

УДК 623.746.-519+629.7.018.2

Разработка центральной вычислительной системы барражирующей крылатой ракеты

Клёнов И.Л., студент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
Аэрокосмический факультет, кафедра «Системы автоматизированного управления»*

Научный руководитель: Жигулёвцев Ю.Н., к.т.н, профессор

Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана

bauman@bmstu.ru

Барражирующая крылатая ракета (комплекс БПЛА) – беспилотный ЛА среднего класса, предназначенный для загоризонтного обнаружения целей и целеуказания противокорабельным крылатым ракетами, с запуском с подводного носителя. Более подробно специфику и особенности применения аппарата см. [1]. Общий вид проектируемого БПЛА БКР и характерное распределение давлений в аэродинамическом потоке в проекции на корпус аппарата показаны на рис. 1:

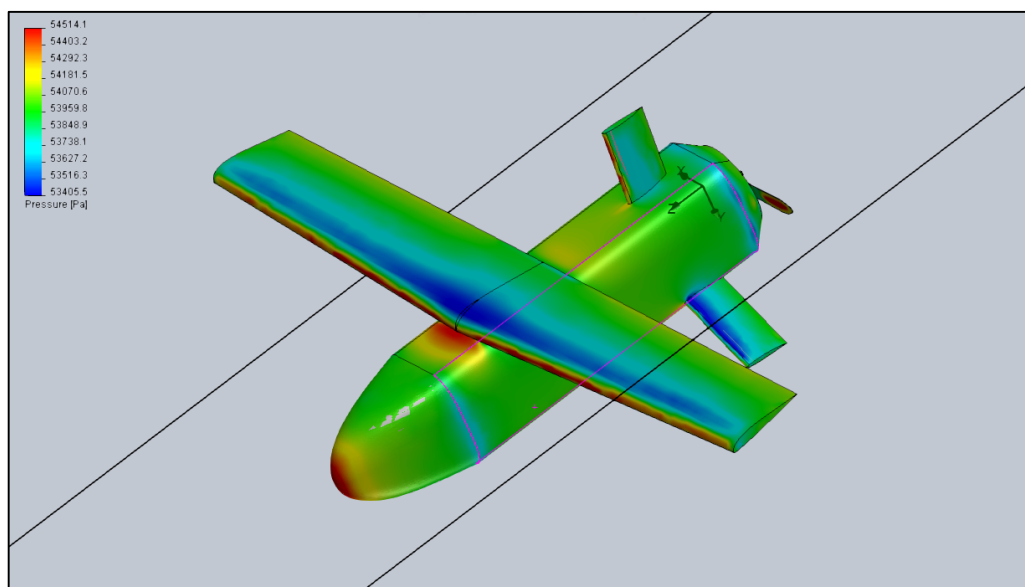


Рис. 1. Схема распределения давлений в проекции на геометрию аппарата

БЦВМ, как центральный компонент проектируемой системы, представляет собой бортовой вычислитель, реализуемый в рамках концепции децентрализованной системы

управления на базе так называемого адаптерного подхода. Децентрализованная система имеет такое название потому, что функции, исполняемые в “классической” системе управления единым крупным центральным модулем, в такой системе распределены между совокупностью небольших модулей, каждый из которых решает свою конкретную задачу. Функцией центрального модуля при таком подходе становится реализация общего закона управления и обеспечение обмена данными между отдельными модулями.

В процессе работы над ЦВС были использованы современные приемы управления проектными разработками, часто применяемые крупными зарубежными аэрокосмическими концернами (NASA, Lockheed Martin, Tesla Industries и проч.) – такие, как:

В рамках спиральной модели процесс разработки аппарата (или ПО) проходит несколько итераций или “витков спирали”, завершение каждого из которых соответствует созданию функционально законченного образца (прототипа), см. рис. 2. Важным достоинством спиральной модели является совмещение проектирования и поэтапного прототипирования в непрерывный процесс с минимумом проектных рисков.

Рис. 2. Спиральная модель разработки ПО технически сложной системы [2]

Методика модельно-ориентированного проектирования (МОП) изначально была создана для проектирования технически сложных изделий в рамках спиральной модели. МОП – это современная высокоэффективная методика разработки сложных программно-аппаратных комплексов, позволяющая создать надёжный и качественный продукт путём сведения всех этапов работы с системой к взаимодействию с единой структурно-имитационной моделью комплекса. В ходе данной работы использовались спиральная модель и МОП, они сравнивались с таким традиционным методом управления НИОКР, как итерационная каскадная модель ведения проекта (она же модифицированная “Модель водопада”, англ. Waterfall method).

Основной механизм МОП – это **абстракция**, то есть введение конструкций, позволяющих кратко и наглядно в общем виде построить **модель** комплекса, процесса или явления. Обычно в качестве таких конструкций выступают **блочные диаграммы** – блок-схемы, приведённые к строго установленной форме.

При этом каждый из элементов (шагов блок-схемы) обычно соответствует совершению атомарной операции над величиной или сигналом в системе. Под атомарностью здесь понимается логическая неразделимость операции на более мелкие.

Применение механизма абстракции позволяет переложить реализацию частных на сторонние (многократно верифицированные) инструменты разработки. Такими инструментами являются автоматические трансляторы, без участия человека создающие “нижний уровень” ПО на основе сертифицированных и многократно оттестированных алгоритмов.

Достигается более чем двойная экономия времени и средств разработчиков – не требуется заново проектировать, разрабатывать и тестировать универсальные и платформенно-зависимые компоненты.

Дополнительно, разработанное высокоуровневое решение обладает свойством переносимости. Даже замена всего аппаратного обеспечения системы на полностью несовместимое с прежним - другая разрядность, системы адресации и т.п. – потребует от разработчика всего лишь перенастроить транслятор.

Модель в МОП позволяет, не создавая множество физических прототипов, проводить достоверные имитационные компьютерные эксперименты. Это избавляет разработчика от необходимости создавать отдельно имитационную, отдельно реализуемую на реальном оборудовании модели – а, значит, экономит его время и усилия.

Программное обеспечение БЦВМ организовано согласно иерархическим принципам. В целях разделения логики исполнения кода, минимизации ущерба от возможных ошибок

и обеспечения гибкости продукта, ПО разделено по уровням абстракции. Схема представленной архитектуры ПО БЦВМ изображена на рис. 3:



Рис. 3. Иерархическая организация ПО БЦВМ

Здесь в составе ПО БЦВМ между пользовательским (*высокоуровневым*) кодом и алгоритмами, взаимодействующими напрямую с аппаратурой (“нижним уровнем”) присутствует т.н. слой аппаратных абстракций (Hardware Abstraction Layer, HAL) или, иначе, инфраструктурная прослойка.

Инфраструктурная прослойка (*Framework layer*) – компонент программного обеспечения, обеспечивающий взаимодействие пользовательского алгоритма с аппаратной частью. Такой слой нужен для того, чтобы уйти от деталей реализации конкретного множества функций. Потому ***Framework layer*** – это множество библиотек - «черных ящиков», производящих внутри все необходимые операции по работе с аппаратной частью.

Для подтверждения и экспериментальной проверки приведённых положений, на базе любительских микроконтроллеров серии Arduino была собрана концептуальная модель комплекса полунатурного моделирования с БЦВМ как центральным элементом контура управления. Более подробно устройство комплекса освещено в [4], а его функциональная схема представлена на рис. 4:

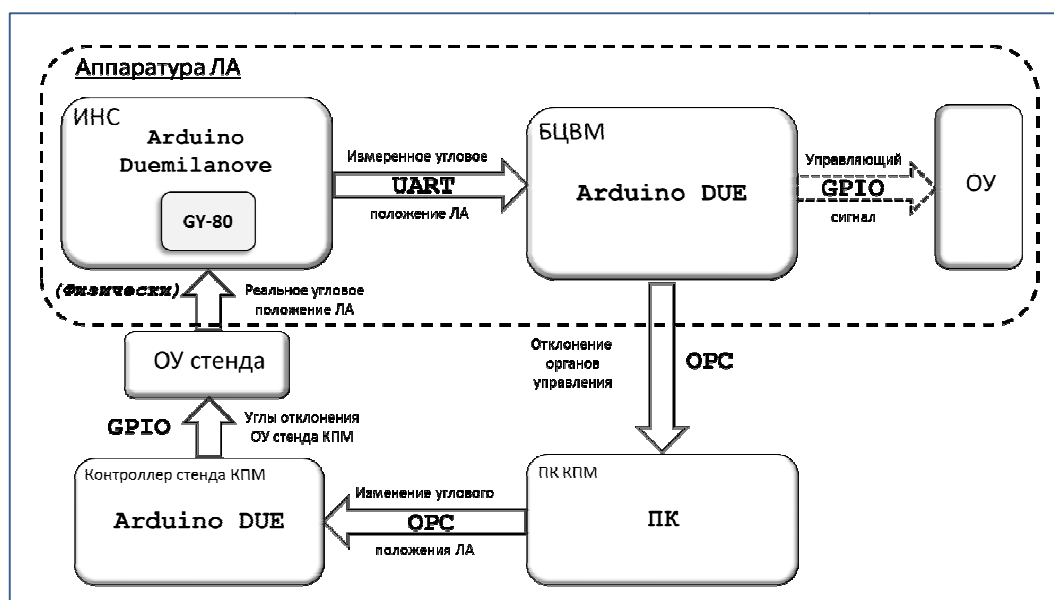


Рис. 4. Функциональная схема комплекса полунатурного моделирования

Протокол OPC позволяет в режиме реального времени обмениваться информацией между ПК и микроконтроллером, или между несколькими микроконтроллерами. Для микроконтроллеров Arduino компанией ISM Solar была создана совместимая со стандартом версия библиотеки OPC для обеспечения поддержки платами стандарта OPC. Однако использование данной библиотеки выявило ряд программных ошибок, о чём было сообщено разработчику. В сотрудничестве с ведущим инженером ISM Solar ошибки были диагностированы, обнаружены и исправлены. За помощь в исправлении ошибок компанией ISM Solar были вынесены неоднократные благодарности [5].

Помимо данной библиотеки и среды Matlab (Simulink), в состав ПО комплекса полунатурного моделирования также вошла инструментальная прослойка, выполняющая функции объединения алгоритмов, создаваемых инструментом Embedded Coder (Simulink Coder), и библиотек аппаратной платформы Arduino (см. рис. 5):



Рис. 5. Код разработанной инфраструктурной прослойки в IDE Arduino

Стандартная аппаратная прослойка, предоставляемая Simulink для обеспечения функционирования пользовательского кода на контроллерах Arduino, неудобна для использования, поскольку не имеет стандартного механизма для встраивания драйверов для внешних по отношению к микроконтроллеру устройств. Обойти это можно путём добавления в структуру самой модели всех алгоритмов преобразования сигнала к доступному для устройства виду, но подобный подход заметно перегружает модель и делает схему нечитаемой.

Разработанный прототип инфраструктурной прослойки позволяет:

- Создав для имитационной компьютерной модели управляемого объекта алгоритм управления в Simulink, в несколько кликов загружать его в реальный контроллер.
- В качестве входов и выходов алгоритма, работающего в контроллере, подключать как физически присоединённые к контроллеру реальные, так и виртуальные устройства - точные имитации реальных приборов, создаваемые средствами среды Simulink.
- Свободно управлять любыми подключаемыми устройствами с использованием единого механизма обмена данными через универсальный для устройств любого типа и природы программный интерфейс.
- “На лету” задавать набор и конфигурацию подключаемых устройств.
- Выводя и получая в реальном времени любые пользовательские данные из среды

Simulink, осуществлять полунатурное и имитационное моделирование.

Данная инфраструктурная прослойка спроектирована таким образом, что созданный код может быть выполнен в контроллере как самостоятельно, так и в рамках потоков и процессов ОС РВ.

Результаты работы подтвердили данные литературных источников – использование МОП в рамках спиральной модели позволяет в 2 и более раз сэкономить время и ресурсы по сравнению с каскадной моделью.

Совместное использование спиральной модели и МОП для разработки децентрализованной системы управления позволило решить задачи, упомянутые в начале статьи, в сравнительно короткие сроки.

Список литературы

1. Бельский С.Ф., Клёнов И.Л. Проектирование стенда имитационного моделирования на основе платформы Гью-Стюарта // Вопросы проектирования ЛА и КА: Научно-техническая конференция студентов Аэрокосмического факультета «Студенческая Научная Весна»: труды. Реутов: Изд-во НПОМаш, 2014. С. 34-35.
2. Павлов А.М. и др. Современное состояние и тенденции развития бортовых систем информационного обмена / под общ. ред. Е.А. Федосова. М.: НИЦ ГосНИИАС, 1991. 25 с.
3. Избачков Ю.А., Петров В.Н. Информационные системы: учебник для вузов. СПб.: «Питер», 2005. 655 с.
4. Клёнов И.Л. Концептуальная модель БЦВМ барражирующей крылатой ракеты «Выпь». БЦВМ как компонент комплекса полунатурного моделирования // Теория автоматического управления: КП. Реутов: Изд-во НПОМаш, 2013. С. 1-76.
5. Arduino OPC Support Forum :: Topic: Empty Variables (1/6) // Arduino OPC tools for makers 2013. Режим доступа: <http://www.arduinoopc.com/bugs/17-empty-variables> (дата обращения: 24.12.2013).