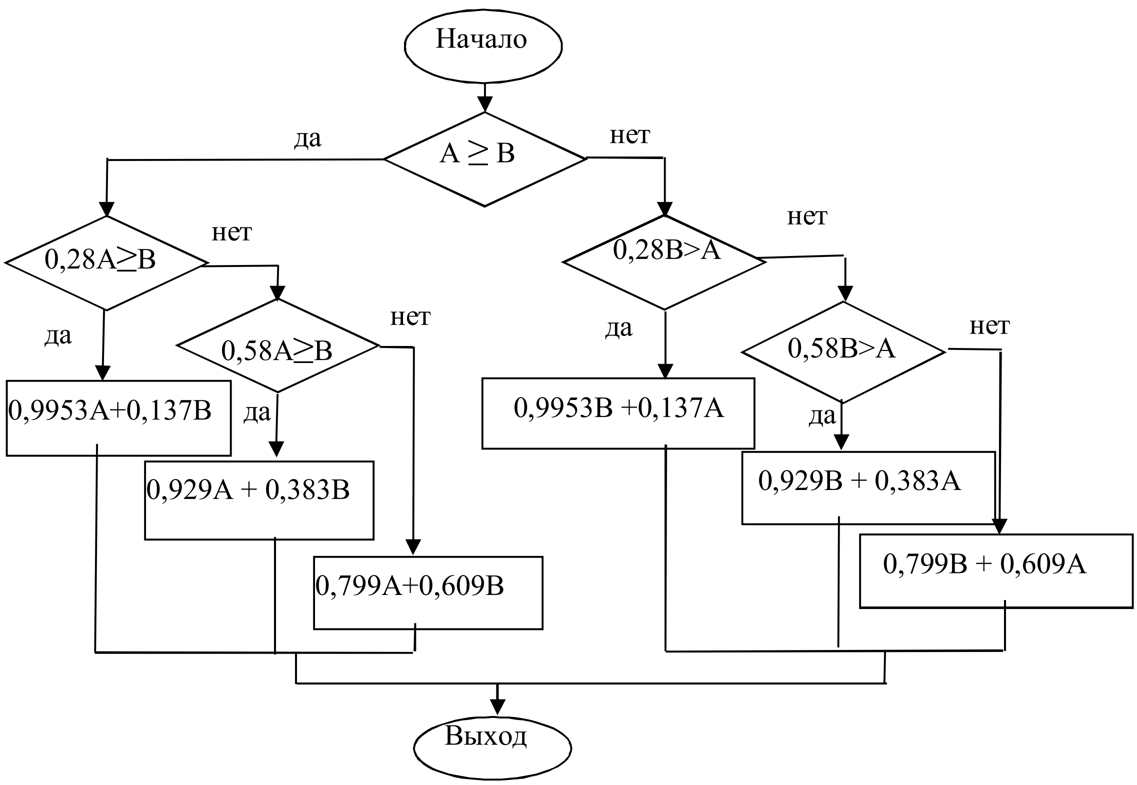


Рисунок 3.7 Структурная схема алгоритма вычисления функции $R = \sqrt{A^2 + B^2}$ с тремя условиями

Операция деления A/B заменяется операцией умножения числителя на обратную величину знаменателя: $A/B = A\times(1/B)$. Функция $F(B)=1/B$ аппроксимируется в диапазоне $B\in[2^{-10};1]$ тремя полиномами наилучшего приближения на трех подинтервалах с примерно равными абсолютными максимальные значения погрешностей. Измерение фазы с использованием предварительно определенных подинтервалов аппроксимации вычисления амплитуды сигнала по ортогональным составляющим обеспечивает уменьшение погрешности примерно в 10 раз при фиксированных остальных критериях вычислительного процесса [18].

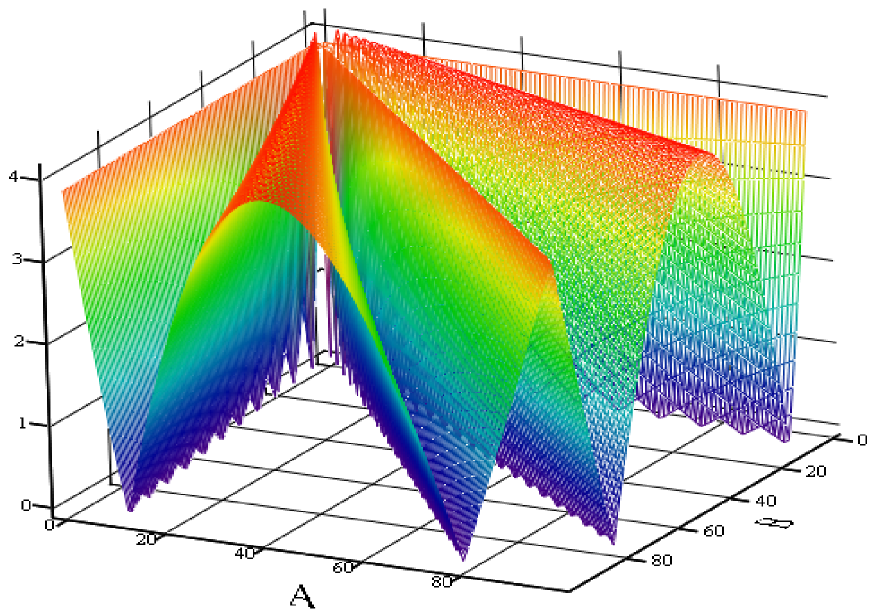


Рисунок 3.8 График погрешности алгоритма вычисления функции $R=\sqrt{A^2+B^2}$

Выводы к главе 3

1. Для решения широкого класса прикладных задач разработаны улучшенные вычислительные алгоритмы воспроизведения стандартных функций, созданы алгоритмы вычисления тригонометрических функций sin(x), tg(x), arctg(x), arcsin(x), решения типовых задач с устранением избыточной точности путем обеспечения дискретного приращения 2… 3… и более значащих двоичных цифр результата при фиксированном возрастании сложности алгоритма не более чем на 2-3 операции в диапазоне представления выходных данных 3...32

двоичными разрядами. Обеспечено повышение быстродействия в 2 раза при фиксированных точностных характеристиках и программно-аппаратных затратах.

2. Для функции sin(x) при увеличении степени полинома на 2 получаем хорошие нарастающие приращения числа разрядных цифр на одну операцию: ΔE =1,06. Целесообразно для аппроксимации функции sin(x) в диапазоне погрешностей $10^{-1}-10^{-11}$ и ниже использовать полиномиальную аппроксимацию на интервале $x\in[0;\pi/2]$. Для уменьшения дискрета приращения числа операций целесообразно исключать константу a_1 (брать $a_1=1$). Простое урезание разрядных цифр до 8 увеличивает значение погрешности результата в 58 раз, а при проведении компенсации погрешность уменьшается в 41 раз, что соответствует сокращению разрядных сеток операндов на 5,5 двоичных разрядов.

3. При вычислении более медленно сходящейся функций tg(x), приемлемым является набор полиномов наилучшего приближения с нечетными степенями на интервале $x\in[-\pi/4;\pi/4]$. Применение интервала $x\in[-\pi/2;\pi/2]$ усложняет задачу аппроксимации, т.к. при $x\to\pi/2$ $tg\alpha\to\infty$. Применение схемы аппроксимации с двумя полиномами 5-й степени при погрешности $1{,}57\cdot10^{-5}$ является более предпочтительным, чем применение одного полинома 7-й степени с погрешностью $2{,}1\cdot10^{-5}$. Обеспечена взаимная компенсация составляющих погрешностей, при которой погрешность уменьшается в 20 раз.

4. Для вычисления функции arctg(x) приемлемым является набор полиномов наилучшего приближения с нечетными степенями на интервале $x\in[-1;1]$. При увеличении степеней полиномов на 2 погрешность уменьшается примерно в 8 раз, что соответствует приращению только 1 двоичной цифры на одну операцию. По сравнению с функцией sin(x) для функции arctg(x) целесообразно введение подинтервалов, поскольку приращение числа разрядных цифр на одну операцию будет больше чем 1. Обеспечено сокращение разрядных сеток операндов. Для полинома 15-й степени с компенсацией $\delta_p=4{,}72\cdot10^{-8}$, погрешность уменьшилась в 36 раз.

5. При вычислении функции arcsin(x) одним полиномом на всем интервале задания аргумента $x \in [-1;1]$ для диапазона погрешностей δ_{MM} от 0,15 до 0,0015 обеспечивается незначительное приращение числа значащих цифр результата на 1 операцию при последовательном увеличении степени полинома с 1-й до 9-й и выше. Поэтому для поиска набора полиномов с большим увеличением значений приращения числа двоичных цифр результата на одну операцию исследован интервал [0;0,707]. Обеспечено и уменьшение дискрета приращения числа операций путем фактического исключения константы a_i в полиноме. Разбиение интервала [0;0,707] на подинтервалы становится эффективным при значении максимальной погрешности порядка $5 \cdot 10^{-7}$ и менее. При взаимной компенсации составляющих погрешностей результата обеспечено уменьшение разрядных сеток операндов для полиномов 7-й степени и выше примерно на 2-3 двоичных разряда.

6. Алгоритмы преобразования ортогональных составляющих в амплитуду для значений относительной погрешности результата от 4% до 0,26%, по сравнению с алгоритмами с непосредственным вычислением квадратного корня обеспечивают выигрыш по быстродействию в 2…2,75 раза. При этом не происходит как переполнения разрядной сетки, так и исчезновения порядка чисел A и B.

7. Разработано программное обеспечение: программа аппроксимации элементарных функций, а также программа поиска алгоритма наилучшего приближения функции $R = \sqrt{A^2 + B^2}$ ортогональных составляющих в амплитуду и фазу.

Глава 4 Разработка алгоритма воспроизведения траекторий движения воздушных объектов

4.1 Отличие используемого подхода воспроизведения траекторий движения от существующих

Целью разработки нового подхода к формированию траекторий движения воздушных объектов является создание более адекватной реальной ситуации трассы полета в трехмерном пространстве без скачков скоростей и ускорений. Это позволит существенно повысить качество подготовки операторов, обеспечить более эффективный функционально-диагностический контроль аппаратуры ИИС. Поставленная цель достигается тем, что предлагается использовать более эффективный метод сегментации траекторий – на основе гладко совмещенных кривых различных степеней, построенных на общей опорной ломаной, которая задана группой из последовательно расположенных отрезков прямых. Для полученных кинематических параметров движения объекта по заданной траектории предписан и скоростной режим движения таким образом, чтобы определить текущую точку на трассе, в которой будет находиться объект в произвольный момент времени. С этой целью параллельно с траекторией задаются и уравнения скорости $V(t_p)$ на основе ее значений в узловых точках и соответствующих производных, представляющих ускорения [37]. Для связывания между собой геометрии (кусочно-заданной кривой) и кинематики (профиль скорости) движения используется, объединяющий их параметр – пройденный путь S. Предусмотрена возможность контроля скорости, тангенциального ускорения и общей переносимой перегрузки в каждой точке пути.

Для достижения технического результата положение и геометрическое представление траектории движения объекта в пространстве плавно комплексируется из последовательно сопрягаемых сегментов с итерационными, наглядными графическими интуитивными изменениями их формы и кривизны, расчетом параметрических уравнений движения в каждом сегменте по трем координатам $x(t)$, $y(t)$, $z(t)$ зоны обзора ИИС в функции безразмерного параметра $t \in [0;1]$ с преобразованием его при различных законах изменений линейной скорости движения в функцию от аргумента - линейно–нарастающего временного интервала реального времени t_p воспроизведения траектории путем определения проходимого по сегментам пути $S(t_p)$, вычисления в соответствии с обратной функцией значений $t(S)$ в каждом сегменте и последующим вычислением в блоке расчета координат по параметрическим уравнениям движения значений текущих декартовых координат объекта.

В предлагаемом методе:

- геометрическая форма каждого сегмента общей траектории движения гибко выстраивается на основе итерационного изменения расположения координат n опорных точек ($\{P_i\} = \{x_i, y_i, z_i\}, i = 0 \ldots n-1$) опорной ломаной линии кривой Безье сегмента с n узлами, автоматическим расчетом параметрических уравнений движения в каждом сегменте по трем координатам $x(t)$, $y(t)$, $z(t)$ зоны обзора радиолокационной станции в функции безразмерного параметра $t \in [0;1]$, по параметрическим уравнениям определяется максимальный путь по сегменту $S_{\text{мах}}$ – длина сегмента, минимальный радиус кривизны, обеспечивающий проверку и исключение критических перегрузок при максимальных линейных скоростях движения.

- переход с одного сегмента кривой Безье на другой производится с плавным изменением радиуса кривизны, обеспечением непрерывности первой и второй производных двух сопрягаемых кривых, когда для сопрягаемых сегментов из кубической параболы и отрезка прямой или двух кубических парабол соответственно их пять смежных вершин лежат на одной прямой или составляют выпуклый многоугольник;

- в соответствии с параметрическими уравнениями каждого сегмента в функции безразмерного параметра t вычисляется путь, пройденный по параметрически заданной кубической кривой в функции $S(t)$, воспроизводится обратная функция $t(S)$ с последующей аппроксимацией ее полиномом наилучшего приближения $g(S)$;

- по заданному закону воспроизведения скорости на заданном сегменте определяются текущее значение пройденного пути $S(t_{p1})$ в функции текущего временного интервала t_{p1} от начала сегмента и времени движения по сегменту, время прохождения сегмента;

- по общему, связывающему параметру текущего пути S на сегменте осуществляется переход от времени t_{p1} к параметру t с последующей подстановкой его в параметрические уравнения вычисления прямоугольных координат,

- предварительные действия оператора, направленные на проектирование модели движения, подготовительные расчеты обеспечивают в общем временном интервале формирования траектории движения определить моменты времени начала, продолжительности и окончания движения объекта по каждому сегменту, которые позволяют идентифицировать именно тот сегмент, по которому перемещается объект в момент времени t_p воспроизведения каждого сегмента с присвоением ему порядкового номера, к каждому порядковому номеру сегмента привязывается таблица уравнений движения на заданном сегменте.

Данный подход сочетает в себе предварительные действия оператора, направленные на проектирование модели движения, подготовительные расчеты, необходимые для определения траекторий движения, а также расчеты, вычисляющие мгновенные значения координат движущегося объекта в реальном масштабе времени. В таблице 4.1 представлено сравнение различных методов формирования траекторий воздушных объектов. Эти процессы можно представить исходя из обоснования моделей воспроизведения траекторий [38].

Таблица 4.1 Характеристики методов формирования траекторий движения воздушных объектов

Методы задания траекторий	При помощи полиномов	Сплайн-интерполяция	Кривые Безье
Поддерживаемые виды движений	Равномерное, равноускоренное	Равномерное, равноускоренное, неравноускоренное	Равномерное, равноускоренное, неравноускоренное
Поддержка фрагментации	+		
Обеспечение плавности траектории	-	+	+
Контроль параметров движения	-	+	+
Быстрота и наглядность формирования траектории	-	-	+

4.2 Математические модели воспроизведения геометрии трассы движения объекта. Параметрические уравнения для воспроизведения геометрической формы кривой

4.2.1 Описание кривых Безье. Рассмотрим более подробно предлагаемый метод формирования траекторий движущихся объектов. Параметрическое представление кривой Безье имеет вид

$$B(t) = \sum_{i=0}^{n} P_i B_i^n(t), \qquad (4.1)$$

где n – степень кривой; i – порядковый номер опорной вершины; P_i –вектор координат i-й опорной точки; $B_i^n(t) = \frac{n!}{i!(n-i)!} \cdot t^i \cdot (1-t)^{n-i}$ - полином Бернштейна степени n, t – безразмерный параметр, расположенный в интервале $t \in [0;1]$ [39].

В качестве траектории движения объекта зададим кусочно-заданную пространственную кривую, состоящую из плавно совмещаемых сегментов в виде кривых Безье (4.1) преимущественно первого, второго и третьего порядков. Совокупность такого набора сегментов позволяет представить прямолинейные участки траектории, участки с ненулевой кривизной, участки с ненулевой кривизной и ненулевым кручением – и таким образом описать различные виды маневра воздушного объекта. Использование кривой Безье порядка выше третьего в составе траектории существенно не развивает ее свойства, но усложняет аналитические выражения для расчета мгновенных координат движущегося объекта [40]. Геометрическая форма каждого сегмента общей траектории движения выстраивается на основе расположения n опорных точек ($\{P_i\} = \{x_i, y_i, z_i\}, i = 0...n-1$), т.е. опорной ломаной линии с n узлами.

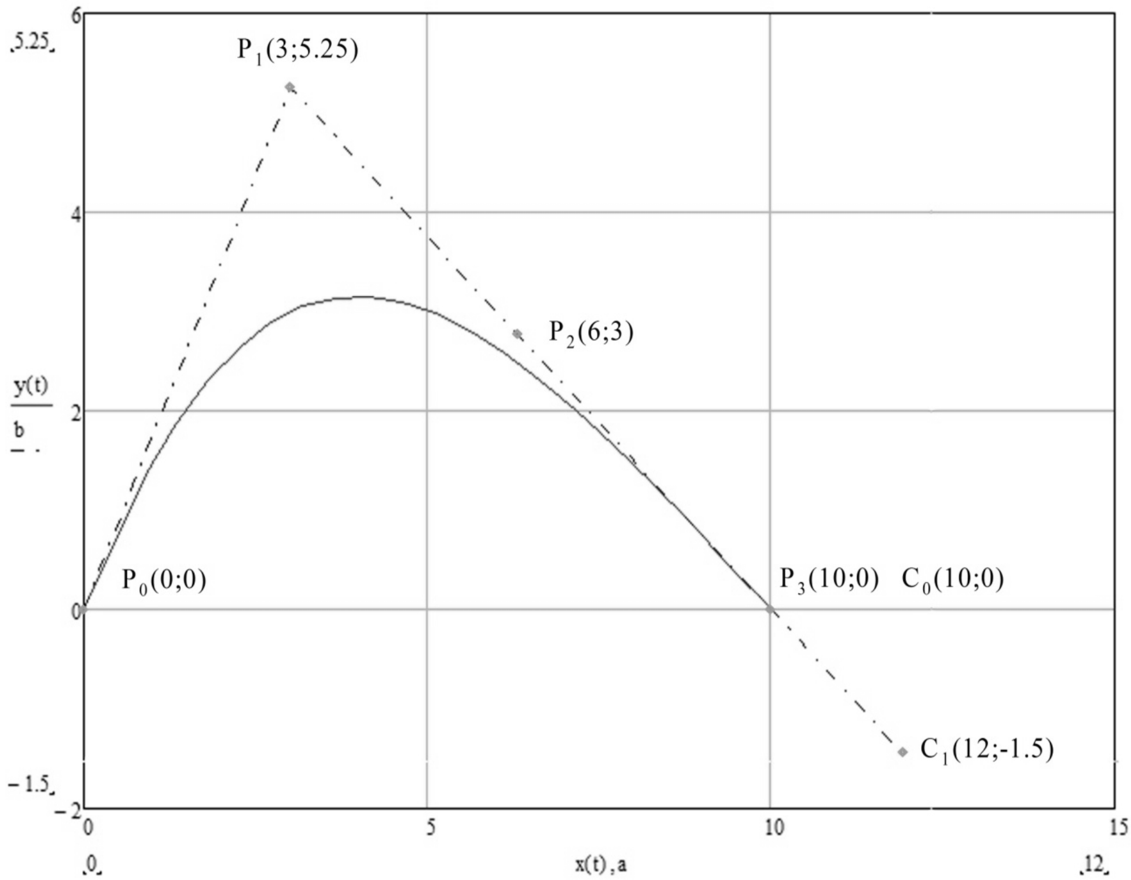


Рисунок 4.1 Сопряжение кривой Безье 3-й степени с отрезком прямой линии

На рис. 4.1 штрих-пунктирной кривой показана заданная оператором кривая, а сплошная кривая – результат представления кривой Безье.

4.2.2 Определение условий сопряжения сегментов траекторий. Конкретное расположение смежных точек стыка на отдельных сопрягаемых сегментах необходимо выбрать таким образом, чтобы получить в целом гладкую и гибкую пространственную траекторию, управляемую точками P_i, положение объекта на каждом сегменте которой определяется как безразмерным нормированным параметром $t \in [0;1]$ (рис. 4.1), так и реальным моментом времени. На рис.4.1 приведена траектория, состоящая из двух сегментов в виде кривых Безье 3-й и 1-й степеней, формируемых в соответствии с (4.1) векторами координат точек $P_0(0;0)$, $P_1(3;5{,}25)$; $P_2(6;3)$, $P_3(10;0)$ и C_0 (10;0) с C_1 (12;-1,5). Точки $P_3(10;0)$ и C_0 (10;0) сопрягаемых сегментов кубической параболы и отрезка прямой имеют одинаковые координаты на плоскости. Для обеспечения плавного перехода с одной кривой Безье на другую обеспечено плавное изменение радиуса кривизны, что выполнимо при непрерывности первой и второй производных сопрягаемых кривых. Для обеспечения непрерывности первой производной в соответствии с (4.1) достаточно, чтобы три смежные опорные точки двух кривых (P_2, P_3, C_0, C_1) лежали на одной прямой, тогда при построении траектории две сопрягаемые кривые будут иметь общую касательную в точке стыка $C_0(10;0)$, т.е. равные первые производные. Для обеспечения непрерывности второй производной необходимо, чтобы пять смежных вершин двух кривых лежали на одной прямой или составляли выпуклый многоугольник для сохранения непрерывности на стыке. В данном случае достаточно 4-х точек P_1, P_2, P_3, C_0, C_1, поскольку точки P_3, C_0 общие для двух сегментов Данное требование существенно ограничивает множество кривых; поэтому на практике для соблюдения непрерывности вторых производных при сопряжении можно использовать полиномиальные кривые и более высокого порядка или несколько кубических кривых.

4.2.3 Сопряжение кубической кривой с отрезком прямой. Для построения произвольных траекторий движения объектов часто необходимо обеспечить выполнение сопряжения отрезка прямой и кривой Безье, а также сопряжение кривых Безье произвольных форм с исключением скачков скорости и допустимыми значениями суммарного ускорения, определяемого как векторная сумма центростремительного $a_ц$ и тангенциального $a_т$ ускорений. Например, широко распространенный переход объекта с прямолинейной на криволинейную траекторию и обратно не должен сопровождаться скачком центростремительной силы:

$$F_ц = mV^2 / R_k = ma = mgn_ц ,$$

где R_k – радиус кривизны; V – линейная скорость по кривой; $g\ n_ц$ – перегрузка (g=9,8 м/с^2 - ускорение свободного падения, $n_ц$ – числовой коэффициент). Если известны допустимая перегрузка и линейная скорость объекта при его движении по криволинейной траектории, то можно рассчитать минимальный радиус кривизны дуги: $R_{k\min} = V^2 / (gn_{ц\max})$. Так, при скорости самолета V=1000 м/с и максимальной переносимой пилотом перегрузке $g\ n_{ц\max}$=8g минимальный радиус кривизны составит: $R_{k\min} = 12,74$ км. Т.е. на предельных для самолета скоростях радиус кривизны на переходной кривой не может быть меньше 13 км.

Если радиус кривизны будет иметь меньшее значение, то необходимо менять опорные точки сопрягаемых сегментов траектории в (4.1) таким образом, чтобы обеспечить более плавный переход или вводить кривые 4-й степени, или вместо одной кривой Безье использовать несколько кривых. Таким образом, при задании геометрической формы кривой необходимо одновременно определять и минимальный радиус ее кривизны. Следует также предусмотреть и запас на величину допустимого линейного ускорения, которое векторно суммируется с тангенциальным. Для данного случая сопряжения достаточно, чтобы три точки кубической кривой Безье (рис.4.1) P_1(3;5,25); P_2(6;3), P_3(10;0) находились на одной прямой с точками C_0 и C_1, задающими уравнение отрезка прямой. В соответствии с (4.1) при $t \in [0;1]$ уравнение кривой Безье третьего порядка будет иметь вид:

$$B(t) = (1-t)^3 P_0 + 3t \cdot (1-t)^2 P_1 + 3t^2 (1-t) P_2 + t^3 P_3 \tag{4.2}$$

Для опорных точек P_0(0;0), P_1(3;5,25); P_2(6;3), P_3(10;0) (рис.4.1) в соответствии с (4.2) определим параметрические уравнения траектории при сопряжении отрезка прямой и кривой Безье

$$\begin{aligned} x(t) &= 9t + t^3, \\ y(t) &= 15,75t - 22,5t^2 + 6,75t^3. \end{aligned} \tag{4.3}$$

Первая и вторая производные функций $x(t)$, $y(t)$ равны

$$x'(t) = (9t + t^3)' = 9 + 3t^2,$$

$$y'(t) = (15,75t - 22,5t^2 + 6,75t^3)' = 15,75 - 45t + 20,25t^2,$$

$$x''(t) = (9 + 3t^2)' = 6t,$$

$$y''(t) = (15,75 - 45t + 20,25t^2)' = 40,5t - 45.$$

Первая производная функции y по x равна

$$y_x' = \frac{y'(t)}{x'(t)} = \frac{15,75 - 45t + 20,25t^2}{9 + 3t^2}.$$

При $t = 1$

$$y_x'(1) = \frac{15,75 - 45 \cdot 1 + 20,25 \cdot 1^2}{9 + 3 \cdot 1^2} = -\frac{9}{12} = -0,75.$$

Вторая производная y по x определяет радиус кривизны

$$y_x'' = \frac{y''(t) \cdot x'(t) - y'(t) \cdot x''(t)}{(x'(t))^3} = \frac{((40,5t - 45) \cdot (9 + 3t^2)) - ((15,75 - 45t + 20,25t^2) \cdot (6t))}{((9 + 3t^2))^3}.$$

В точке $t = 1$

$$y_x''(1) = \frac{-54 + 54}{144 \cdot 12} = 0.$$

Таким образом, в конечной точке кубической кривой Безье $P_3(10;0)$ имеем нулевой радиус кривизны $R = \infty$.

В соответствии с параметрическим уравнением линейной кривой Безье $B(t) = (1 - t)P_0 + t \cdot P_1$, сопрягаемого отрезка прямой с координатами $C_0(10;0)$ и $C_1(12;-1,5)$, лежащей на одной прямой с точками P_1 и P_3 получим параметрические уравнения по каждой из координат

$$x(t) = 10 + 2t,$$
$$y(t) = -1,5t.$$

Уравнение прямой $y = f(x)$ в прямоугольной системе координат будет иметь вид

$$y = 7,5 - 0,75x,$$

т.е. действительно, имеем совпадение первых производных сопрягаемой кубической кривой и отрезка прямой. И визуально из рис.4.1 следует, что имеем плавный переход от отрезка прямой с радиусом кривизны $R = 0$ до некоторого фиксированного минимального радиуса кривой Безье. Для определения максимальной перегрузки необходимо для кубической кривой определить минимальный радиус. Функция изменения радиуса кривизны от времени определяется в соответствии с выражением [41]

$$R(t) = \frac{((x'(t))^2 + (y'(t))^2)^{\frac{3}{2}}}{y''(t) \cdot x'(t) - y'(t) \cdot x''(t)} = \frac{(((9+3t^2))^2 + (15,75 - 45t + 20,25t^2)^2)^{\frac{3}{2}}}{((40,5t-45)\cdot(9+3t^2)) - ((15,75-45t+20,25t^2)\cdot(6t))}.$$

Построим график зависимости $R(t)$ (рис.4.2), по которому определим минимальный радиус кривизны, значение которого равно $R_{\min} = 30$.

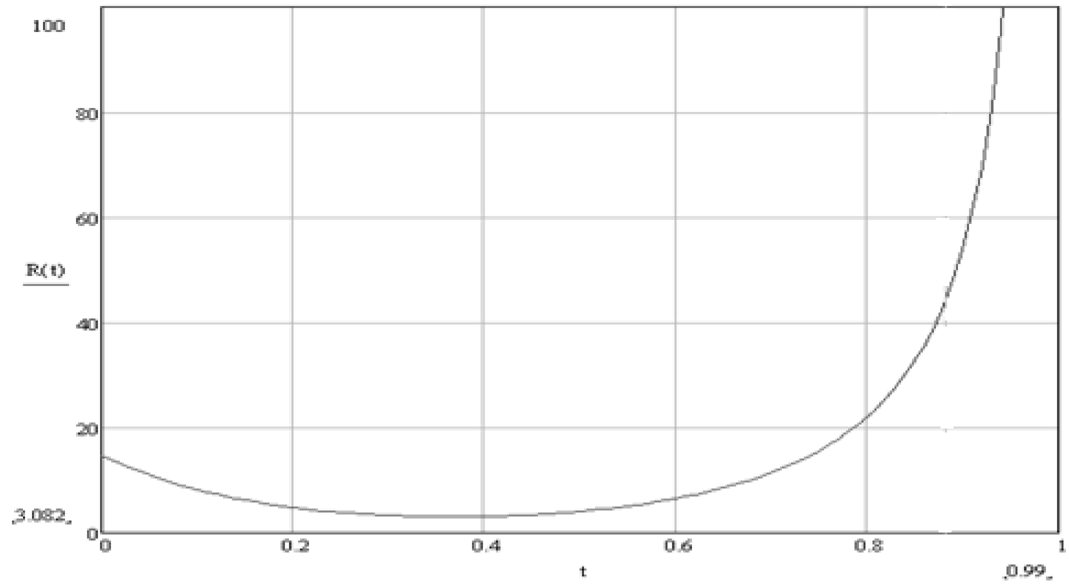


Рисунок 4.2 График для определения минимального радиуса кривизны

Данное требование ограничивает множество кривых; поэтому на практике для соблюдения непрерывности вторых производных при сопряжении можно использовать полиномиальные кривые и более высокого порядка или несколько кубических кривых.

Имитация траектории движения должна производиться в реальном масштабе времени, начиная с некоторого начального значения времени воспроизведения траектории $T_{нач.}$ =0. После чего текущее значение времени можно представить в виде линейно нарастающей непрерывной или решетчатой функции времени $t_p = kT_m$ с некоторым дискретом T_m. В тоже время в соответствии с (4.1) кривая Безье для любого сегмента является функцией безразмерного нормированного параметра $t \in [0;1]$, который будем трактовать как некоторую функцию нормированного времени $t \in [0;1]$. Необходимо связать нормированное время с реальным определенной функциональной зависимостью. Например, для отдельных сегментов траектории с разной длиной сегмента или пути $S_{мах}$ при постоянной скорости движения текущее значение пути в функции времени прохождения по сегменту изменяется по линейному закону. Таким образом, для различных сегментов траектории с разной длиной пути $S_{мах}$ и, при постоянной линейной скорости движения на этих участках необходимо перенормировать безразмерный параметр $t \in [0;1]$ к реальному значению времени $t_{р}$ прохождения каждого заданного сегмента исходя из необходимого условия, что каждому максимальному нормированному значению t=1 будет соответствовать реальное время прохождения сегмента $t_{мах}=S_{мах}/V$. Чтобы получить полную эффективную кинематику движения объекта по траектории в функции реального времени при изменении линейной скорости необходимо исследовать наиболее рациональные методы получения функциональных зависимостей нормированного значения времени t от аргумента $t_{р}$ действительного текущего значения времени. Однозначно нормированное и реальное время связывает текущее значение пройденного пути S. Путь S и (или) текущее реальное значение времени могут быть связующим параметром для определения текущих нормированных значений времени, подставляемых в параметрические уравнения для воспроизведения текущих значений координат траектории (4.3).

В соответствии с вышеизложенным, предварительно определим функциональные зависимости: $S(t)$, длины дуг - $S_{мах}$, пройденных по заданным сегментам, обратные функции $t(S)$, а также их аппроксимации полиномами наилучшего приближения.

4.3 Определение пути пройденного по параметрически заданной кривой и обратной функции

Предварительно определим функциональные зависимости: $S(t)$, длины дуг - $S_{мах}$, пройденных по заданным сегментам, обратные функции $t(S)$, а также их аппроксимации полиномами наилучшего приближения. В соответствии с параметрическими уравнениями каждого сегмента в функции безразмерного параметра t вычисляется путь, пройденный по параметрически заданной кубической кривой в функции $S(t)$, воспроизводится с высокой точностью обратная эталонная функция $t=g_{эт}(S)$ с последующей аппроксимацией ее полиномом наилучшего приближения $g(S)$.

Путь, пройденный по параметрически заданной кривой на плоскости (рис. 4.1) в функции нормированного времени, определяется формулой:

$$S(t)=\int_0^1 \sqrt{((x'(t))^2+((y'(t))^2}\,dt \tag{4.4}$$

Поскольку для интеграла (4.4) табличное представление отсутствует, то его можно определить в системе MathCAD путем численного интегрирования. При этом можно получить как длину пути $S_{мах}$ на заданном сегменте, так и точные, отдельные значения и графики функций $S(t)$ и $t(S)$ Например, путь вдоль линейной кривой Безье из т. $P_3(10;0)$ в т. $C_1(12;-1.5)$ в соответствии с параметрическими уравнениями $x(t)=10+2t, y(t)=-1.5t$ будет равен

$$S(t)=\int_0^1 \sqrt{((10+2t)')^2+((-1,5t)')^2}\,dt=2,5\,.$$

Путь, пройденный по параметрически заданной кубической кривой на плоскости между точкой $P_0(0;0)$, и точкой $C_1(12;-1,5)$, (рис. 4.1) равен

$$S(t)=\int_0^1 \sqrt{((9t+t^3)')^2+((15,75t-22,5t^2+6,75t^3)')^2}\,dt=12,23\,.$$

Для каждого из типов сегментов траектории предварительно по параметрическим уравнениям движения $x(t)$, $y(t)$, $z(t)$ в функции параметра $t\in[0;1]$ определяется максимальное значение пути $S_{мах}$ на каждом сегменте. Это значение в последующем используется для определения уравнения скорости в функции реального времени $V= v(t_р)$. Из текущих, точных значений численного интегрирования выражения $S(t)=\int_0^1 \sqrt{((x^{'}(t))^2+((y^{'}(t))^2}$ для воспроизведения сразу «эталонного» полинома Ньютона для обратной функции $t=g_{эт}(S(t))$ с равномерным расположением узлов интерполяции, причем аргумент и функция меняются местами, выбирается примерно 5-8 дискретных значений функции t_i и соответствующие им значения аргумента S_i (узлов интерполяции полинома). С помощью полинома Ньютона 5-8 степени воспроизводится «эталонная» монотонно нарастающая обратная функция $t=g_{эт}(S)$ с приведенной относительной погрешностью порядка долей % с последующей аппроксимацией ее фактически полиномом наилучшего приближения Чебышева $g(S) \cong g_{ЭТ}(S)$ более низкой степени. В этом полиноме для исключения скачков пути для границ интервалов (сегментов траектории) задаются значения: (t_0=0, S_0=0); $t_{мах}$=1, S=$S_{мах}$). Вычисляются и оцениваются максимальные $g_{ЭТ}(S)$-$g(S)$ и приведенные значения погрешностей. Для последней значения не должны превышать (1-2)%.

Монотонно изменяющиеся прямые и обратные функциональные зависимости S(t), t(S) можно и аппроксимировать с высокой степенью точности полиномами наилучшего приближения 2-й или 3-й степеней в соответствии с их графиками (фиг. 3-6). Для параметрических уравнений кривой Безье:

$$x(t)=9t+t^3,$$
$$y(t)=15.75\cdot t-22.5\cdot t^2+6.75\cdot t^3.$$

В соответствии с графиком функций $S(t)$ получен полином наилучшего приближения 2-й степени. На рис. 4.3 кривая $S(t)$ построена в соответствии с (4.4), а кривая $g(t)=0,462+10,495\cdot t+1,05\cdot t^2$ - полином наилучшего приближения 2-й степени.

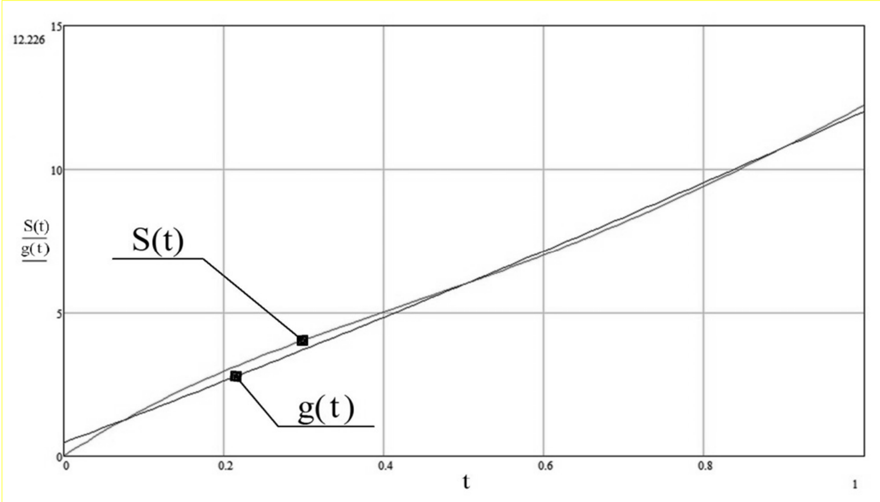


Рисунок 4.3 Аппроксимация полиномом 2-й степени пути *S*(*t*)

Коэффициенты полинома вычислены исходя из матричных уравнений

$$A = \begin{pmatrix} 0,075^0 & 0,075^1 & 0,075^2 \\ 0,5^0 & 0,5^1 & 0,5^2 \\ 0,895^0 & 0,895^1 & 0,895^2 \end{pmatrix}, \qquad B = \begin{pmatrix} 1,255 \\ 5,972 \\ 10,696 \end{pmatrix}$$

при оптимальных узлах аппроксимации полинома (значения безразмерного параметра *t* нормированного времени) 0,075, 0,5 и 0,895.

Аппроксимирующий полином 3-й степени: $g(t) = 0,077 + 16,298 \cdot t - 13,258 \cdot t^2 + 9,129 \cdot t^3$ получен в соответствии с матрицами

$$A = \begin{pmatrix} 0,05^0 & 0,05^1 & 0,05^2 & 0,05^3 \\ 0,32^0 & 0,32^1 & 0,32^2 & 0,32^3 \\ 0,67^0 & 0,67^1 & 0,67^2 & 0,67^3 \\ 0,99^0 & 0,99^1 & 0,99^2 & 0,99^3 \end{pmatrix}, \; B = \begin{pmatrix} 0,86 \\ 4,234 \\ 7,791 \\ 12,076 \end{pmatrix}.$$

Графики функций *S*(*t*) и *g*(*t*), построенные средствами программы MathCAD приведены на рис. 4.4.

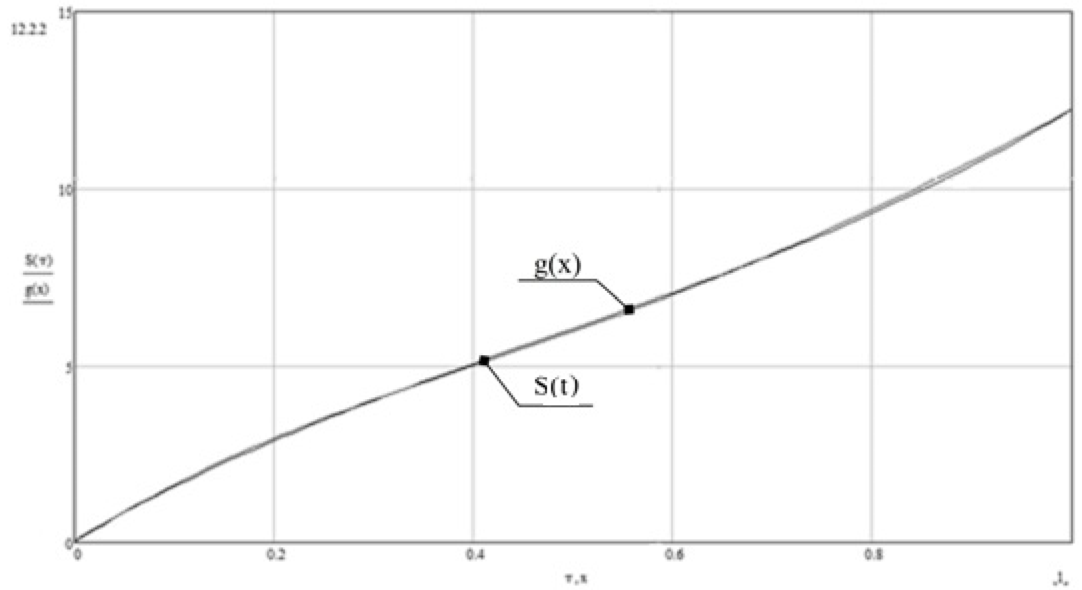


Рисунок 4.4. Аппроксимация полиномом 3-й степени пути $S(t)$

Построены графики погрешности аппроксимирующих полиномов 2-й и 3-й степеней для обратной функции (рис.4.5) $t_2(S)=0{,}086\,S-0{,}0003088\,S^2$ и $t_3(S)=0{,}047\cdot S+0{,}008807\cdot S^2-0{,}0004889S^3$ по отношению к «эталонному» полиному 5-й степени обратной функции.

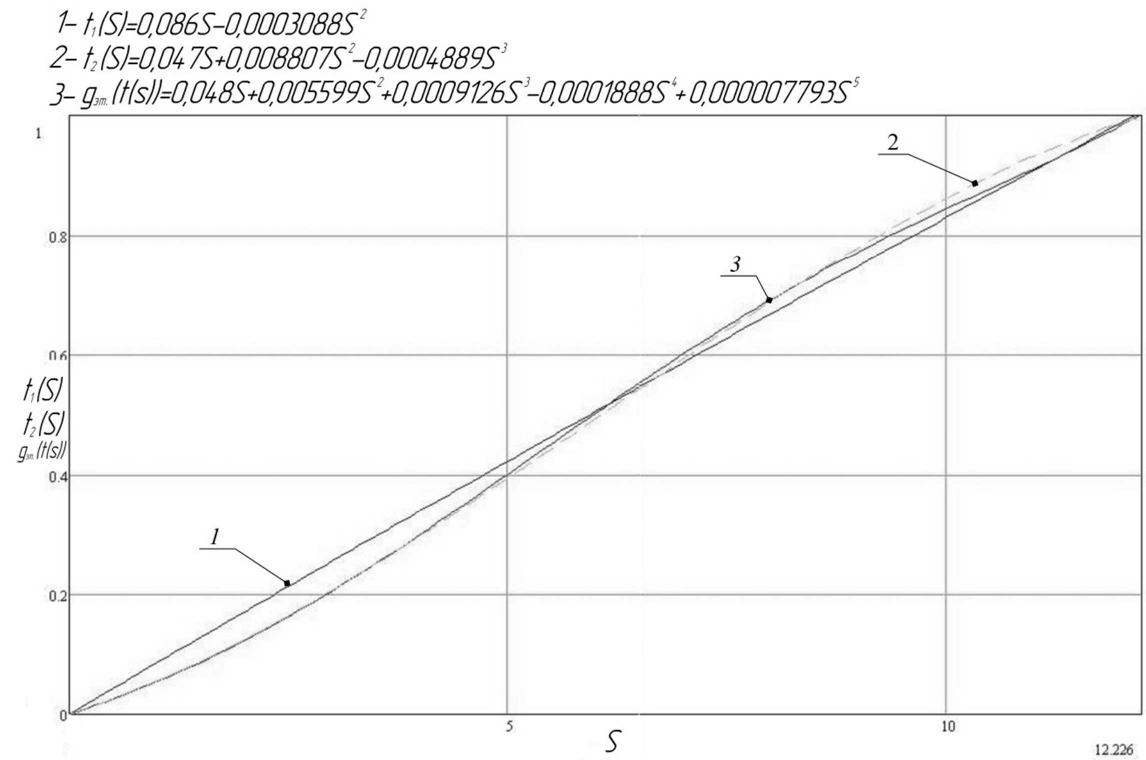


Рисунок 4.5 Аппроксимация полиномами 2-й, 3-й степени обратной функции $t(S)$, которая реализуется как «эталонная» полиномом Ньютона 5-й степени

На рис. 4.6 приведены графики погрешности аппроксимирующих полиномов 2-й и 3-й степеней для обратной функции $t(S)$ соответственно.

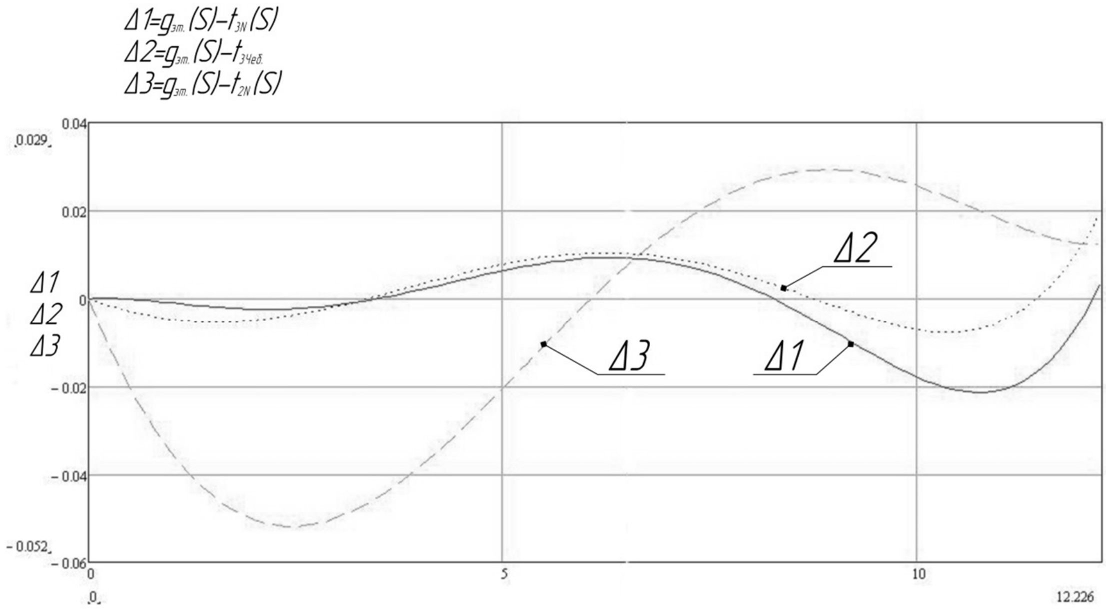


Рисунок 4.6. График погрешности полиномов Ньютона 2-й и 3-й степеней $t_2(S) = 0{,}086\,S - 0{,}0003088\,S^2$ и $t_3(S) = 0{,}047 \cdot S + 0{,}008807\,S^2 - 0{,}0004889\,S^3$ по отношению к «эталонному» полиному 5-й степени

Относительная приведенная погрешность аппроксимации текущего значения пути не превышает для полиномов второй и третьей степеней соответственно 0,05/12,22=0,004 и 0,02/12,22=0,001. Т.е. значение погрешности уменьшилось примерно в 4 раза. Определенные параметры, уравнения аппроксимации функциональных зависимостей являются исходными для определения кинематических параметров движения объекта, текущих значений координат траектории в функции реального времени.

4.4 Расчет параметров кинематики при равномерном и равноускоренном движении объекта. Комплексирование отдельных сегментов в траекторию функции реального времени

Движение с постоянной скоростью является наиболее простым и в тоже время наиболее распространенным способом движения. Для первого сегмента траектории с такими условиями можно принять начальное значение времени $T_{нач}$=0. При воспроизведении сегмента траектории в виде отрезка прямой с равномерным движением зависимость пути как от нормированного, так и от реального времени будет линейной. И при подстановке в параметрические уравнения воспроизведения траектории текущих нормированных значений времени (безразмерных значений параметра) при равномерном движении по отрезку прямой необходимо использовать выражение

$$t=t_p / t_{\text{мах}},$$

где реальное время прохождения сегмента $t_{\text{мах}}=S_{\text{мах}}/V$. При равномерном движении по кубической кривой для текущих значений t_p на заданном сегменте определяются линейно - нарастающие значения пути $S= Vt_p$. По текущим значениям S определяются в соответствии с аппроксимирующим полиномом обратную функцию $t(S)$ и нормированные значения времени t. После чего по параметрическим уравнениям вычисляются текущие значения прямоугольных координат [39].

При переменной скорости движения по сегментам расчет временных параметров будет более сложным. При равнозамедленном, равноускоренном или других видах движении для исключения скачков скорости и ускорения линейная скорость и ускорение в начале следующего сегмента движения должны быть равны скорости и ускорению в конце предшествующего сегмента. Текущее значение пути при равноускоренном движении по заданному отрезку прямой или кривой можно определить в соответствии с формулой

$$S= Vt_p+at_p^2/2.$$

После чего по аргументу - текущему пути S с помощью полинома, аппроксимирующего обратную функцию $t(S)$ как и ранее определяется нормированное время t. Например, для отрезка прямой в функции нормированного времени $t(S)$ определяются текущие значения координат

$$x(t) = 10 + 2t,$$
$$y(t) = -1,5t.$$

При более сложных заданиях функции скорости от времени значение пути на сегменте вычисляется с помощью интеграла $S = \int_0^t V(t_p)dt$.

Комплексирование, стыковка отдельных сегментов траектории может быть произведена последовательно, исходя из следующих положений. Форму сегмента, его длину - максимальное значение пути $S_{\text{мах}}$ по сегменту можно определять и варьировать при задании геометрической формы кривой. При назначенном значении начальной скорости величину ускорения, например, при равноускоренном движении можно изменять так, чтобы получить заданное время прохождения сегмента траектории в соответствии с квадратным уравнением

$$t_{1,2} = (-2V_H \pm \sqrt{4V_H^2 + 8aS_{\max}})\big/2a,$$

где V_H – начальная скорость; a – ускорение.

Например, при S=1000 м, $V_{нач.}$=10 м/с, a=2 м/с^2, получим $t_{1,2} = (-20 \pm \sqrt{400 + 8 \cdot 2000})\big/4$ м/с^2 $= 27{,}156$ с и т.д.

Итак, в соответствии с параметрическими уравнениями каждого сегмента в функции безразмерного параметра t вычисляется путь, пройденный по параметрически заданной кубической кривой в функции $S(t)$, воспроизводится обратная функция $t(S)$ с последующей аппроксимацией ее полиномом наилучшего приближения $g(t)$, по заданному закону воспроизведения скорости на заданном сегменте определяются текущее значение пройденного пути $S(t_{p1})$ в функции текущего временного интервала t_{p1} от начала сегмента и времени движения по сегменту, время прохождения сегмента, по общему, связывающему параметру текущего пути S на сегменте осуществляется переход от времени t_{p1} к параметру t с последующей подстановкой его в параметрические уравнения вычисления прямоугольных координат.

После проведения предварительных расчетов имеем отдельные сегменты, для начала и концов которых заданы абсолютные начальные и конечные значения прямоугольных координат, привязанные к точке стояния ИИС. Произведена привязка начала и конца формирования каждого сегмента траектории к реальному значению текущего временного интервала, формируемого таймером реального времени. Для каждого сегмента получены: время воспроизведения его начала и время завершения, уравнения зависимости длины пути в сегменте от начала сегмента в функции относительного реального времени от начала временного интервала формирования сегмента, аппроксимирующие полиномы для воспроизведения зависимостей нормированного времени $t(S)$ в каждом сегменте, параметрические уравнения воспроизведения координат в функции нормированного времени в каждом сегменте. Предварительные действия оператора, направленные на проектирование модели движения, подготовительные расчеты обеспечивают в общем временном интервале формирования траектории движения определить моменты времени начала, продолжительности и окончания движения объекта по каждому сегменту, которые позволяют идентифицировать именно тот сегмент, по которому перемещается объект в момент времени t_p воспроизведения каждого сегмента с присвоением ему порядкового номера. К каждому порядковому номеру сегмента привязывается таблица уравнений движения на заданном сегменте.

Все рассчитанные функциональные зависимости, уравнения движения заносятся в память, например рабочего места инструктора-оператора. После чего они в момент начала тренировки с воспроизведением траекторий движения на экране системы отображения информации передаются в блок расчета координат в соответствии с заданными геометрическими и кинематическими параметрами траектории. С начала тренировки, как и в прототипе по линейно - нарастающему временному интервалу, с таймера блок вычисления координат определяет время начала работы на очередном сегменте, вычисляет текущие прямоугольные координаты в пределах сегмента, фиксирует время перехода к следующему сегменту.

4.5 Действия оператора, направленные на проектирование модели движения

Подготовительные расчеты обеспечивают в общем временном интервале формирования траектории движения определить моменты времени начала, продолжительности и окончания движения объекта по каждому сегменту, которые позволяют идентифицировать именно тот сегмент, по которому перемещается объект в момент времени t_p воспроизведения каждого сегмента с присвоением ему порядкового номера. К каждому порядковому номеру сегмента привязывается его тип и таблица уравнений движения на заданном сегменте. Конечной целью при задании траектории движения, проводимого при этом математического моделирования является создание в некритичном масштабе времени наиболее простых вычислительных алгоритмов с достаточно высокими точностными характеристиками для последующей реализации их в критическом реальном масштабе времени воспроизведения траектории с минимальным числом вычислительных операций и обращений к памяти. При этом максимальный положительный эффект может быть получен, когда в параметрические уравнения воспроизведения траекторий в заданном сегменте в качестве аргумента в конечном счете будут подставляться непосредственно текущие значения функция $t \cong F(t_p)$. Погрешности воспроизведения функции $t \cong F(t_p)$ приведут только к колебаниям скорости движения по траектории. Предварительно последовательно составляется расчетная таблица.

Таблица 4.2 – Коэффициенты уравнений, параметров движения объекта по сегментам траектории.

№ сег-мента	Тип сегмента	$T_н$ с	$T_к$ с	Параметрические уравнения воспроизведения траекторий	Sмах	t=$g(S) \cong g_{эт}(S)$	$V_н$	$V_к$	$a_н$	$a_к$	V=$v(t_р)$	S=$s(t_р)$	$V_к$ $S(t_р)$	V=$f(t_р)$	h_0
0				$x(t)$, $y(t)$, $z(t)$											
N-1															

где $T_н$ и $T_к$ – моменты времени начала и окончания движения объекта по сегменту, которые позволяют идентифицировать именно тот сегмент, по которому перемещается объект в момент времени $t_р$.

Для каждого из типов сегментов траектории предварительно по параметрическим уравнениям движения $x(t)$, $y(t)$, $z(t)$ в функции параметра $t \in [0;1]$ определяется максимальное значение пути Sмах на каждом сегменте. Это значение в последующем используется для определения уравнения скорости в функции реального времени V= $v(t_р)$. Из текущих значений численного интегрирования выражения $S(t) = \int_0^1 \sqrt{((x'(t))^2 + ((y'(t))^2}$ для воспроизведения сразу «эталонного» полинома Ньютона для обратной функции t=$g_{эт}(S(t))$ с равномерным расположением узлов интерполяции, причем аргумент и функция меняются местами, выбирается примерно 5-8 дискретных значений функции t_i и соответствующие им значения аргумента S_i (узлов интерполяции полинома). С помощью полинома Ньютона 5-8 степени воспроизводится «эталонная» монотонно нарастающая обратная функция t=$g_{эт}(S)$ с приведенной относительной погрешностью порядка долей % с последующей аппроксимацией ее фактически полиномом наилучшего приближения Чебышева $g(S) \cong g_{ЭТ}(S)$ более низкой степени. В этом полиноме для исключения скачков пути для границ интервалов (сегментов траектории) задаются значения: (t_0=0, S_0=0); $t_{мах}$=1, S=$S_{мах}$).

Вычисляются и оцениваются максимальные $g_{ЭТ}(S)$-$g(S)$ и приведенные значения погрешностей. Для последней значения не должны превышать (1-2) %. Предварительно задается закон изменения линейной скорости в функции $t_р$ внутри каждого сегмента, позволяющий определить ее в заданный момент времени или пути. При равномерном движении ускорение равно нулю. При равноускоренном или равнозамедленном движении в сегменте задаются начальные и конечные значения ускорения. Фиксированные, контрольные значения скоростей также задаются в начальной и конечной точках сегмента. Каждое из уравнений применимо только для своего номера и (или) типа сегмента. Должен быть произведен поверочный расчет по плавному изменению скоростей при переходе с сегмента на сегмент с плавными без скачков изменениями скорости и ускорений. Определение проходимого пути S=$s(t_р)$ в каждом сегменте производится в соответствии с заданным законом изменения скорости V= $v(t_р)$. Проходимый путь в реальном масштабе времени при более сложном законе изменения скорости определяется выражением $S = \int_0^t V(t_p)dt$ и должен быть равен пути $S_{мах}$, проходимому по всему сегменту кривой Безье. Возможен итерационный подбор уравнения скорости в сегменте V= $v(t_р)$, определяющего фактическое время окончания прохождения сегмента $T_к$, чтобы получить заданные временные интервалы совпадения прохождения начал и концов сегмента. После чего по общему, связывающему параметру текущего пути S на сегменте с помощью функций t=$g(S) \cong g_{эт}(S)$, V= $v(t_р)$ и S=$s(t_р)$ осуществляется переход от времени $t_{р1}$ к параметру t с последующей подстановкой его в параметрические уравнения вычисления прямоугольных координат. Такая операция проводится для каждого сегмента. В результате формируется окончательная расчетная таблица 4.3, которая передается в память для последующего воспроизведения траектории каждого воздушного объекта.

Таблица 4.3 Расчетные данные для воспроизведения траектории

№ сегмента	Тип сегмента	$T_н$, с	$T_к$, с	Параметрические уравнения траекторий в заданном сегменте	$t=g(S)$	$S=s(t_р)$
0				$x(t)$, $y(t)$, $z(t)$		
N-1						

С начала тренировки по линейно - нарастающему временному интервалу с таймера блок вычисления координат определяет время начала работы на очередном сегменте, вычисляет текущие прямоугольные координаты в пределах сегмента, фиксирует время перехода к следующему сегменту. Применяя расчетные формулы движения по преодолеваемому в момент времени $t_р$ сегменту траектории (для которого выполняется условие $T_н \le t < T_k$), с использованием уравнений таблицы 4.3 можно получить текущие значения координат с заданной скоростью движения воздушного объекта. Данные расчеты осуществляются в автоматическом режиме по периодическим запросам от таймера. Отсчет времени в общем случае идет от начала воздушного налета.

4.6 Расчет величины пройденного пути при движении согласно квадратичному профилю скорости

Для функции, определенной между узлами интерполяции $\left(S_k, \vartheta_k\right)$ и $\left(S_{k+1}, \vartheta_{k+1}\right)$ и представляющей собой некоторый k-й участок сплайна профиля скорости, запишем функцию $\sigma\left(s\right)=\dfrac{1}{\upsilon\left(s\right)}=\dfrac{1}{a_k s^2+b_k s+c_k}$. Поскольку профиль скорости задан, ее коэффициенты a_k, b_k и c_k известны. Зависимость $\sigma\left(s\right)$ имеет важный геометрический смысл: ее интеграл по некоторому пути $l=S-S_k$ будет равен времени, затраченному объектом на преодоление этого пути:

$$\tau\left(l\right)=\int\limits_{S_k}^{S_k+l}\sigma\left(s\right)ds=\int\limits_{S_k}^{S_k+l}\frac{ds}{a_k s^2+b_k s+c_k} \qquad (4.5)$$

Однако, нас будет интересовать обратная зависимость – $l(\tau)$. Выбор второй степени сплайна и соответствующее выражение для $\tau(l)$ позволяют получить обратную функцию аналитически, не прибегая к численным методам. Интеграл (4.5) выражается следующим образом:

$$\int \frac{ds}{a_k s^2 + b_k s + c_k} = \begin{cases} \frac{1}{\sqrt{D}} \cdot \ln\left|\frac{2a_k s + b_k - \sqrt{D}}{2a_k s + b_k + \sqrt{D}}\right| + C, \text{при } D = b_k^2 - 4a_k c_k \ge 0 \\ \frac{2}{\sqrt{-D}} \cdot arctg\left(\frac{2a_k s + b_k}{\sqrt{-D}}\right) + C, \text{при } D = b_k^2 - 4a_k c_k < 0 \end{cases} \tag{4.6}$$

Конкретное аналитическое выражение зависит от дискриминанта квадратного уравнения. Рассмотрим каждый из двух вариантов отдельно, найдем для обоих случаев выражение $l(\tau)$, а также значение верхней границы интервала изменения параметра времени $\tau \in [0, \tau_k]$. Это значение соответствует времени, за которое объект преодолеет путь $l_k = S_{k+1} - S_k$.

Нахождение $l(\tau)$ для случая $D < 0$:

Интеграл (4.6), определенный при отрицательном дискриминанте для приращения пути $l = S - S_k$, будет иметь вид:

$$\tau(l) = \int_{S_k}^{S_k+l} \frac{ds}{a_k s^2 + b_k s + c_k} = \frac{2}{\sqrt{-D}} \cdot \left(arctg\left(\frac{2a_k(S_k + l) + b_k}{\sqrt{-D}}\right) - arctg\left(\frac{2a_k S_k + b_k}{\sqrt{-D}}\right) \right) \tag{4.7}$$

Взяв тангенс от обеих частей и, выполнив возможные преобразования с учетом того, что $a_k S_k^2 + b_k S_k + c_k = \upsilon_k$, получим искомую зависимость:

$$l(\tau) = \frac{2\upsilon_k}{\sqrt{-D} \cdot ctg\left(\frac{\tau\sqrt{-D}}{2}\right) - (2a_k S_k + b_k)}. \tag{4.8}$$

Выражение для τ_k:

$$\tau_k = \tau(l_k) = \frac{2}{\sqrt{-D}} arcctg\left(\frac{\frac{2\upsilon_k}{S_{k+1} - S_k} + 2a_k S_k + b_k}{\sqrt{-D}} \right) \tag{4.9}$$

Нахождение $l(\tau)$ для случая $D \geq 0$:

При $D \geq 0$ формулу зависимости, следует записать так:

$$\tau(l) = \frac{1}{\sqrt{D}} \cdot \left(\ln\left| \frac{2a(S_k + l) + b - \sqrt{D}}{2a(S_k + l) + b + \sqrt{D}} \right| - \ln\left| \frac{2aS_k + b - \sqrt{D}}{2aS_k + b + \sqrt{D}} \right| \right) = \\ = \frac{1}{\sqrt{D}} \cdot \ln\left| \frac{2aS_k + b - \sqrt{D} + 2al}{2aS_k + b + \sqrt{D} + 2al} \cdot \frac{2aS_k + b + \sqrt{D}}{2aS_k + b - \sqrt{D}} \right| \tag{4.10}$$

Эта функция не определена при любом из следующих условий: $2a(S_k + l) + b - \sqrt{D} = 0$, $2a(S_k + l) + b + \sqrt{D} = 0$, $2aS_k + b - \sqrt{D} = 0$, $2aS_k + b + \sqrt{D} = 0$, т.е. при значениях аргумента $s = \dfrac{-b \pm \sqrt{D}}{2a}$ в узловых точках сплайна и вне их. Таким образом, парабола не должна пересекать прямую $\upsilon = 0$, что означает непрерывное и невозвратное движение по предписанной траектории от начальной к конечной точке.

Принимая $P = 2aS_k + b - \sqrt{D} = \text{const}$, $Q = 2aS_k + b + \sqrt{D} = \text{const}$, и, соответственно, $P + 2al = 2a(S_k + l) + b - \sqrt{D}$, $Q + 2al = 2a(S_k + l) + b + \sqrt{D}$, а также используя обе части равенства в качестве показателя степени числа e (основания натурального логарифма), получаем:

$$e^{\tau\sqrt{D}} = \left| \frac{Q(P + 2al)}{P(Q + 2al)} \right| \Rightarrow e^{\tau\sqrt{D}} = \begin{cases} \dfrac{Q(P + 2al)}{P(Q + 2al)}, \text{при } \dfrac{Q(P + 2al)}{P(Q + 2al)} > 0 \\ -\dfrac{Q(P + 2al)}{P(Q + 2al)}, \text{при } \dfrac{Q(P + 2al)}{P(Q + 2al)} < 0 \end{cases} \tag{4.11}$$

Для корректного выражения $l(\tau)$ требуется выяснить, в каких случаях модуль раскрывается с положительным, а в каких – с отрицательным знаком. Это зависит от знаков отдельных множителей, входящих в состав выражения в модульных скобках.

Рассмотрим график функции, представляющей собой часть сплайна скорости. Он существует только в правом верхнем квадранте, поскольку и аргумент-путь и функция-скорость должны быть положительными величинами (путь может также принимать значение 0). В рассматриваемом варианте раскрытия интеграла дискриминант многочлена положительный, т.е. парабола $\upsilon(S)$ пересекает ось абсцисс в двух точках. Этим условиям соответствуют два варианта расположения параболы:

- с вершиной ниже оси абсцисс и направленными вверх ветвями $a > 0 \wedge c < 0$

- с вершиной выше оси абсцисс и направленными вниз ветвями $a < 0 \wedge c > 0$.

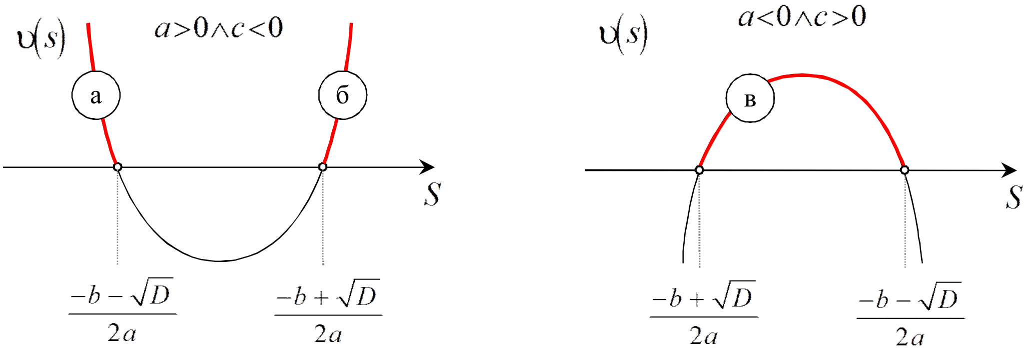


Рисунок 4.7 Расположение параболы и ее фрагментов, составляющих сплайн скорости

Отмеченные на рис. 4.7 участки параболы (а, б и в) могут служить в качестве фрагментов сплайна скорости. Остальные – нет. Это означает, что и узловые точки пути S_k, и сам изменяющийся аргумент интерполянта S должны одновременно принадлежать одному из этих участков. Эти условия можно записать следующим образом:

$$: \begin{cases} a > 0 \\ s, S_k < \dfrac{\sqrt{D} - b}{2a} \\ s, S_k < \dfrac{-\sqrt{D} - b}{2a} \end{cases} \qquad : \begin{cases} a \geq 0 \\ s, S_k > \dfrac{\sqrt{D} - b}{2a} \\ s, S_k > \dfrac{-\sqrt{D} - b}{2a} \end{cases} \qquad : \begin{cases} a < 0 \\ s, S_k > \dfrac{\sqrt{D} - b}{2a} \\ s, S_k < \dfrac{-\sqrt{D} - b}{2a} \end{cases}$$

а затем преобразовать к виду:

$$:\quad \begin{cases} a>0 \\ 2aS_k-\sqrt{D}+b<0 \\ 2aS_k+\sqrt{D}+b<0 \\ 2aS-\sqrt{D}+b<0 \\ 2aS+\sqrt{D}+b<0 \end{cases} \quad : \quad \begin{cases} a>0 \\ 2aS_k-\sqrt{D}+b>0 \\ 2aS_k+\sqrt{D}+b>0 \\ 2aS-\sqrt{D}+b>0 \\ 2aS+\sqrt{D}+b>0 \end{cases} \quad : \quad \begin{cases} a>0 \\ 2aS_k-\sqrt{D}+b>0 \\ 2aS_k+\sqrt{D}+b<0 \\ 2aS-\sqrt{D}+b>0 \\ 2aS+\sqrt{D}+b<0 \end{cases}$$

Учитывая эти условия, делаем вывод, что во всех возможных вариантах конфигурации участков парабол, составляющих сплайн скорости, выражение в модульных скобках всегда будет положительным, а значит, модуль раскрывается единственным способом. Отсюда:

$$l(\tau)=\frac{1-e^{\tau\sqrt{D}}}{2a\left(\frac{e^{\tau\sqrt{D}}}{Q}-\frac{1}{P}\right)}=\frac{1-e^{\tau\sqrt{D}}}{2a\left(\frac{e^{\tau\sqrt{D}}}{2aS_k+b+\sqrt{D}}-\frac{1}{2aS_k+b-\sqrt{D}}\right)} \tag{4.12}$$

$$\tau_k=\frac{1}{\sqrt{D}}\cdot\ln\left|\frac{\left(2aS_{k+1}+b-\sqrt{D}\right)\left(2aS_k+b+\sqrt{D}\right)}{\left(2aS_{k+1}+b+\sqrt{D}\right)\left(2aS_k+b-\sqrt{D}\right)}\right| \tag{4.13}$$

Рассмотрим пример расчета величины пройденного пути при движении согласно линейному профилю скорости. Если некоторый *k*-й фрагмент сплайна скорости является полиномом первой степени $\upsilon(s)=a_k s+b_k$, то процесс нахождения $l(\tau)$ для него выглядит следующим образом:

$$\tau(s)=\int\frac{ds}{a_k s+b_k} \tag{4.14}$$

$$\tau(l)=\frac{1}{a_k}\ln\left|a_k\left(S_k+l\right)+b_k\right|-\frac{1}{a_k}\ln\left|a_kS_k+b_k\right|=\frac{1}{a_k}\ln\left|\frac{a_k\left(S_k+l\right)+b_k}{a_kS_k+b_k}\right|. \tag{4.15}$$

В выражении (4.15) числитель и знаменатель в модульных скобках – это значения скорости объекта при текущем пути S_k+l и в начальной точке пути S_k соответственно. Поскольку эти величины заведомо положительные, модуль раскрывается с положительным знаком, и закон изменения пути будет происходить из (4.15) в следующем виде:

$$l(\tau) = \left(S_k + \frac{b}{a} \right) \left(e^{a_k \tau} - 1 \right), \qquad (4.16)$$

$$\tau_k = \tau(l_k) = \frac{1}{a_k} \ln \left(\frac{a_k S_{k+1} + b_k}{a_k S_k + b_k} \right). \qquad (4.17)$$

В соответствии с выражениями 4.5-4.17 была построена кривая, представленная на рис. 4.8.

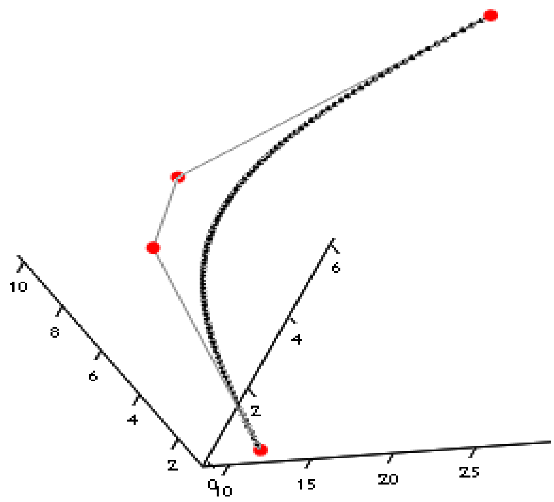


Рисунок 4.8 Смоделированная в MathCAD траектория на основе кривой Безье

4.7 Разработка программы имитации воздушной обстановки

В соответствии с вышеизложенным наиболее оптимальной формой организации программы имитации траекторий движения объекта является модульная структура (рис. 4.9).

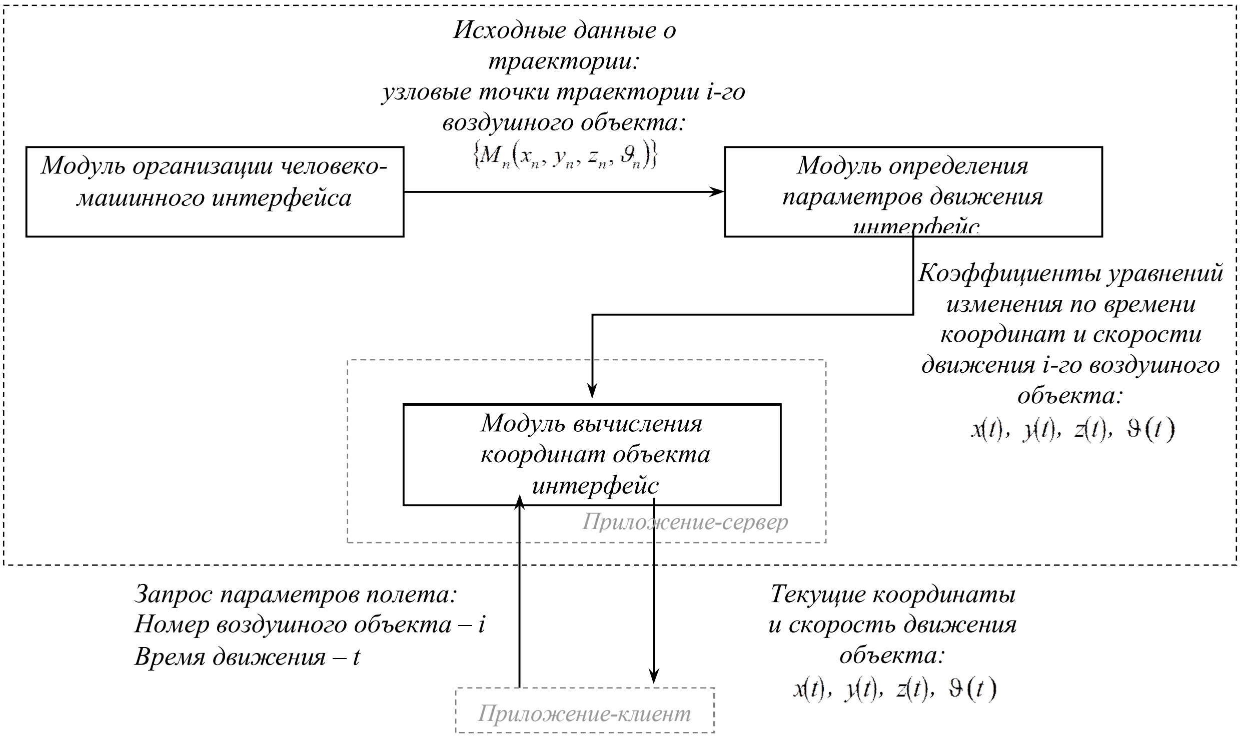


Рисунок 4.9. Общая структура программы, ее взаимодействие с приложением-клиентом

Программа должна состоять из двух относительно независимых частей: с одной стороны – приложение, содержащее человеко-машинный интерфейс и алгоритмы для задания траектории (Air Situation Designer), с другой – программа, представляющая собой сервер или службу, обрабатывающую поступающие запросы на положение цели (Air Situation Server).

Ввод траекторий движения воздушных объектов осуществляется на предварительном этапе работы приложения-сервера. В дальнейшем исключается какое-либо взаимодействие двух структурных единиц программы. Поэтому реализовать обмен данными можно посредством файла определения воздушной обстановки: как только сценарий движения объектов определен, и коэффициенты уравнений движения рассчитаны, они сохраняются в файл в виде массива однотипных записей – по одной для каждого объекта. В момент запуска приложения-сервера, оно загружает подготовленных файл воздушной обстановки в память в виде, наиболее удобном для последующего расчета координат и скоростей объектов.

Сервер имеет максимально простую реализацию. Он включает в себя функцию, которая в качестве входных параметров принимает время t, идентификатор или порядковый номер объекта i, а также массив констант, загруженных из файла определения воздушной обстановки. Вызов функции и передача параметров происходит в момент подачи клиентом на порт, занимаемый сервером, сообщения «GetState (i, t)». После того, как запрашиваемые параметры движения определены, сервер посылает клиенту ответ в виде строки типа «State i: x, y, z, v, α, β», в котором передает координаты и скорость объекта i, а также азимут (α) и угол места (β) направления его движения. Если в запрашиваемый момент времени объект не появился в зоне обзора или уже вышел из зоны обзора, сообщение от сервера принимает вид: «State i: not presented».

Сервер представлен единственным исполняемым файлом, при запуске которого из-под ОС Windows на экране появляется окно, запрашивающее два входных параметра: файл воздушной обстановки и занимаемый порт. После указания файла и запуска сервера, приложение может быть свернуто в системный трей (рис.4.10).

Рисунок 4.10 Внешний вид приложения-сервера

Основная функциональная нагрузка ложится на программу задания воздушной обстановки. Она обладает значительно более сложным интерфейсом (рис. 4.11).

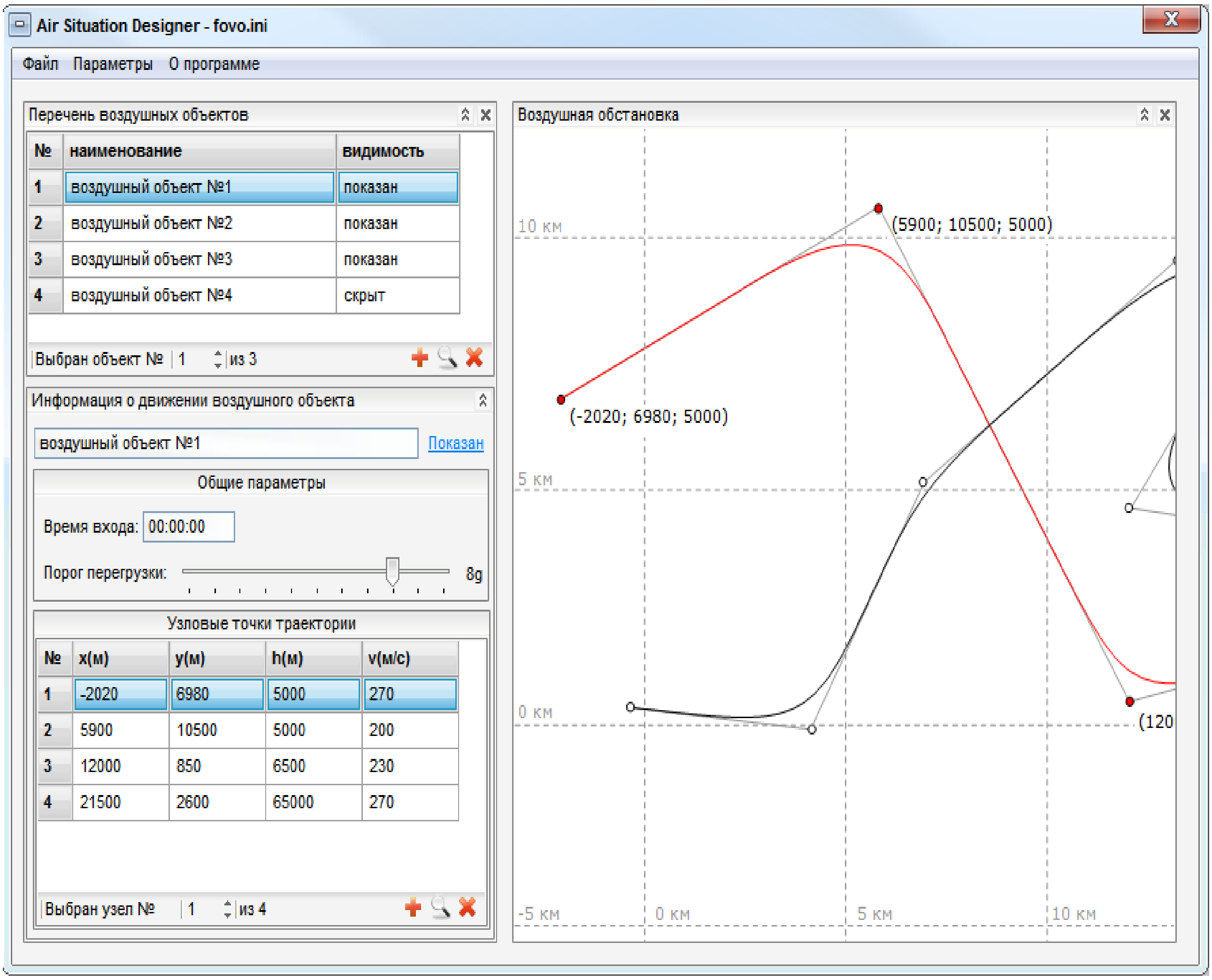


Рисунок 4.11 Внешний вид приложения-дизайнера

Управление процессом создания воздушной обстановки осуществляется посредством меню приложения (рис. 4.12), состоящего из двух основных частей: списка добавленных воздушных объектов (вверху) и параметров их перемещения (внизу).

Выбор строки в таблице воздушных объектов автоматически выделяет соответствующую ему траекторию в окне воздушной обстановки. То же самое можно делать из панели статуса данной группы элементов управления.

В соответствии с выбранным воздушным объектом меняется содержимое группы, отображающей информацию о движении данного воздушного объекта. Здесь может быть задано его наименование, видимость, время входа в зону обзора, а также узловые точки траектории движения.

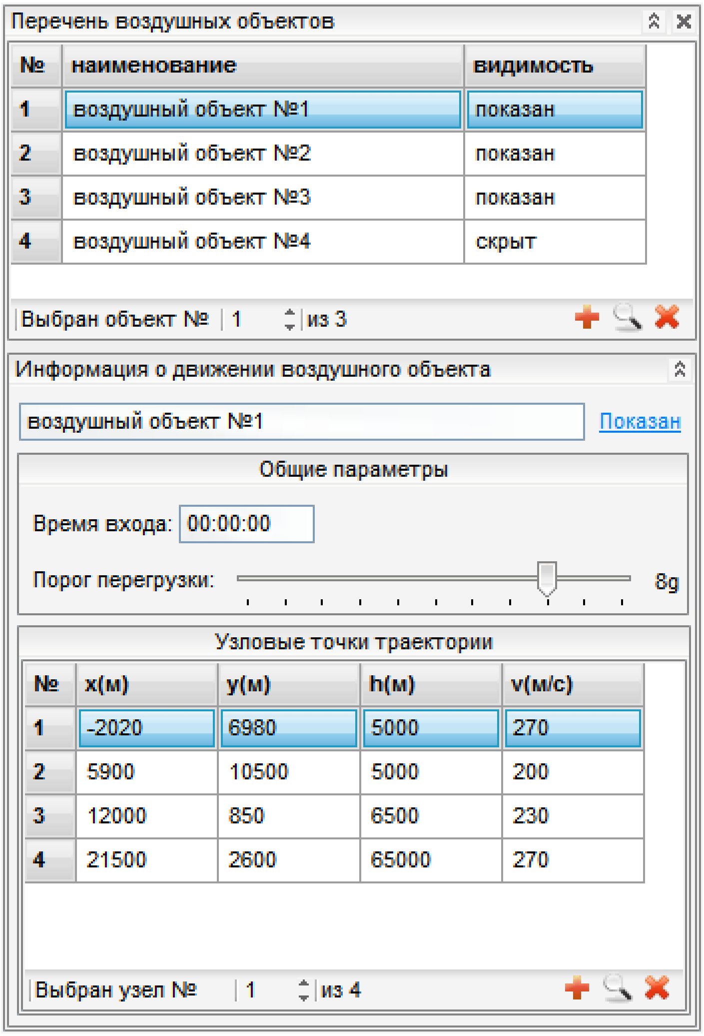


Рисунок 4.12 Меню приложения-дизайнера

Для наиболее удобного и наглядного формирования трехмерных трасс полета, составляющих воздушную обстановку, все они графически представлены в главном окне приложения. Каждая траектория с набором узловых точек, опорной ломаной линией, трассой полета представляет собой экземпляр класса TAirTrace. Этот класс содержит поля и методы, позволяющие задавать внешний вид и свойства траектории, а также отрисовывать траекторию на предложенной канве. В области визуализации воздушной обстановки класс TAirTrace создает дополнительный слой – объект типа TBitmap, который может быть включен или выключен из потока визуализации.

Реализуя принцип WYSIWYG, траектории полета интерактивны: узловые точки опорных ломаных линий, определяющих характер движения, доступны для перемещения мышью – таким образом можно легко задавать координаты движения объекта в плоскости XY. Для более точного их определения существует возможность ручного ввода значений координат в инспекторе объектов приложения. Вручную вводятся также высота *h* каждого узла над поверхностью Земли и его скорость (из шкалы допустимых значений). Всякий раз, когда оператор завершает ввод координат узловой точки (отпускает мышь при перетаскивании точки или закрывает модальное окно ввода параметров узла (рис. 4.13), программа сглаживает опорную линию.

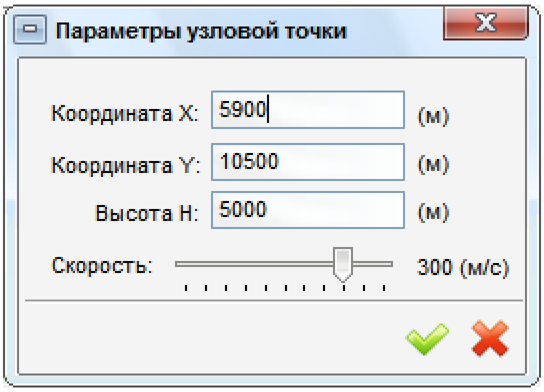


Рисунок 4.13 Окно ввода параметров узловой точки

При возникновении коллизии построение траектории прерывается на соответствующем сегменте – таким образом, оператор видит необходимость внесения изменений в расположение узлов опорной ломаной линии.

Если на этапе расчета обнаруживается, что перегрузка объекта превысила заданный допустимый порог, оператору выводится соответствующее предупреждающее сообщение. На момент сохранения воздушной обстановки в файл такого рода замечания должны быть устранены изменением скоростного режима или геометрии линии – в противном случае возможность сохранения пропадает. То же самое касается и коллизий с переходными кривыми.

Сохраненный файл воздушной обстановки в последствии может быть открыт для внесения изменений.

Выводы к главе 4

1. Разработан метод воспроизведения траекторий движения воздушных объектов из плавно сопрягаемых сегментов в системе координат зоны обзора

ИИС и преобразованием безразмерного аргумента параметрических уравнений кривых Безье при различных законах изменений линейной скорости в аргумент реального линейно–нарастающего интервала времени воспроизведения траектории.

2. При комплексировании по полной траектории движения с помощью сегментов, выполненных в виде кривых Безье 3-го и 1-го порядков, обеспечен плавный переход для широко распространенного сопряжения от кубической кривой Безье к отрезку прямой и обратно с выполнением условия равенства первой и второй производных.

3. Для уменьшения вычислительных затрат проведена аппроксимация обратной зависимости времени от пути движения при помощи полинома наилучшего приближения второго и третьего порядков. Относительная приведенная погрешность аппроксимации текущего значения пути не превышает для полиномов второй и третьей степеней соответственно 0,365/12,223=0,0298 и 0,108/12,223=0,0088. Т.е. значение при увеличении степени аппроксимирующего полинома погрешности уменьшилось в 0,0298/0,0088=3,39 раз.

4. В среде Delphi было создано два независимых приложения: Air Situation Designer – для визуального монтажа воздушной обстановки и сохранения результатов работы в файл, а также Air Situation Server – для обслуживания асинхронных запросов о состоянии движения воздушных объектов, поступающих от сторонних приложений. Также было создано приложение, иллюстрирующее движение воздушных объектов согласно предоставляемым сервером данным.

Заключение

1. Разработаны методы и алгоритмы поиска полиномов наилучшего приближения различных степеней для аппроксимации функциональных зависимостей, повышающих точность представления типовых функций и минимизацию программно-аппаратурных затрат.

2. Созданы алгоритмы аппроксимации стандартных функций, входящие в основные алгоритмы обработки информации РТС, арифметических и векторных операций с диапазоном представления от 3 до 64 двоичных разрядов. Устранена избыточная точность путем обеспечения максимального дискретного приращения 2…20 и более двоичных цифр результата при возрастании сложности алгоритма не более чем на 1…6 операций. Обеспечено уменьшение погрешностей результата путём взаимной компенсации составляющих погрешностей.

3. Для гибридных алгоритмов преобразования координат, ортогональных составляющих сигналов в амплитуду и фазу увеличено быстродействие вычислителя в 2 раза. Рациональное использование предлагаемых алгоритмов в технических приложениях позволяет обеспечить формирование от 1 до 32 и более значащих двоичных разрядов операндов, избегая невостребованной избыточной точности результата.

4. Разработан метод воспроизведения траекторий воздушных объектов из плавно сопрягаемых сегментов на основе параметрических уравнений кривых Безье с контролем перегрузок.

Литература

1. Самарин, О.Ф. Радиолокационные системы многофункциональных самолетов / О.Ф. Самарин, А.А. Соловьев, Т.В. Шарова. под ред. А. И. Канащенкова и В.И. Меркулова. - М.: Радиотехника - 2007. - 280с.

2. Чекушкин, В.В. Оптимизация методов реализации вычислительных процессов в устройствах радиолокационных станций / В.В. Чекушкин // Радиотехника №6. - 2007.– С. 72-74.

3. Ширман, Я.Д. Радиоэлектронные системы: Основы построения и теория / Под ред. Я.Д. Ширмана Справочник. Изд. 2-е, перераб. и доп. – М.: Радиотехника. - 2007. – 512с.

4. Бахвалов, Н.С. Численные методы / Н.С. Бахвалов, Н.П. Жидков, Г.М. Кобельков – М.: Лаборатория Базовых Знаний. - 2000. – 624с.

5. Ларионов, В.А. Концепция калибровки интеллектуальных датчиков / В.А. Ларионов // Приборы и системы. Управление, контроль, диагностика, № 2012. - №12 – С. 46-51.

6. Чекушкин, В.В. Метод поиска полиномов наилучшего приближения для воспроизведения функциональных зависимостей калибровки датчиков и измерительных систем / В.В. Чекушкин, А.М. Аверьянов // Датчики и системы. – 2009. - №3 – С. 2-6.

7. Ремез, Е.Я. Основы численных методов чебышевского приближения / Е.Я. Ремез – Киев.: Наукова Думка. - 1969. – 625с.

8. Чекушкин, В.В. Повышение точности измерительных систем с нестабильными параметрами / В.В. Чекушкин, В.В. Булкин // Измерительная техника. – 2006. - №1 – С.7-11.

9. Чекушкин, В.В. Коррекция погрешностей в измерительных приборах / В.В. Чекушкин, Л.Г. Алексеева // Приборы и системы. Управление, контроль, диагностика. – 2008. - № 5 - С.38-43.

10. Сапельников, В.М. Функциональный цифроаналоговый преобразователь в широкополосном цифроуправляемом калибраторе фазы / В.М. Сапельников, Р.А. Хакимов // Приборы и техника эксперимента. – 2005. - № 4 – С. 43-46.

11. Красников, А.К. Модели и методы оценки точности и устойчивости алгоритмов / А.К. Красников, В.А. Красникова, С.В. Матис // Приборы и системы. Управление, контроль, диагностика. - 2012. - №12 - С. 22-30.

12. Гвоздева, Т.В. Проектирование информационных систем: учеб. пособие / Т.В. Гвоздева, Б.А. Баллод. - Ростов Н/Д: Феникс. - 2009. - 508с.

13. Сосулин, Ю.Г. Теоретические основы радиолокации и радионавигации / Ю.Г. Сосулин – М.: Радио и связь - 1992. – 304с.

14. Квасов, Б.И. Методы изогеометрической аппроксимации сплайнами / Б.И. Квасов – М.: ФИЗМАЛИТ - 2006.-360с.

15. Каханер, Д. Численные методы и программное обеспечение: пер. с англ. / Д. Каханер, К. Моулер, С. Нэш – М.: Мир - 2001.– 575с.

16. Чекушкин, В.В. Совершенствование полиномиальных методов воспроизведения функциональных зависимостей в информационно-измерительных системах / В.В. Чекушкин, И.В. Пантелеев, К.В. Михеев // Измерительная техника. - 2015. –№4.- С. 16-21.

17. Chekushkin, V.V. Improving polynomial methods of reconstruction of functional dependences in information-measuring systems / V.V. Chekushkin, I.V. Panteleev, K.V. Mikheev // Measurement Techniques. - 2015, Volume 58, Issue 4, PP 385-392. ISSN 0543-1972.

18. Чекушкин, В.В. Быстродействующие алгоритмы преобразования ортогональных составляющих сигналов в амплитуду и фазу / В.В. Чекушкин, К.В. Михеев, И.В. Пантелеев // Цифровая обработка сигналов. – 2015. - №1. - С.32-35.

19. Аверьянов, А.М. Имитация траекторий движения воздушных объектов для радиолокационных систем управления и контроля воздушного пространства / А.М. Аверьянов, М.С. Бобров, В.В. Чекушкин // Мехатроника, автоматизация, управление – М.: «Новые технологии» - 2009. - №9 - С. 70–80.

20. Бобров, М.С. Реализация трасс движения воздушных объектов в тренажерно-моделирующих системах / М.С. Бобров, А.М. Аверьянов, Г.Г. Пискунов, В.В. Чекушкин // Вопросы радиоэлектроники. - Серия ЭВТ. – 2009. - №4 - С. 157–166.

21. Аверьянов, А.М. Оценка ускорения при аппроксимации параболических сегментов траектории движения объекта радиолокационного обнаружения / А.М. Аверьянов, М.С. Бобров // Вопросы радиоэлектроники. Серия РЛТ. – 2011 – вып. 1 - С. 210-217.

22. Чекушкин, В.В. Описание движения вдоль кусочно-заданной кубической кривой Безье через оценку параметра кривой по пройденному пути / В.В. Чекушкин, А.М. Аверьянов // Методы и устройства передачи и обработки информации. – 2011. - №13 - С.99-104.

23. Аверьянов, А.М. Математические модели формирования тестовых сигналов в радиотехнических устройствах имитации воздушной обстановки: автореф. дис. Канд. тех. наук: 05.12.04 / Аверьянов Александр Михайлович – Влад-р.,2011. – 18 с.

24. Чекушкин В.В. Вычислительные процессы в информационно-измерительных системах: учеб. пособие. / В.В. Чекушкин, В.В. Булкин // Изд. - полиграфический центр МИВлГУ. - 2009. - 120с.

25. Байков, В.Д. Аппаратурная реализация элементарных функций в ЦВМ / В.Д. Байков, В.Б. Смолов / Л. Изд-во Ленингр. ун-та. - 1975. – 96с.

26. Чекушкин, В.В. Совершенствование полиномиальных методов воспроизведения тригонометрических функций в информационно-вычислительных системах / В.В. Чекушкин, И.В. Пантелеев // Радиотехнические и телекоммуникационные системы. 2013. - № 1. - С.53-59.

27. Чекушкин, В.В. Быстродействующие алгоритмы поиска полиномов наилучшего приближения для воспроизведения функциональных зависимостей в информационно-измерительных системах / В.В. Чекушкин, К.В. Михеев // Измерительная техника. - 2016. – №4.- С. 7-10.

28. Чекушкин, В.В. Быстродействующие методы воспроизведения функциональных зависимостей в радиоэлектронных системах / В.В. Чекушкин // Радиотехнические и телекоммуникационные системы – 2014. - № 1. - С. 87-99.

29. Люстерник, Л.А. Справочная библиотека / Под общ. редакцией Л.А. Люстерника и А.П. Янпольского / Математический анализ. Вычисление элементарных функций. М.: Физматгиз. - 1963. - 248с.

30. Чекушкин, В.В. Совершенствование методов поиска полиномов наилучшего приближения для воспроизведения функциональных зависимостей в измерительных системах / В.В. Чекушкин, И.В. Пантелеев, И.Р. Сарибжанов // Радиотехнические и телекоммуникационные системы. – 2012. – №2. – С.49-52.

31. Caro, D. Direct digital frequency synthesizer using nonuniform piecewiselinear approximation / D. Caro, N. Petra, A. Strollo // IEE Trans. Circuit Syst .1. Reg. Papers. – 2011. - vol. 58. - p. 2409-2419.

32. Чекушкин, В.В. Программа поиска полиномов наилучшего приближения для воспроизведения функциональных зависимостей с взаимной компенсацией составляющих погрешностей результата / В.В. Чекушкин, К.В. Михеев // Свидетельство об официальной регистрации программы для ЭВМ №2015610539 от 13.01.2015.

33. Григорьев, Б.М. Радиолокационное обеспечение полетов военной авиации. Системотехника и приложения. Под ред. В.Ю. Кузменкова / Б.М. Григорьев, А.В. Губанов, А.А. Крылов, В.Ю. Кузьменков, Ю.Н. Петухов, Ю.В. Спирин, Д.Л. Тупарев – Черноголовка. Ред. – изд. Отдел. ИПХФ РАИ. – 2015. – 304с.

34. Чекушкин, В.В. Реализация вычислительных процессов в системах управления и контроля: учеб. пособие / В.В. Чекушкин.– Муром. - 2001. – 44с.

35. Schulte A, Michall J. A family of variable – precision interval arithmetic processors / A Schulte, J. Michall, Earl E Swarz Lander (Jr) // IEEE Trans Comput 2000/ 49.– № 5.– С. 387-397.

36. Чекушкин, В.В. Совершенствование методов преобразования ортогональных составляющих сигналов в амплитуду и фазу / В.В. Чекушкин, О.В. Юрин // Метрология. - 2001. - №11 - С.9-17.

37. Аверьянов, А.М. Параметрическое задание кинематики движения воздушного объекта на участке маневрирования / А.М. Аверьянов, М.С. Бобров, В.В. Чекушкин // Мехатроника, автоматизация, управление – М.: «Новые технологии» - 2010. - №5 - С. 67–73

38. Чекушкин, В.В. Способ имитации траекторий движения воздушных объектов / В.В. Чекушкин, А.М. Аверьянов, М.С. Бобров // Патент № 2419072 от 20.05.2011 – Бюл. №14.

39. Чекушкин, В.В. Метод воспроизведения траекторий движения объектов. / В.В. Чекушкин, К.В. Михеев // Наука и образование в развитии промышленной, социальной и экономической сфер регионов России: VIII Всероссийские научные Зворыкинские чтения: сб. тез. докл. Всероссийской межвузовской научной конференции. Муром, 5 февр. 2016. – С.172-173.

40. Rasch, P. Comparison of Shape Preserving Interpolators-National center for atmospheric research USA / P. Rasch, D.A. Williamson Boulder. – Colorado. – 1989. - 53 p.

41. Шауман А.М. Основы машинной арифметики / А.М. Шауман – Л.: Изд-во Ленингр. ун-та. - 1979. – 312с.

42. Чекушкин, В.В. Способ калибровки измерительной системы / В.В. Чекушкин, И.В. Пантелеев, А.Д. Богатов // Патент на изобретение №2476896 от 27.02.2013. - Бюл. № 6.

43. Сапельников, В.М. Методы построения цифроуправляемых калибраторов фазы в приборостроении / В.М. Сапельников, А.Д. Максутов, С.Е. Клименко // Измерительная техника. - 2012. - №3 – с. 53-57.

44. Huynh, H. Accurate monotone cubic interpolation / H. Huynh - SIAM J. Numer. Anal. - Vol. 30 – 1993 - №3. - p. 57-100.

45. Чекушкин, В.В. Быстродействующие методы воспроизведения функциональных зависимостей в радиоэлектронных системах / В.В. Чекушкин // Радиотехнические и телекоммуникационные системы - 2014. - №1 - с. 87-99.

46. Аверьянов, А.М. Методы повышения быстродействия и точностных характеристик преобразователей ортогональных составляющих сигнала в амплитуду / А.М. Аверьянов, И.В. Пантелеев, В.В. Чекушкин // Измерительная техника - 2012. - №8. - С.9-14.

47. Чекушкин, В.В. Реализация вычислительных процессов в информационно-измерительных системах: монография / В.В. Чекушкин, О.В. Юрин, В.В. Булкин – Муром. Изд.- полиграфический центр МИ ВлГУ - 2005. – 158с.

48. Михеев, К.В. Использование отладочных плат для ПЛИС при проектировании новых устройств. / К.В. Михеев, А.Б. Оранский // Всероссийская молодежная научная конференция «Научный потенциал молодежи – будущее России». Сб. тез. докладов VI Всероссийской научной конференции (Муром, 12 апреля 2014 г.). - Муром: Изд.-полиграфический центр МИ ВлГУ. - 2014. - С. 789.

49. Кнут, Д.Э. Искусство программирования, том 2. Получисленные алгоритмы, 3-е изд.: Пер. с англ. / Д.Э. Кнут - Издательский дом «Вильямс». - 2000. - 832с.

50. Амосов, А.А. Вычислительные методы для инженеров / А.А. Амосов, Ю.А. Дубинский, Н.В. Копчёнова – М.: Высш. шк. - 1994. – 544с.

51. Чекушкин, В.В. Программа поиска полиномов наилучшего приближения для воспроизведения функциональных зависимостей / В.В. Чекушкин, К.В. Михеев, И.В. Пантелеев // Свидетельство об официальной регистрации программы для ЭВМ № 2014615085 от 16.05.2014.

52. Chen, Y. H.A direct digital frequency synthesizer based a new form of polynomial approximations / Y.N. Chen, Y.A. Chau // IEEE Trans. Consum. Electron - 2010. - V. 56. - N. 2. - p. 436–440.

53. Чекушкин, В.В. Реализация вычислительных процессов на терминальном уровне управления: Учеб. Пособие / В.В. Чекушкин - Владим. политехн. ин-т. – Владимир - 1993. - 52с.

54. Байков, В.Д. Решение траекторных задач в микропроцессорных системах ЧПУ / В.Д. Байков, С.Н. Вашкевич под. Ред. В.Б. Смолова. – Л.: Машиностроение, Ленингр. отд-ние - 1986. – 106с.

55. Сапельников, В.М. Цифроаналоговые преобразователи для воспроизведения тригонометрических функций / В.М. Сапельников, Р.А. Хакимов, Г.Ю. Коловертнов // Измерительная техника. - 2001. - № 3. - С.17-19.

56. Чекушкин, В.В. Методы построения цифровых синус-косинусных преобразователей индикатора кругового обзора / В.В. Чекушкин // Деп. рукопись РЖ серия общетехническая. – 1979. - №7.

57. Давыдов, П.С. Техническая диагностика радиоэлектронных устройств и систем / П.С. Давыдов. – М.: Радио и связь - 1988. – 256с.

58. Чекушкин, В.В. Моделирование структур цифровых аппроксиматоров для воспроизведения функций синуса на персональном компьютере / В.В. Чекушкин, О.В. Юрин // Измерительная техника. – 1999. – № 6 – С. 12-14.

59. Потапов, В.И. Устройство для вычисления функций синуса и косинуса А.С. № 622090 / В.И. Потапов, А.Н. Флоренков // 1978. – Бюл. № 32.

60. Мейзор, С. Быстродействующее устройство деления для специализированных ИС различного назначения / С. Мейзор, Б. Элтман // Электроника. – 1989. – № 14 – С. 29-35.

61. Чекушкин, В.В. Устройство для вычисления элементарных функций таблично-алгоритмическим методом / В.В. Чекушкин. - Патент № 2136041 // 1999. – Бюл. № 24.

62. Шикин, Е.В. Кривые и поверхности на экране компьютера. Руководство по сплайнам для пользователей / Е.В. Шикин, А.И. Плис – М.: ДИАЛОГ-МИФИ. - 1996. – 24 с.

63. Чекушкин, В.В. А.С. Устройство для вычисления элементарных функций табличным методом / В.В. Чекушкин - № 1442984 // Опубл. 1988. – Бюл. № 45.

64. Чекушкин, В.В. О построении цифровых кусочно-полиномиальных аппроксиматоров нулевого и первого порядка для воспроизведения функции синуса / В.В. Чекушкин, С.В. Чекушкин // Изв. вузов. Приборостроение. – 1999 – Т. 42. - № 9 – С. 42-45.

65. Чекушкин, В.В. Цифровые кусочно-полиномиальные аппроксиматоры для воспроизведения функций / В.В. Чекушкин // Приборы и системы управления. – 1999. – № 2. – С. 36-39.

66. Чекушкин, В.В. Программа поиска метода наилучшего приближения функции $R = \sqrt{A^2 + B^2}$ / В.В. Чекушкин, К.В. Михеев, И.В. Пантелеев // Свидетельство об официальной регистрации программы для ЭВМ №2015611477 от 29.01.2015.

67. Ercegovas, M. D. Reciprocation sguare root, inverse sgaure root, and some elementaru Sunctions Using small multipliers / M. D. Ercegovas // IEEE Trans Comput. 2000 - №7 - С. 628-637.

68. Микулович, В.Н. Цифровой алгоритм измерения амплитуды и фазы гармонических составляющих вибрации роторных машин / В.Н. Микулович, В.Т. Шнитко // Измерительная техника. – 1995. – № 4 – С. 41-43.

69. Чекушкин, В.В. Устройство для вычисления функции $X = \sqrt{A^2 + B^2}$ / В.В. Чекушкин АС № 964634 СССР // Опубл. - 1982. – Бюл. № 37.

70. Чекушкин, В.В. Устройство для вычисления квадратного корня / В.В. Чекушкин АС № 842806 СССР // Опубл. 1981. – Бюл. № 24.

71. Чекушкин, В.В. Реализация преобразования представлений ортогональных составляющих сигналов в амплитуду и фазу / В.В. Чекушкин // Измерительная техника – 2001. - №4 - С.18-21.

72. Альберт, Дж. Теория сплайнов и её приложения. Пер. с англ. / Дж. Альберт, Э. Нильсон, Дж. Уолт – М.: Мир - 1972. – 316с.

73. Дьяконов, В.В. MATLAB R2006/2007/2008 + Simulink 5/6/7. Основы применения. -2-е изд, перераб. и доп. / В.В. Дьяконов - М.: СОЛОН-ПРЕСС - 2008. - 800с.

74. Иглин, С.П. Математические расчеты на базе MATLAB / С.П. Иглин - СПб.: БХВ-Петербург - 2005. - 640с.

75. Михеев, К.В. Совершенствование полиномиальных методов воспроизведения функциональных зависимостей. / К.В. Михеев, А.Р. Синев // Всероссийская молодежная научная конференция «Научный потенциал молодежи – будущее России». Сб. тез. докладов VII Всероссийской научной конференции (Муром, 12 апреля 2015 г.). - Муром: Изд.-полиграфический центр МИ ВлГУ, 2015. - С. 651-652.

76. Архангельский, А.Я. Программирование в C++ Builder 6 / А.Я. Архангельский, М.А. Тагин - 2006. – М.: ООО «Бином-Пресс» – 1184с.

77. Батоврин, В.К. LabVIEW: практикум по основам измерительных технологий / В.К. Батоврин, А.С. Бессонов, В.В. Мошкин – М.: ДМК Пресс - 2009. – 232с.

78. Чекушкин, В.В. Способ имитации траекторий движения воздушных объектов / В.В. Чекушкин, М.С. Бобров, А.М. Аверьянов // Патент на изобретение № 2419072 от 01.06.2009.

79. Чумакова, Е.В. Алгоритмы операций умножения и деления для реализации на ПЛИС / Е.В. Чумакова // Проектирование и технология электронных средств. 2005. - №2. - С.54-57.

80. Чекушкин, В.В. Методы повышения эффективности реализации вычислительных структур. / В.В. Чекушкин, К.В. Михеев, А.Б. Оранский // Радиолокационная техника: устройства, станции, системы РЛС-2015. Тезисы докладов Третьей Всероссийской научно-практической конференции акционерного общества «Муромский завод радиоизмерительных приборов». - 9-10 июня 2015. - С.68-69.

81. Опадчий, Ю.Ф. Исследование методов вычислений элементарных математических функций и их реализация на ПЛИС / Ю.Ф. Опадчий, Е.В. Чумакова // Информационные технологии. 2013. - №4. - С.52-56.

82. Ashrafi, A. Theoretical upperbound of the spurious free dynamic range in direct digital frequency synthesizers realized by polynomial interpolation metods / A. Ashrafi, R. Adhami // IEEE Trans. Circuit Syst. I, Reg. Papers – 2007. - vol.54, p.2252-2261.

83. Тревис Дж. Lab VIEW для всех / Дж. Тревис - М: ДМК.: Пресс; Прибор Комплект. - 2004. – 544с.

84. Чекушкин, В.В. Создание банка данных воспроизведения стандартных функций с диапазоном представления от 3 до 64 двоичных разрядов. / В.В. Чекушкин, К.В. Михеев // Наука и образование в развитии промышленной, социальной и экономической сфер регионов России. VIII Всероссийские научные Зворыкинские чтения: сб. тез. докл. Всероссийской межвузовской научной конференции. Муром, 5 февр. 2016. – С.174-175.

85. Гришин, В.Ю. Совершенствование методов, математических моделей реализации вычислительных процессов в радиолокационных системах / В.Ю. Гришин, И.В. Пантелеев, В.В. Чекушкин // Вестник воздушно-космической обороны - 2014. №3 - С.20-25.

86. Чекушкин, В.В. Способ калибровки измерительных систем / В.В. Чекушкин, В.В. Булкин // Патент № 2262713 от 20.10.2005.

87. Чекушкин, В.В. Способ калибровки измерительных систем / В.В. Чекушкин, И.В. Пантелеев, А.Д. Богатов // Патент №2476896 от 27.02.2013.

88. Чекушкин, В.В. Повышение точности измерительных систем с нестабильными параметрами / В.В. Чекушкин, В.В. Булкин // Измерительная техника – 2006. - №1 – С. 7-11.

89. Пантелеев, И.В. Улучшение методов калибровки измерительных систем / И.В. Пантелеев, В.В. Чекушкин // Проектирование и технология электронных средств. - 2011. - №1 - С. 20-24.

90. Антуфьев, Р.В. Способ и устройство имитации радиолокационной информации / Р.В. Антуфьев, М.С. Бобров, Г.Г. Пискунов, В.В. Чекушкин, И.В. Пантелеев, М.А. Царьков // Патент на изобретение № 2489753 от 10.08.2013.

91. Гришин, В.Ю. Совершенствование полиномиальных методов воспроизведения функций в цифровых системах обработки информации / В.Ю. Гришин, И.В. Пантелеев, И.Р. Сарибжанов, В.В. Чекушкин // Радиопромышленность. - 2012. - № 2. - С.63-68.

92. Чекушкин, В.В. Способ и устройство вычисления квадратного корня // В.В. Чекушкин, А.М. Аверьянов, А.Д. Богатов // Патент №2438160 от 27.12.2011.

93. Демидович, Б.П. Основы вычислительной математики / Б.П. Демидович, И.А. Марон – М.: Физматгиз - 1963. – 660с.

94. Аверьянов, А.М. Улучшение методов преобразования ортогональных составляющих сигнала в амплитуду / А.М. Аверьянов, В.В. Чекушкин // Приборы и системы: управление, контроль, диагностика.– 2009. - №9 - С. 46–51.

95. Пантелеев, И.В. Совершенствование полиномиальных методов воспроизведения тригонометрических функций в информационно-вычислительных системах / И.В. Пантелеев, В.В. Чекушкин // Радиотехнические и телекоммуникационные системы. 2013. - № 1. - С.53-59.

96. Иглин, С.П. Математические расчеты на базе MATLAB / С.П. Иглин - СПб.: БХВ-Петербург - 2005. - 640с.

97. Миронов, И.Я. Устройство для извлечения квадратного корня из суммы квадратов / Ю.В. Малинин, Т.Г. Лазебник, Л.И. Новикова. А.С. №1001094 А. от 28.02.1983.

98. Hagan, P. Interpolation Methods for Curve Construction / P. Hagan, G. West // Applied Mathematical Finance – 2006. № 2, p. 89-129.

99. Безяев, В.С. Тренажерный комплекс подсистемы управления средствами ПВО / В.С. Безяев, А.Н. Воробьев // Вопросы радиоэлектроники; серия ЭВТ.- 2008 - С. 17-24.

100. А.В. Гусев, Тренажер оператора локационных станций / А.С. № 991479 // 1983 Бюл. №3.

101. Faraway, J. Reed M. and Wang J. Modeling 3D trajectories using Bezier curves with application to hand motion Applied Statistics / J. Faraway, M. Reed, J. Wang. // 2007. - №56. - p. 571-585.

102. Barrientos, A. UAV trajectory generation: planning and guidance / A. Barrientos, P. Gutierrez, J. Colorado // Aerial Vehicles. – Austria, Vienna - 2009. - p. 55– 82.

103. Chekushkin, V.V. Fast search algorithms for the best approximation polynomials for reproduction of functional dependences in data-measurement systems / V.V. Chekushkin, K.V. Mikheev // Measurement Techniques. – 2016, Volume 59, Issue 4, PP. 351-356. ISSN 0543-1972.

104. Чекушкин, В.В. Математическое моделирование и вычислительные алгоритмы в радиотехнических системах / В.В. Чекушкин, К.В. Михеев // НОЦ ВКО «Алмаз-Антей» им. академика В.П. Ефремова I Всероссийская научно-техническая конференция «Математическое моделирование и инженерные расчеты» - 2016. Сб. тезисов докладов – С.28.

Приложение 1 – Листинг программы поиска полинома наилучшего приближения для среды C++ Builder №6

```
//---------------------------------------------------------------------------
#include <vcl.h>
#pragma hdrstop
#include <iostream.h>
#include <math.h>
#include <stdio.h>
#include <string.h>
#include <Math.hpp>
#include <stdlib.h>
#include <vcl/dstring.h>
#include "Unit1.h"
#include "Unit2_funkcii.cpp"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
#pragma package(smart_init)
#pragma resource "*.dfm"
//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
        : TForm(Owner){
  int i;
 StringGrid1->Cells[0][0]="№ Полинома";
  for(i=1;i < StringGrid1->ColCount;i++)
    StringGrid1->Cells[i][0]=IntToStr(i);
  StringGrid1->Cells[0][1]="тчк. ин.";
  StringGrid1->Cells[0][2]="коэфф.";
  StringGrid1->Cells[0][3]="дискрет  =";
  StringGrid1->Cells[1][3]=0.01;
  StringGrid1->Cells[2][3]="       погр. метода =";
  StringGrid1->Cells[3][3]=0;
  StringGrid1->Cells[4][3]="погр. дискретизации  =";
  StringGrid1->Cells[5][3]=0;
  StringGrid1->Cells[6][3]="    погр. суммарная =";
  StringGrid1->Cells[7][3]=0;
  Edit1->Text=4;
  Edit2->Text=0.707;
  Edit3->Text=1;
  ComboBox1->ItemIndex=6;
  ComboBox2->ItemIndex=0;}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button1Click(TObject *Sender){
  int i,j;
  n = Edit1->Text.ToDouble();
  a = Edit2->Text.ToDouble();
  b = Edit3->Text.ToDouble();
  funk=ComboBox1->ItemIndex;
  vid_pol=ComboBox2->ItemIndex;
  arg();
```

```
ZapolnenieMatrici();
gaus();
str=" Кол-во членов полинома:";
str+=Edit1->Text;
str+=";  Используемые члены полинома:";
switch(vid_pol){
  case 0:str+=" все;";
  break;
  case 1:str+=" нечётные;";
  break;
  case 2:str+=" чётные;";
  break; }
str+="  Апроксимируемая функция:";
switch(funk){
  case 0: str+="sin(x)"; break;
  case 1: str+="cos(x)"; break;
  case 2: str+="tan(x)"; break;
  case 3: str+="asin(x)"; break;
  case 4: str+="atan(x)"; break;
  case 5: str+="sqrt(x)"; break;
  case 6: str+="1/(x)"; break;
  case 7: str+="1/sqrt(x)"; break;
  case 8: str+="exp^x"; break; }
for(j=1;j<11;j++)      {
      if(a!=0 || funk==1 || funk==6 || funk==7||RadioButton1->Checked==true)
      StringGrid1->Cells[j][0]=j-1;
      else
      StringGrid1->Cells[j][0]=j;

      StringGrid1->Cells[j][1]=0;
      StringGrid1->Cells[j][2]=0;      }
StringGrid1->Cells[3][3]=0;
StringGrid1->Cells[5][3]=0;
StringGrid1->Cells[7][3]=0;
for(j=n+1;j>=1;j--)      {
       if(a!=0 || funk==1 || funk==6 || funk==7||RadioButton1->Checked==true){
         StringGrid1->Cells[j][1]=argument[n+1-j];
         StringGrid1->Cells[j][2]=Koeff[n+1-j];         }
       else{
         StringGrid1->Cells[j][1]=argument[n-j];
         StringGrid1->Cells[j][2]=Koeff[n-j];   }     }
Series1->Clear();
Series2->Clear();
Series3->Clear();
Series4->Clear();
max_p=0;
for(i=0;i<=200;i++)   {
   x2=absPogr(gg*i+a);
   x2=fabs(x2);
   Series1->AddXY(gg*i+a,absPogr(gg*i+a),"",clRed);
   if(max_p<x2) max_p=x2;   }
StringGrid1->Cells[3][3]=max_p;}
```

```
//---------------------------------------------------------------------------
void __fastcall TForm1::Button2Click(TObject *Sender){
if(flag==0)  {
  schet=0;
  kl=1;
  flag=1;  }
else  {
  flag=0;
  Edit5->Text=schet;
  }}
//---------------------------------------------------------------------------
void __fastcall TForm1::Timer1Timer(TObject *Sender){
if(flag==1){
schet++;
switch(funk){
case 0:
  if(a!=0 || RadioButton1->Checked==true)
    exstr=1;
  else
    exstr=0;
break;
case 1:
 exstr=1;
break;
case 2:
  if(a!=0 || RadioButton1->Checked==true)
    exstr=1;
  else
    exstr=0;
break;
case 3:
  if(a!=0 || RadioButton1->Checked==true)
    exstr=1;
  else
    exstr=0;
break;
case 4:
  if(a!=0 || RadioButton1->Checked==true)
    exstr=1;
  else
    exstr=0;
break;
case 5:
  if(a!=0 || RadioButton1->Checked==true)
    exstr=1;
  else
    exstr=0;
break;
case 6:
  exstr=1;
break;
case 7:
```

```
  exstr=1;
break;}
 int i,j,k;
 if(flag==1) {
  Series1->Clear();
  Series2->Clear();
  Series3->Clear();
  Series4->Clear();
  for(i=0;i<11;i++) x[i]=0;
  max_p1=max_p;
  max_p=0;
  k=1;
  for(i=0;i<=200;i++)   {
     x2=absPogr(gg*i+a);
     x2=fabs(x2);
     Series1->AddXY(gg*i+a,absPogr(gg*i+a),"",clRed);
     if(RadioButton1->Checked==true){
       if(i==0) x[0]=x2;
       if(argument[n-k]>=(gg*i+a)){
         if(x[k]<x2) x[k]=x2;
         if(max_p<x2) max_p=x2;       }
       else
         k++;
       if(i==200) x[n+1]=x2;     }
     if(RadioButton2->Checked==true){
       if(argument[n-k]>=(gg*i+a)){
         if(x[k]<x2) x[k]=x2;
         if(max_p<x2) max_p=x2;       }
       else
         k++;
       if(i==200) x[n+1]=x2;     }    }
  if(max_p1<max_p) kp++;
  if(kp > 15)    {
      kl*=0.1;
      kp=0;
      Edit5->Text=schet;    }
  if(RadioButton1->Checked==true){
    for(i=0;i<(n+1);i++){
      if(x[i]<x[i+1])
        argument[n-i]+=kl*gg;
      else
        argument[n-i]-=kl*gg;    }  }
  if(RadioButton2->Checked==true){
    for(i=1;i<(n+1);i++){
      if(x[i]<x[i+1])
        argument[n-i]+=kl*gg;
      else
        argument[n-i]-=kl*gg;    }  } }
  ZapolnenieMatrici();
  gaus();
  str=" Кол-во членов полинома:";
  str+=Edit1->Text;
```

```
  str+="; Используемые члены полинома:";
  switch(vid_pol){
    case 0:str+=" все;";
    break;
    case 1:str+=" нечётные;";
    break;
    case 2:str+=" чётные;";
    break; }
  str+=" Приближаемая функция:";
  switch(funk){
    case 0: str+="sin(x);"; break;
    case 1: str+="cos(x);"; break;
    case 2: str+="tan(x);"; break;
    case 3: str+="asin(x);"; break;
    case 4: str+="atan(x);"; break;
    case 5: str+="sqrt(x);"; break;
    case 6: str+="1/(x);"; break;
    case 7: str+="1/sqrt(x);"; break;
    case 8: str+="exp^x"; break; }
  str+=" Коэффициенты полинома:";
  for(j=1;j<11;j++)
    if(a!=0 || funk==1 || funk==6 || funk==7 ||RadioButton1->Checked==true)
      StringGrid1->Cells[j][0]=j-1;
    else
      StringGrid1->Cells[j][0]=j;
  for(j=n+1;j>=1;j--)      {
       if(a!=0 || funk==1 || funk==6 || funk==7 ||RadioButton1->Checked==true){
         StringGrid1->Cells[j][1]=argument[n+1-j];
         StringGrid1->Cells[j][2]=Koeff[n+1-j];
         str+=" k";
         str+=j;
         str+="="+ StringGrid1->Cells[j][2];          }
       else{
         StringGrid1->Cells[j][1]=argument[n-j];
         StringGrid1->Cells[j][2]=Koeff[n-j];
         str+=" k";
         str+=j;
         str+="="+ StringGrid1->Cells[j][2]; }       }
  StringGrid1->Cells[3][3]=max_p;
  str+="; Погрешность метода ="+ StringGrid1->Cells[3][3];
 if(kl<0.01)   {
    kl=1;
    flag=0;
    Edit5->Text=schet;   }}}
//---------------------------------------------------------------------------
 // постройка графика погрешности по точкам интерполяции
void __fastcall TForm1::Button3Click(TObject *Sender){
  int i,j;
  for(j=n+1;j>=1;j--)
    if(a!=0 || funk==1 || funk==6 || funk==7 ||RadioButton1->Checked==true)
      argument[n+1-j]=StringGrid1->Cells[j][1].ToDouble();
    else
```

```
    argument[n-j]=StringGrid1->Cells[j][1].ToDouble();
  ZapolnenieMatrici();
  gaus();
  for(j=1;j<11;j++)       {
       StringGrid1->Cells[j][1]=0;
       StringGrid1->Cells[j][2]=0;
       StringGrid1->Cells[j][3]=0;      }
  for(j=n+1;j>=1;j--)       {
        if(a!=0 || funk==1 || funk==6 || funk==7){
          StringGrid1->Cells[j][1]=argument[n+1-j];
          StringGrid1->Cells[j][2]=Koeff[n+1-j];          }
        else{
          StringGrid1->Cells[j][1]=argument[n-j];
          StringGrid1->Cells[j][2]=Koeff[n-j];   }       }
  Series1->Clear();
  Series2->Clear();
  Series3->Clear();
  Series4->Clear();
  max_p=0;
  for(i=0;i<=200;i++)   {
    x2=absPogr(gg*i+a);
    x2=fabs(x2);
    Series1->AddXY(gg*i+a,absPogr(gg*i+a),"",clRed);
    if(max_p<x2) max_p=x2;   }
  StringGrid1->Cells[3][3]=max_p; }
//---------------------------------------------------------------------------
 // постройка графика погрешности по коэффициентам
void __fastcall TForm1::Button4Click(TObject *Sender){
  int i,j;
  StringGrid1->Cells[1][3]=0;
  str=" Кол-во членов полинома:";
  str+=Edit1->Text;
  str+=";  Используемые члены полинома:";
  switch(vid_pol){
    case 0:str+=" все;";
    break;
    case 1:str+=" нечётные;";
    break;
    case 2:str+=" чётные;";
    break;  }
  str+="  Апроксимируемая функция:";
  switch(funk){
    case 0: str+="sin(x)"; break;
    case 1: str+="cos(x)"; break;
    case 2: str+="tan(x)"; break;
    case 3: str+="asin(x)"; break;
    case 4: str+="atan(x)"; break;
    case 5: str+="sqrt(x)"; break;
    case 6: str+="1/(x)"; break;
    case 7: str+="1/sqrt(x)"; break;
    case 8: str+="exp^x"; break;  }
  str+=";  Коэффициенты полинома: ";
```

```
  for(j=n+1;j>=1;j--)
    if(a!=0 || funk==1 || funk==6 || funk==7 ||RadioButton1->Checked==true)   {
        Koeff[n+1-j]=StringGrid1->Cells[j][2].ToDouble();
        str+="  k";
        str+=j;
        str+="   "+ StringGrid1->Cells[j][2];      }
    else      {
        Koeff[n-j]=StringGrid1->Cells[j][2].ToDouble();
        str+="  k";
        str+=j;
        str+="   "+ StringGrid1->Cells[j][2];      }
  Series1->Clear();
  Series2->Clear();
  Series3->Clear();
  Series4->Clear();
  max_p=0;
  for(i=0;i<=200;i++)   {
     x2=absPogr(gg*i+a);
     x2=fabs(x2);
     Series1->AddXY(gg*i+a,absPogr(gg*i+a),"",clRed);
     if(max_p<x2) max_p=x2;   }
  StringGrid1->Cells[3][3]=max_p;
  str+=";  погрешность метода ="+ StringGrid1->Cells[3][3];}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button5Click(TObject *Sender){
float hh,gg;
int i,j;
long f,ll;
hh=StrToFloat(Edit4->Text);
gg=hh;
  for(j=n+1;j>=1;j--)
    if(a!=0 || funk==1 || funk==6 || funk==7 ||RadioButton1->Checked==true)  {
        Koeff[n+1-j]=StringGrid1->Cells[j][2].ToDouble();
        ll=1;
        for(hh;hh>0;hh--)
          ll=ll*10;
        hh=gg;
        f=Koeff[n+1-j]*ll;
        Koeff[n+1-j]=(double)f/ll;   }
    else      {
        Koeff[n-j]=StringGrid1->Cells[j][2].ToDouble();
        ll=1;
        for(hh;hh>0;hh--)
          ll=ll*10;
        hh=gg;
        f=Koeff[n-j]*ll;
        Koeff[n-j]=(double)f/ll;    }
  for(j=n+1;j>=1;j--)      {
         if(a!=0 || funk==1 || funk==6 || funk==7){
           StringGrid1->Cells[j][1]=argument[n+1-j];
           StringGrid1->Cells[j][2]=Koeff[n+1-j];   }
         else{
```

```
        StringGrid1->Cells[j][1]=argument[n-j];
        StringGrid1->Cells[j][2]=Koeff[n-j];  } }}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button6Click(TObject *Sender){
  Series1->Clear();
  Series2->Clear();
  Series3->Clear();
  Series4->Clear();
  double param=0;
  double pol=0, rez=0,x1=0;
  long int j,i,fl=0;
  double pro=0,pro1=0;
  max_p_d=0;
  max_p_d1=0;
  dxt=StringGrid1->Cells[1][3].ToDouble();
  deskret=dxt;
  for(i=0;i<65000;i++){
   param=gg1*i+a;
   x1=param;
   pol=0;
   if(exstr)
   for(j=n;j>=0;j--){
    if(j==n) pol+=Koeff[j];
    else
    switch(vid_pol){
      case 0:     pol+=Koeff[j]*x1;     x1*=param;     break;
      case 1:     pol+=Koeff[j]*x1;     x1*=param;     x1*=param;     break;
      case 2:     x1*=param;     pol+=Koeff[j]*x1;     x1*=param;     break;    }  }
  else
   for(j=0;j<n;j++){
    switch(vid_pol){
      case 0:     pol+=Koeff[j]*x1;     x1*=param;     break;
      case 1:     pol+=Koeff[j]*x1;     x1*=param;     x1*=param;     break;
      case 2:     x1*=param;     pol+=Koeff[j]*x1;     x1*=param;     break;     }   }
   switch(funk)
    {
      case 0:     rez=sin(param);     break;
      case 1:     rez=cos(param);     break;
      case 2:     rez=tan(param);     break;
      case 3:     rez=asin(param);     break;
      case 4:     rez=atan(param);     break;
      case 5:     rez=sqrt(param);     break;
      case 6:     rez=1/param;     break;
      case 7:     rez=1/sqrt(param);     break;
      case 8:     rez=pow(M_E,param);     break;    }
     Series3->AddXY(param,rez-pol,"",clBlue);
     if(fl==1){
     Series2->AddXY(param,rez-pro,"",clGreen);
     Series4->AddXY(param,rez-pro1,"",clGreen);    }
     if(param>deskret){
      if(fl==1){
        deskret=double(deskret+dxt);
```

```
        if(max_p_d<fabs(rez-pro))
          max_p_d=fabs(rez-pro);
        if(max_p_d1<fabs(rez-pro1))
          max_p_d1=fabs(rez-pro1);        }
      pro=rez;
      pro1=pol;
      fl=1;      }    }
  str+=";  Значение дискрета = "+StringGrid1->Cells[1][3];
  StringGrid1->Cells[5][3]=max_p_d;
  str+=";  Погрешность дискретизации = "+StringGrid1->Cells[5][3];
  StringGrid1->Cells[7][3]=max_p_d1;
  str+=";  Суммарная погрешность = "+StringGrid1->Cells[7][3];}
//---------------------------------------------------------------------------
void __fastcall TForm1::SpeedButton1Click(TObject *Sender){
if ( SaveDialog1->Execute() )  {
    file1=fopen(SaveDialog1->FileName.c_str(),"wt");
    fputs(str.c_str(),file1);
    name=SaveDialog1->FileName.c_str();
    name+=".bmp";
    Chart1->SaveToBitmapFile(name);
    fclose(file1);
    str="";  }}
//---------------------------------------------------------------------------

//--- расчет первоначальных точек интерполяции --------------------
void arg(){
  int i,k,kk;
  for(i=0;i<10;i++)   {
    Koeff[i]=0;
    argument[i]=0;   }
  gg=(b-a)/200;
  gg1=(b-a)/65000;
  kx=(b-a)/(n);
  k=n;
  for (i=0;i<(n+1);i++)    {
      argument[i]=kx*k+a;
      k--;    }}
//--- подпрограмма заполнения матрици в согасии с типом полнома-----
void ZapolnenieMatrici(){
  int i,j;
  double agrL=0;
  for(i=0;i<20;i++)
    for(j=0;j<20;j++)
      MatrG[i][j]=0;
if(exstr)
  for(i=0;i<(n+1);i++){
    kx=argument[i];
    agrL=argument[i];
    for (j=n+1;j>=0;j--){
        if(j!=(n+1))         {
          if(j==n){
            MatrG[i][j]=1;            }
```

```
          else
          switch(vid_pol)           {
            case 0:           MatrG[i][j]=kx;           kx*=argument[i];          break;
            case 1:           MatrG[i][j]=kx;           kx*=argument[i];            kx*=argument[i];
break;
            case 2:           kx*=argument[i];            MatrG[i][j]=kx;            kx*=argument[i];
break;
           }        }
        else
         switch(funk)           {
            case 0: MatrG[i][j]= sin(agrL); break;
            case 1: MatrG[i][j]= cos(agrL); break;
            case 2: MatrG[i][j]= tan(agrL); break;
            case 3: MatrG[i][j]= asin(agrL); break;
            case 4: MatrG[i][j]= atan(agrL); break;
            case 5: MatrG[i][j]= sqrt(agrL); break;
            case 6: MatrG[i][j]= 1/agrL; break;
            case 7: MatrG[i][j]= 1/sqrt(agrL); break;
            case 8: MatrG[i][j]= pow(M_E,agrL); break;   // возведение экспоненты в степень           }
}  }
else
  for(i=0;i<n;i++){
    kx=argument[i];
    agrL=argument[i];
    for(j=0;j<(n+1);j++){
       if(j!=(n))    {
          switch(vid_pol)      {
            case 0:           MatrG[i][j]=kx;           kx*=argument[i];          break;
            case 1:           MatrG[i][j]=kx;           kx*=argument[i];            kx*=argument[i];
break;
            case 2:           kx*=argument[i];            MatrG[i][j]=kx;            kx*=argument[i];
break;} }
        else
         switch(funk)    {
            case 0: MatrG[i][j]= sin(agrL); break;
            case 1: MatrG[i][j]= cos(agrL); break;
            case 2: MatrG[i][j]= tan(agrL); break;
            case 3: MatrG[i][j]= asin(agrL); break;
            case 4: MatrG[i][j]= atan(agrL); break;
            case 5: MatrG[i][j]= sqrt(agrL); break;
            case 6: MatrG[i][j]= 1/agrL; break;
            case 7: MatrG[i][j]= 1/sqrt(agrL); break;
            case 8: MatrG[i][j]= pow(M_E,agrL); break; }    }  } }
//--- решение СЛАУ методом Гауса ------------------------------
void gaus(){
 char m,i,j;
 char kk,pp,ll;
 if(exstr){
   kk=n;
   pp=n+2;
   ll=n+1; }
 else{
```

```
  kk=n-1;
  pp=n+1;
  ll=n; }
 for(m=0;m<kk;m++)   {
  for(i=(m+1);i<ll;i++)  {
    gkl=MatrG[i][m]/MatrG[m][m];
    if(MatrG[i][m]>0&&MatrG[m][m]>0 ||
      MatrG[i][m]<0&&MatrG[m][m]<0)gkl*=-1;
    for(j=m;j<pp;j++)      {
      gkl_1=MatrG[m][j];
      gkl_2=gkl_1*gkl;
      MatrG[i][j]=gkl_2+MatrG[i][j];     }    }   }
  kx=0;                  // расчитываем корни уравнений
 if(exstr)
  for(i=kk;i>=0;i--)    {
     Koeff[i]= (MatrG[i][n+1]-kx) / MatrG[i][i];
     kx=0;
     for(j=kk;j>=i;j--)
      if(i>=1) kx += MatrG[i-1][j] * Koeff[j];    }
 else
  for(i=kk;i>=0;i--)    {
     Koeff[i]= (MatrG[i][n]-kx) / MatrG[i][i];
     kx=0;
     for(j=kk;j>=i;j--)
      if(i>=1) kx += MatrG[i-1][j] * Koeff[j];    }  }
//-----------------------------------------------------------------------
double absPogr(double param){
  double pol=0, rez=0,x1=0;
  int j;
  x1=param;
 if(exstr)
  for(j=n;j>=0;j--){
   if(j==n) pol+=Koeff[j];
   else
   switch(vid_pol){
     case 0:     pol+=Koeff[j]*x1;     x1*=param;     break;
     case 1:     pol+=Koeff[j]*x1;     x1*=param;     x1*=param;     break;
     case 2:     x1*=param;     pol+=Koeff[j]*x1;     x1*=param;     break;     }  }
 else
  for(j=0;j<n;j++){
   switch(vid_pol){
     case 0:     pol+=Koeff[j]*x1;     x1*=param;     break;
     case 1:     pol+=Koeff[j]*x1;     x1*=param;     x1*=param;     break;
     case 2:     x1*=param;     pol+=Koeff[j]*x1;     x1*=param;     break;     } }
  switch(funk)   {
     case 0:     rez=sin(param) - pol;     break;
     case 1:     rez=cos(param) - pol;     break;
     case 2:     rez=tan(param) - pol;     break;
     case 3:     rez=asin(param) - pol;     break;
     case 4:     rez=atan(param) - pol;     break;
     case 5:     rez=sqrt(param)-pol;     break;
     case 6:     rez=1/param-pol;     break;
```

```
    case 7:     rez=1/sqrt(param)-pol;    break;
    case 8:     rez=pow(M_E,param)-pol;    break;  }
return rez;}
//---------------------------------------------------------------------------
```

Приложение 2 – Акты внедрения результатов диссертационной работы

УТВЕРЖДАЮ

первый заместитель директора МИ ВлГУ

д.т.н., профессор Жизняков А.Л.

____

(подпись)

«07» _______________ 2016 г.

(дата)

АКТ ВНЕДРЕНИЯ

результатов диссертационной работы

Михеева Кирилла Валерьевича

на тему «Разработка вычислительных алгоритмов для устройств обработки и отображения информации радиотехнических систем»

Настоящий акт составлен о том, что материалы диссертационной работы Михеева К.В., представленной на соискание ученой степени кандидата технических наук по специальности 05.12.04 – Радиотехника, в том числе системы и устройства телевидения, внедрены в учебный процесс на кафедре систем автоматизированного проектирования МИ ВлГУ и используются при чтении лекций и проведении практических работ по дисциплине: «Современные микропроцессорные технологии управления и обработки информации». По материалам диссертационной работы разработаны и внедрены в учебный процесс три лабораторных работы, а именно:

1. Алгоритмы поиска полиномов наилучшего приближения для воспроизведения функциональных зависимостей.

2. Улучшение методов вычисления стандартных функций в системах обработки информации полиномиальными методами на примере функции sin(x).

3. Методы и математические модели воспроизведения траекторий движения объектов.

Заведующий лабораториями кафедры САПР — А.Н. Пичугин

Зам. заведующего кафедрой САПР — Я.Ю. Кульков

Начальник НИСа МИ ВлГУ
к.т.н., доцент — Д.Г. Привезенцев

УТВЕРЖДАЮ

Технический директор АО «МЗ РИП»

Мошнин А.К.

(подпись)

«07» 11 2016 г.

(дата)

АКТ ВНЕДРЕНИЯ

результатов кандидатской диссертационной работы

Михеева Кирилла Валерьевича

на тему «Разработка вычислительных алгоритмов для устройств обработки и отображения информации радиотехнических систем»

Настоящий акт подтверждает, что материалы диссертационной работы Михеева К.В. позволют эффективно обеспечивать тренировку расчётов РЛС в условиях адекватных современному состоянию процессов по отражению нападения воздушно-космических средств в зоне контроля воздушного пространства, а также функционально-диагностический контроль и моделирование систем обработки и отображения информации. Используются на АО «Муромский завод радиоизмерительных приборов» при модернизации изделия 64Л6М в части тренажно-имитационной системы, а именно:

- разработанные численные методы воспроизведения стандартных функций с исключением избыточной точности с диапазоном приведённых погрешностей выходных данных до долей %;

- алгоритмы, обеспечивающие многократное сокращение числа итерационных циклов при калибровке измерительных каналов с нестабильными параметрами и разрядных сеток операндов специализированных вычислителей на 2-5 двоичных разрядов;

- метод формирования траекторий движения воздушных объектов с контролем перегрузок, позволяющий повысить качество тренажной информации операторов РТС;

- прикладное программное обеспечение, позволяющее автоматизировать процесс поиска полиномов наилучшего приближения для различных функциональных зависимостей, проводить оптимизацию полиномов под различные спецвычислители, применяемые на АО «МЗ РИП» и построение траекторий движения воздушных объектов.

Зам. главного конструктора
Заслуженный конструктор РФ

Д.Д. Богатов