



Миссия «Свидание с Раму»

Аннотация

В рамках общекосмической подготовки, [мной](#), при поддержке [Анны Апальковой](#), с помощью программ-симуляторов и астродинамического расчётно-навигационного ПО, было выполнено натурное моделирование космической миссии, названной нами «Rendezvous with RAMU».

Цели и задачи миссии

Целью моделируемой миссии был перехват пилотируемым КА спутника на ССО, имитация выполнения ремонтных работ на спутнике, и возвращение пилотируемого КА на Землю

Практическими целями моделирования были:

- Получение опыта применения алгоритмов расчёта орбитальных манёвров, описанных в книге D.Vallado «Fundamentals of Astrodynamics and Applications. 4-th edition».
- Получение реалистичного опыта планирования космических миссий на НОО.
- Получение реалистичного опыта орбитального маневрирования.
- Развитие навыков работы со сложными техническими системами в стрессовых ситуациях в условиях неполноты данных и ограниченного времени.

В процессе подготовки к выполнению миссии пришлось решить большое количество дополнительных задач. В общей сложности процесс подготовки занял приблизительно 3 месяца.

Техническое обеспечение

Для моделирования миссии использовались:

1. ПК (AMD FX-8350, 32Gb RAM, 2xNVidia GeForce GTX770)
2. MacBook Air (Intel Core i5 1.6 GHz, 8Gb RAM, Intel HD Graphics 6000, MacOS Catalina)
3. Джойстики Logitech Extreme 3D Pro, Logitech Wingman Formula Force GP и Gametrix ECS
4. Orange Pi PC2 (Armbian)
5. Noname USB веб-камера
6. Телефоны Asus ZenPhone 5z и Huawei P10 Lite
7. Различные аппаратные средства локальной сети (точки доступа WiFi, коммутаторы и т.д.)

Использованное программное обеспечение:

1. Orbiter 2016 v.160828
 - a. Add-on Multistage Titans (авторы 4throck and Sputnik)
 - b. Add-on Space Tugs (автор Yuri Kulchitsky)
 - c. Add-on Well PMA-DG
 - d. Add-on Formosat-5 (автор BrianJ)
 - e. Add-on ReFuelMFD (автор Roman Starkov)
 - f. Сервисные add-оны (Multistage 2015, Spacecraft4, FlyByWire и др.)
 - g. Метамоды (add-он OrbiterSound, и др.)
2. GMAT R2020A
3. Python 3.9.2
 - a. Numpy (для вычислений с матрицами и векторами)
 - b. sgp4 (для удобного и быстрого конвертирования TLE в state vectors)
 - c. pyatmos (для моделирования атмосферы)
4. LUA 5.1.2
5. C# (Visual Studio 2019, .Net 5.0 Windows)
 - a. Библиотека Stateless 5.11.0
6. ImDisk Virtual Disk Driver for Windows NT/2000/XP/2003, ver. 2.1.1
7. LinPhone Android 4.6.9 + Asterisk 18.12.x
8. Различное прикладное ПО (для обработки векторной графики, чтения .pdf-файлов, и т.д.)

Моделируемые космодромы

Пусковая площадка «Лисий Нос»

Вымышленный космодром, использующийся для пусков частных космических агентств. Расположен возле Санкт-Петербурга, в точке с координатами 60.0° с.ш. и 30.0° в.д., возле посёлка Лисий Нос.

Kennedy Space Center

Широко известный американский космодром, расположенный на мысе Канаверал. Координаты 28.62° с.ш. и 80.62° з.д.

Edwards Air Force Base

Авиабазы Эдвардс, принадлежащая ВВС США. Доступна для посадки частных ЛА, имеет полосу приспособленную для приёма КА типа «Space Shuttle». Координаты 34.92° с.ш. и 117.89° з.д.

Моделируемые РН

Titan III

Снятая с производства РН американского производства. В данной миссии использовалась для выведения на НОО заправщиков RAMU. Двухступенчатая РН с двумя твердотопливными ускорителями на первой ступени, топливная пара ЖРД – гидразин/азотная кислота, полезная нагрузка на НОО приблизительно 13 т (в зависимости от модификации и особенностей конфигурации).

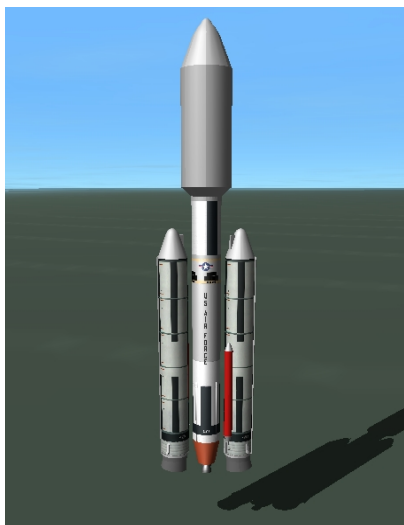


Рисунок 1 - виртуальная модель РН Titan III

Titan IVC “Barbarian”

Вымышленная РН, основанная на проекте РН Titan-Barbarian фирмы Martin-Marietta. Представляет собой Titan IVA с двумя дополнительными твердотопливными ускорителями. В данной миссии использовалась для вывода на орбиту КА DeltaGlider. Двухступенчатая РН с четырьмя твердотопливными ускорителями на первой ступени, топливная пара ЖРД – гидразин/азотная кислота, полезная нагрузка на НОО приблизительно 21 т.



Рисунок 2 - виртуальная модель РН Titan IVC "Barbarian" с установленным в качестве ПН КА DeltaGlider

Данные РН были выбраны по следующим причинам:

- 1.>Add-оны РН семейства Titan в Orbiter 2016 сделаны модульными, на основе библиотеки Multistage 2015, что облегчает их настройку и позволяет установку различных ПН.
2. В отличие от некоторых других адд-онов РН, они управляются стандартными средствами Orbiter, что позволяет эффективно управлять ими вручную или при помощи написанного самостоятельно автопилота.
3. РН семейства Titan реально и активно использовались, в свободном доступе имеется обширная документация по их конструкции и динамике полёта.
4. РН семейства Titan относительно реалистичны в рамках данной миссии. *(Сохранившаяся документация и запас запчастей обеспечивают возможность их постройки. Несмотря на дороговизну и токсичную топливную пару, данные РН проще в эксплуатации чем отечественные РН).*
5. РН семейства Titan используют ту же топливную пару, что и остальные КА, принадлежащие частному космическому агентству.

Динамика РН моделируется с помощью дополнительной библиотеки Multistage 2015. Эмулируемая авионика РН связана со стандартной авионикой симулятора посредством этой же библиотеки.

Моделируемые космические аппараты

Модифицированный ("химический") DeltaGlider

В качестве пилотируемого КА был выбран вымышленный космолан DeltaGlider, входящий в стандартный набор КА симулятора Orbiter 2016. Причиной выбора послужило то, что DeltaGlider является одним из немногих КА в Orbiter, для которых качественно смоделирована виртуальная кабина и функционирование внутренних систем.



Рисунок 3 - виртуальная модель КА DeltaGlider

Стандартный DeltaGlider в Orbiter 2016 оборудован нереалистичной Д.У., имеющей удельный импульс около 4000 сек. Для целей реалистичной симуляции такая Д.У. не подходила. Поэтому параметры Д.У. были изменены с помощью написанного мной LUA-скрипта, заменяющего значения силы тяги F_t и удельного импульса I двигателей. Значения F_t и I были взяты от реально существовавших двигателей, работавших на топливной паре гидразин/азотная кислота и близких по тяговым и габаритным характеристикам. После изучения параметров различных реально существовавших РД, были выбраны следующие параметры двигательной установки:

Двигатель	Количество	Тяга, N	У.И., м/с	Примечание
Маршевый	2	137300	3212.7	Аналог РД-857
Тормозной	2	19620	3223.5	Аналог С.5.98М (14Д30)
Вертикального взлёта	3	19620	3223.5	Аналог С.5.98М (14Д30)
Ориентации	18	1960	2661	Аналог КДУ-414

Все двигатели КА решено было считать модифицированными для обеспечения возможности многократного включения.

Основные характеристики КА:

Сухая масса	11000 кг
Масса топлива для маршевой Д.У.	12900 кг
Масса топлива для двигателей ориентации	600 кг
Масса полезной нагрузки	400 кг
Максимальная взлётная масса	24900 кг
Длина	17.76 м
Размах крыльев	17.86 м
Площадь несущей поверхности	158.60 м ²

NB! В данном документе везде, кроме тех случаев где явно указано иное, под названием *DeltaGlider* понимается его модифицированная («химическая») версия.

RAMU (Refueling Autonomously Managed Unit)

RAMU (Автономно Управляемый Модуль Дозаправки) - вымышленный КА-заправщик, сделанный на основе разгонного блока ДМ-2 из адд-она Space Tugs. RAMU «оборудован» (путём редактирования исходного кода модуля) стыковочным узлом стандарта АПАС-95 (совместимым со стыковочным узлом DeltaGlider). Так же как и другие КА, задействованные в миссии, RAMU использует топливную пару гидразин/азотная кислота.

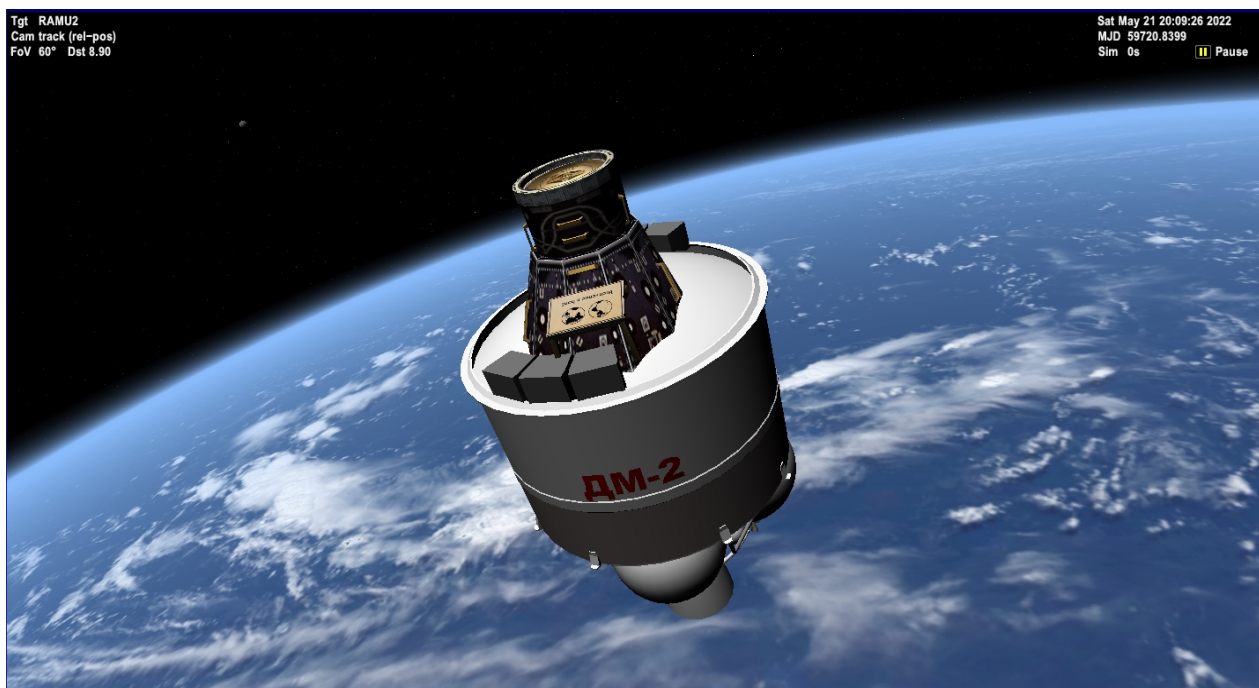


Рисунок 4 - виртуальная модель КА RAMU

Основные характеристики КА:

Сухая масса	3000 кг
Масса топлива для маршевой Д.У.	15000 кг
Масса топлива для двигателей ориентации	1500 кг
Максимальная взлётная масса	24900 кг
Длина	8.26 м
Диаметр	5.56 м
Тяга маршевой Д.У.	83356.525 Н
Количество включений маршевой Д.У.	7

В данной миссии в RAMU было заправлено только 7 т топлива для маршевой Д.У., и 750 кг топлива для RCS. Стартовая масса КА составила 11450 кг.

Название RAMU было выбрано, во многом, из-за внешнего и смыслового сходства КА-заправщика с жестяной бутылкой японского лимонада «Рамуне» - Ramu Bottle:

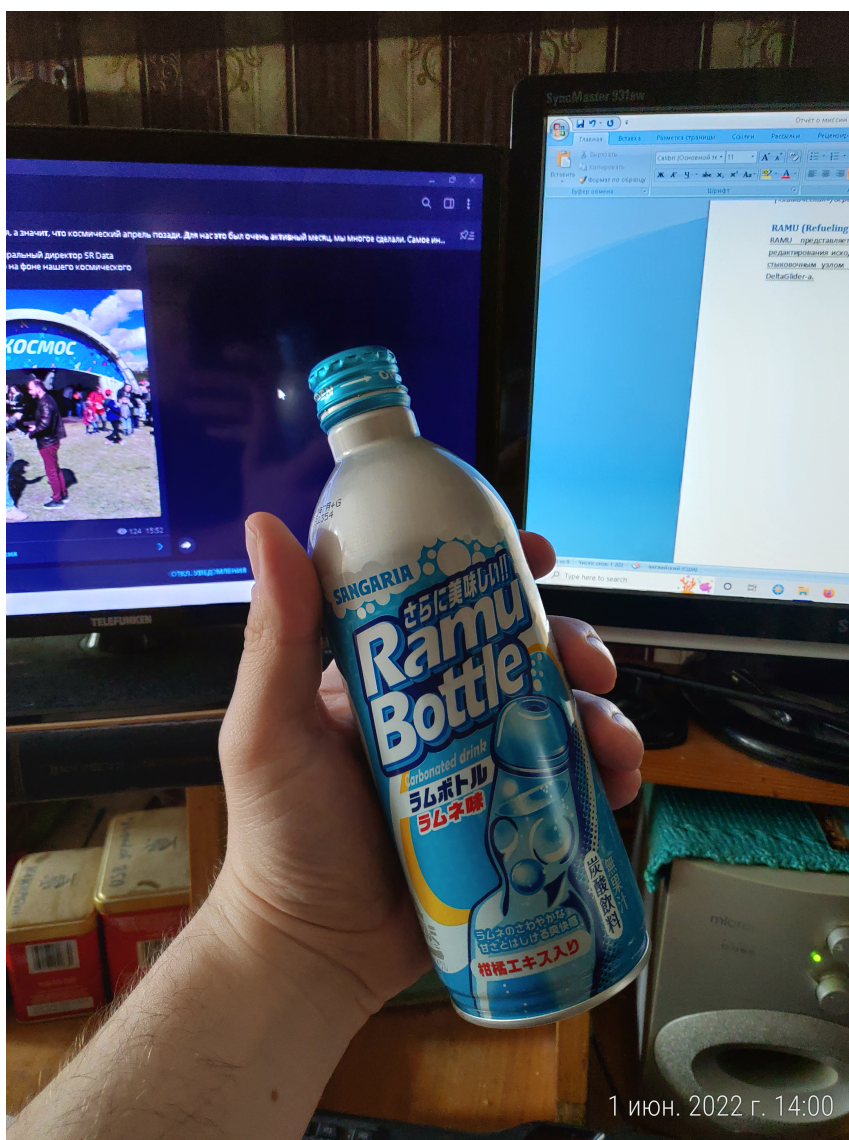


Рисунок 5 - Жестяная бутылка лимонада Ramu Bottle

Курьёзное созвучие названия лимонада с именем древнеиндийского мифического героя и земным названием упоминаемого в повести Артура Кларка инопланетного звездолёта, а так же необходимость двух стыковок с такими запращиками, определили название миссии.

Formosat-5

Formosat-5 (COSPAR/NSSDC 2017-049-A) – оптический спутник ДЗЗ, принадлежащий Национальной космической организации Тайваня, предназначенный для съёмки поверхности Земли в высоком разрешении. Запущен 24 августа 2017 года с базы Ванденберг, США, на PH Falcon.

Известно, что в сентябре 2017 года Formosat-5 испытал сбой, вызвавший ухудшение качества передаваемого им изображения и скорости передачи данных. На 2022 год сбой не устранён.

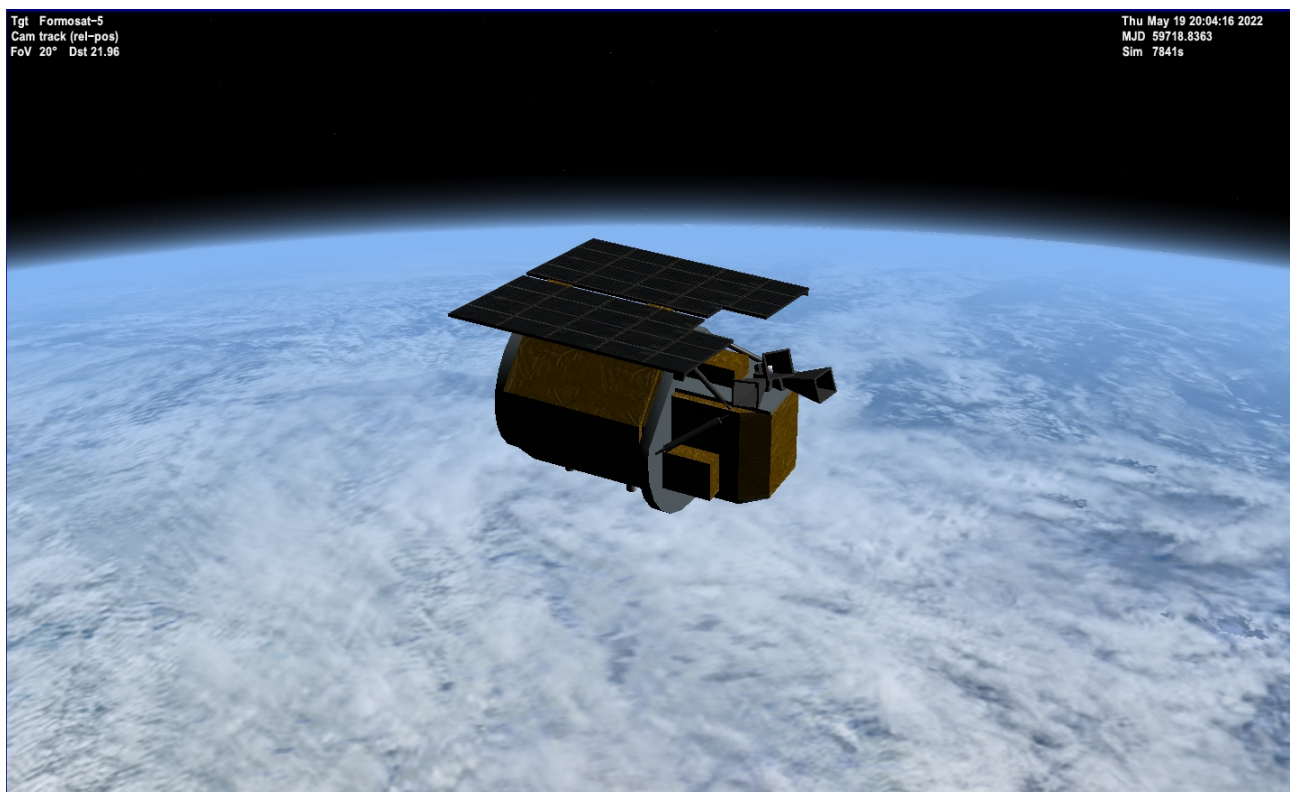


Рисунок 6 - виртуальная модель спутника д33 Formosat-5

Основные характеристики КА:

Высота периапсиса	716.6 км
Эксцентриситет	0.0009
Длина	2.8 м
Диаметр	1.6 м
Масса	450 кг

Задачей миссии решено было считать симуляцию ремонта Formosat-5 на орбите.

Моделируемая авионика

Виртуальное ПО КА в Orbiter 2016

Бортовой компьютер

Каждый КА в симуляторе по умолчанию оборудован компьютером с RTOS, способным обрабатывать различную пилотажно-навигационную информацию. В адд-онах базовый компьютер может быть отключён или заменён другими системами.

Исходные данные принимаются с виртуальных датчиков (IMU, радио- и барометрических высотомеров, ПВД, компасов, приёмников GPS и т.д.), реализованных в виде методов класса, описывающего виртуальную модель КА. Также возможен прямой доступ к ядру симулятора. Набор датчиков может быть различным в зависимости от КА.

Обработка информации ведётся в режиме hard real time, привязанном к реальному времени через частоту обновления графического движка симулятора.

ПО для бортового компьютера в Orbiter 2016 называется «модулями MFD». Модули пишутся на языках LUA и Visual C++ 6.0 (что несколько ограничивает их возможности в настоящее время).

После добавления в Orbiter, модули MFD доступны в бортовых компьютерах всех КА в работающей симуляции (т.е., нельзя сделать два разных набора модулей для двух разных КА - это архитектурное ограничение симулятора).

MFD

Информация выводится на MFD (Multi-Functional Display). Они способны отображать различную графическую и текстовую информацию, и оборудованы программируемыми кнопками управления.

Виртуальная камера

Каждый КА в симуляторе оборудован наружной виртуальной камерой, наподобие внешней камеры КА «Союз»:



Рисунок 7 - изображение с внешней камеры КА "Союз"

На картинку, идущую с камеры (так же как и в случае с реальным КА «Союз») накладывается слой с изображением полётной информации и слой с изображением интерфейса MFD.

Задержка, вызванная ограниченной скоростью распространения сигнала, а так же другие релятивистские эффекты, в Orbiter не эмулируется.

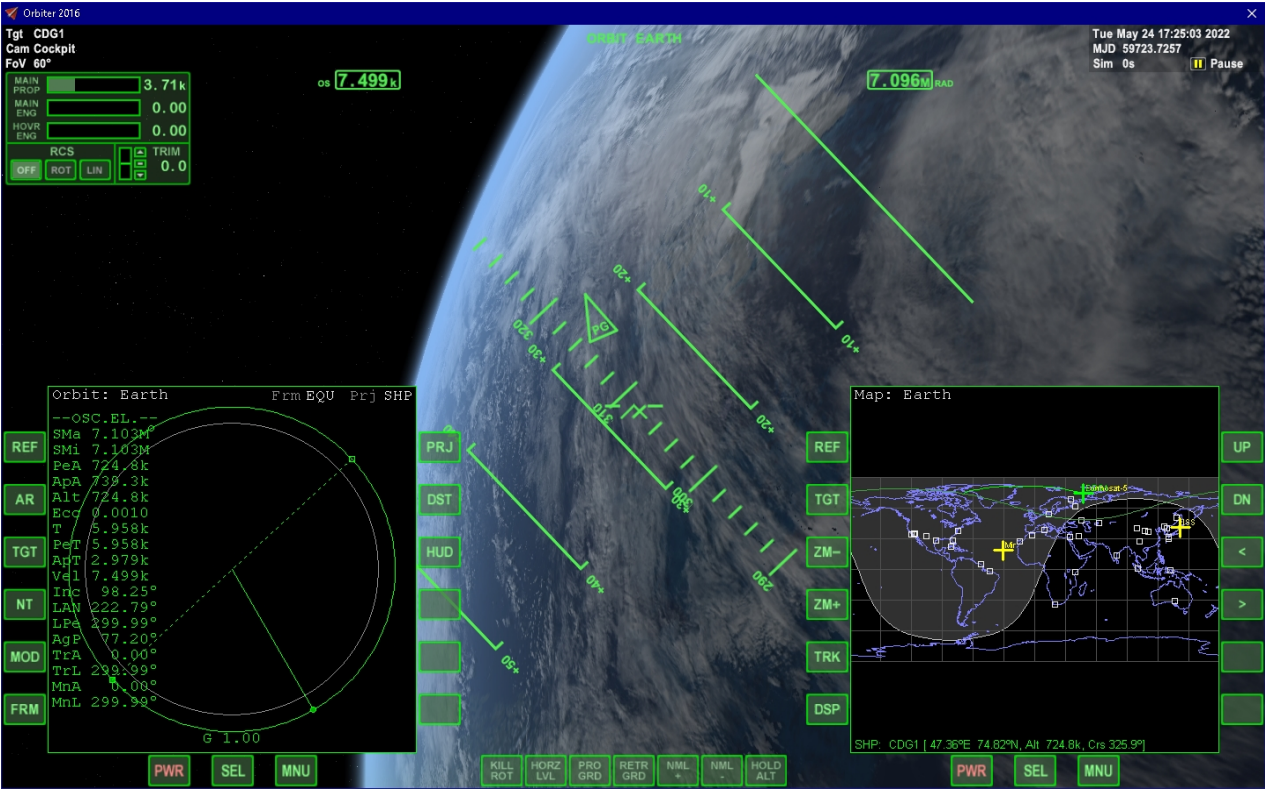


Рисунок 8 - скриншот интерфейса виртуальной камеры Orbiter 2016. Видна полётная информация и 2 экрана MFD.

Дополнительные модули MFD

На бортовые компьютеры моделируемых КА были установлены следующие дополнительные модули MFD:

Название	Язык разработки	Назначение	Автор
ReFuel MFD	Visual C++ 6.0	Управление насосами для перекачки топлива между состыкованными КА.	BrianJ
APLink	LUA 5.1.2	- Межпроцессная коммуникация с внешними (относительно Orbiter-a) приложениями - передача данных с датчиков КА - приём управляющих команд - трансляция принятых управляющих команд на органы управления.	Даниил Галахов

Внешние программы управления

Корректно работающих автопилотов для Orbiter 2016 не существует.

Поэтому, для выполнения задач миссии (в частности, для обеспечения точного вывода на орбиту с заданными параметрами, точных манёвров на орбите и точного reentry), пришлось создать внешние программы, позволяющие управлять КА в симуляторе. Так же потребовалось создать средство обмена данными между виртуальными КА и внешними программами.

Проблема устаревшего ПО

Orbiter создавался в 2000-2016 г.г., и потому модули MFD необходимо создавать при помощи устаревших средств разработки и компиляторов.

Поскольку мой ПК, помимо учебных задач, используется мной как рабочая станция для моего бизнеса (разработка приложений Unity), я посчитал неудобным устанавливать на него устаревшие версии Visual Studio и старые компиляторы C++ для того чтобы разработать MFD-автопилот.

Поэтому, было решено писать системы управления КА на C#, оставив на стороне Orbiter только относительно простую систему на LUA, отвечающую за обмен данными между виртуальным КА и внешними программами.

Среда передачи данных

Поскольку средства межпроцессной коммуникации C++ оказались недоступными, а LUA не обладает такими средствами, было решено передавать данные через файлы, хранимые на быстродействующем RAM-диске, созданном с помощью ImDisk Virtual Disk Driver.

Для этой цели на C# была написана библиотека, обеспечивающая передачу данных через файлы, по принципу многоуровневого сетевого протокола передачи информации с контролем ошибок (функционирующая по принципам, описанным в сетевой модели OSI/ISO). Для обеспечения устойчивой синхронизации был использован принцип маркерного доступа (как в протоколе Token Ring). Библиотека полностью покрыта unit-тестами, для обеспечения абсолютной надёжности работы на уровне софта.

Затем, система была транслирована на LUA 5.1.2, также с полным покрытием тестами.

APLink

Со стороны Orbiter библиотека обмена данными была включена в модуль MFD APLink. Его задачами являются:

- передача внешним приложениям данных с виртуальных датчиков виртуального КА
- приём команд управления от внешней программы-автопилота
- трансляция принятых команд управления на исполнительные механизмы КА.

OLink

Со стороны C# библиотека передачи информации была включена в библиотеку-wrapper OLink, которая обеспечивает единообразный и прозрачный обмен данными любых внешних приложений с виртуальным КА. Также, библиотека преобразует используемые в Orbiter единицы измерения, нормализует управляющие воздействия, и т.д.

При частоте обмена данными 10Hz (10 пакетов в секунду up- и downlink) система работает стабильно: отношение количества ошибочных пакетов к общему количеству переданных пакетов не превышает 0.00133, и все допущенные ошибки корректно обрабатываются, не вызывая сбоев в управлении КА.

Общая схема работы системы передачи данных представлена на рисунке ниже:

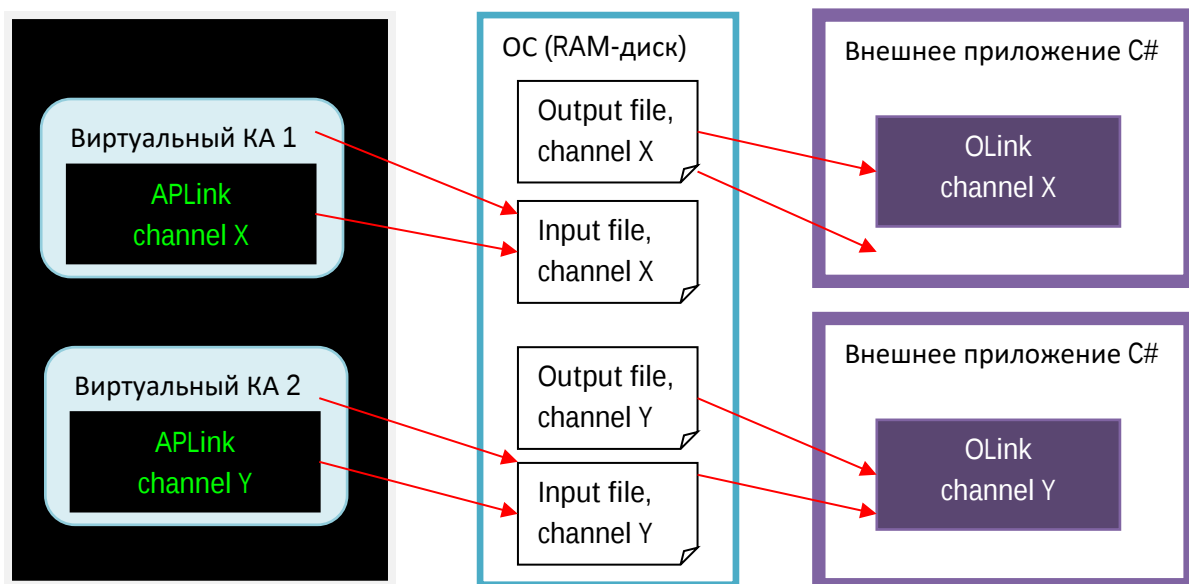


Рисунок 9 - схема системы обмена данными.

После создания системы обмена данными, были реализованы программы-автопилоты для управления виртуальными КА.

Launch autopilot.



Рисунок 10 - GUI Launch autopilot

Launch autopilot (пусковой автопилот) обеспечивает управление РН в процессе выведения КА на орбиту.

Автопилот требует указания широты космодрома, азимута и времени запуска РН. После запуска он обеспечивает стабильное и устойчивое выведение КА на орбиту с заданным апоаписом, наклонением и эксцентриситетом.

После завершения работы автопилота девиация по наклонению составляет не более $0,1^\circ$, по высоте – не более 500 м, по эксцентриситету – менее 0,001.

Управляющие воздействия выдаются PID-регуляторами на основе команд от конечного автомата, описывающего режимы полёта. Параметры PID-регуляторов вычисляются в отдельном приложении по методу Коэна-Куна, или вручную, по методу Циглера-Николса.

Для РН семейства Titan удалось вычислить значения параметров PID, отличающиеся от реальных менее чем на 0,01 (с учётом коэффициента преобразования для Orbiter).

SOMA (Simple Orbital Maneuver Autopilot)

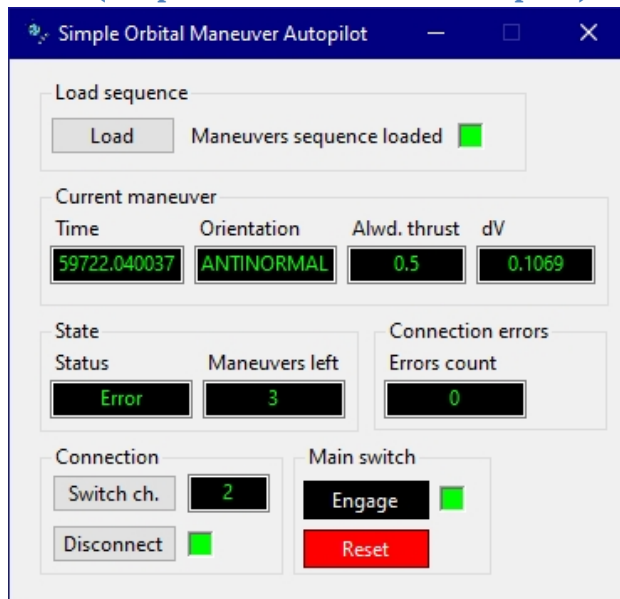


Рисунок 11 - GUI SOMA

Simple Orbital Maneuver Autopilot (автопилот для простых орбитальных манёвров) обеспечивает простые манёвры КА на орбите.

Данный автопилот работает по программе, задаваемой в форме текстового файла, выполняя последовательность указанных там манёвров.

Автопилот вычисляет время и продолжительность включения маршевой Д.У., исходя из времени, заданного максимального уровня тяги, требуемого dV и текущей массы КА, задает ориентацию КА для манёвра, включает и выключает маршевую Д.У. в вычисленное время.

Данный автопилот не способен ориентировать КА в произвольном направлении и самостоятельно рассчитывать орбитальные манёвры. Их следует рассчитывать вручную или в соответствующем астродинамическом софте.

Reentry autopilot



Рисунок 12 - GUI Reentry autopilot

Reentry autopilot (автопилот reentry, входа в атмосферу) обеспечивает вход в атмосферу и аэродинамическое торможение КА типа DeltaGlider, на высотах от 120 до 40 км и скоростях от 7.8 км/с до 5.0 км/с.

Автопилот загружает профиль полёта из файла (должен генерироваться внешним расчётным ПО, в данном случае для этого была реализована специальная программа на Python, рассчитывающая профиль reentry). Далее устанавливается начальный угол атаки (это позволяет изменять «остроту реакции» автопилота, в зависимости от массы DeltaGlider-a).

В процессе полёта автопилот выдерживает задаваемые в профиле полёта углы атаки. Управляющие команды также генерируются с помощью PID-регуляторов.

Сейчас (после завершения миссии) понятно, что применённый метод расчёта профиля reentry неправильный. Reentry autopilot требует серьёзной переработки.

ПО для астродинамических расчётов

Для астродинамических расчётов и расчёта reentry использовалась самодельная библиотека для астродинамических расчётов, реализованная на Python на основе алгоритмов из книги D.Vallado «Fundamentals of Astrodynamics and Applications. 4-th edition» и другой литературы по астродинамике, математике и родственным дисциплинам (R. Battin, A.A.Суханов, J.C.Adams Jr., G. and T. Korn, и др.).

Для проверки результатов работы астродинамической библиотеки и проверки правильности симуляции использовался GMAT R2020A.

Для конвертирования TLE в state vectors использовалась библиотека sgp4 на Python.

Особые условия

Для достижения практических целей моделирования миссии было решено:

1. Смоделировать все операции, необходимые для выполнения миссии, включая:
 - a. планирование совместного полёта нескольких аппаратов
 - b. планирование орбитальных манёвров
 - c. вывод КА на заданную орбиту с точным соблюдением параметров
 - d. уточнение местоположения КА по данным НИПов
 - e. перехват КА в автоматическом режиме
 - f. стыковку
 - g. дозаправку
 - h. ВКД
 - i. ремонтные работы с «железом» и софтом спутника в режиме ВКД
 - j. сход с орбиты
 - k. reentry
 - l. посадку на ВПП на выбранном космодроме
2. Изменить параметры Д.У. стандартного DeltaGlider чтобы обеспечить реалистичные возможности пилотируемого КА.
3. Рассчитывать орбитальные манёвры с помощью астродинамической библиотеки на Python и GMAT. Стандартные модули MFD Align Plane и Sync Orbit, предназначенные для вычисления манёвров изменения наклона и перехвата цели, использовать только в качестве резервных.
4. Использовать save/load только для:
 - a. обеспечения защиты виртуальной миссии от не зависящих от человеческого фактора проблем (сбоев электропитания, «вылетов» симулятора, и т.п.)
 - b. приостановки выполнения миссии во время отдыха или в рабочее время (планируемая общая продолжительность полёта должна была составлять от 3 до 8 суток).
5. В случае допущения нами критической ошибки, которую невозможно было бы исправить силами и средствами частной космической компании, организующей данную экспедицию, миссию решено было считать проваленной.

6. В случае посадки в нерасчётном районе в дикой местности, решено было пешком пройти расстояние, равное расстоянию от фактической точки посадки до ближайшего к ней аэродрома.

Подготовка к миссии

В процессе подготовки к миссии были решены следующие задачи:

1. Изучены особенности совместимости Orbiter 2016 с различными адд-онами.
2. Изучен и исправлен исходный код нескольких адд-онов для обеспечения их работоспособности.
3. Разработан скрипт на языке LUA, обеспечивающий установку реалистичных параметров Д.У. для DeltaGlider.
4. Собрана и систематизирована информация по РН семейства Titan
5. Изучены и полностью переработаны файлы конфигурации адд-она РН Titan, для обеспечения работоспособности и реальных характеристик указанных РН.
6. Собрана и систематизирована информация о КА Formosat-5.
7. Разработан Launch autopilot, в т.ч.,
 - a. Разработана начальная версия автопилота в форме скрипта на LUA, исполнявшегося в терминале виртуального бортового компьютера виртуального КА.
 - b. Разработана улучшенная версия автопилота в форме модуля MFD на LUA (корректного функционирования добиться не удалось).
 - c. Разработана система обмена данными между Orbiter и внешним ПО (модуль MFD APLink и библиотека OLink)
 - d. На C# Реализована имплементация Launch autopilot.
 - i. Изучены особенности методов вывода КА на орбиту со строго заданными параметрами.
 - ii. Изучена математическая модель расчёта траекторий движения КА в Orbiter в различных режимах работы симулятора.
 - iii. Выбраны корректные настройки симулятора для наиболее точного расчёта траекторий. Обеспечена погрешность позиционирования КА (относительно реальных данных, взятых из TLE с сайта [Heavens Above](#)) менее 1 км.
 - iv. Разработана вложенная библиотека, преобразующая параметры орбиты в стандарте Orbiter (очевидно, рассчитанные относительно инвариантной плоскости Солнечной Системы) в параметры орбиты, рассчитанные в экваториальном фрейме (GCRF).
 - e. Разработана программа для вычисления параметров PID-регуляторов, проведены расчёты для определения корректных параметров PID-регуляторов всех каналов управления для используемых РН и КА.
 - f. Проведены виртуальные тестовые пуски различных РН семейства Titan с различными ПН, для тестирования и отладки автопилота.
 - g. Изучены особенности вывода КА на орбиты с наклоном более 90° с высокоширотных космодромов.

8. На языке Python разработаны алгоритмы для вычисления манёвра схода с орбиты и профиля reentry.
9. Разработан и имплементирован (с помощью C# и библиотеки OLink) Reentry autopilot.
 - a. Проведены тестовые виртуальные reentry КА DeltaGlider с орбит высотой 220-320 км.
10. Проведён виртуальный тестовый орбитальный полёт КА DeltaGlider, выведенного РН Titan IVA. *(Неудачный. У DeltaGlider, работавшего как верхняя ступень РН, не хватило запаса dV чтобы выйти на стабильную орбиту, и мы соскользнули на суборбитальную траекторию. Благодаря малому углу входа в атмосферу, удалось сохранить скорость, и, выполнив $\frac{3}{4}$ витка, мы экстренно сели в Peenemünde, выкатившись на песчаный пляж за пределами полосы. К счастью, прочные стойки и шасси DeltaGlider выдержали).*
11. Написана вспомогательная программа на Python, позволяющая рассчитывать запас dV собранной РН.
12. Собрана и проанализирована информация об альтернативных, не реализованных на практике конфигурациях РН Titan, предназначенных для вывода более тяжёлых ПН.
13. Создана виртуальная модель РН Titan IVC, предназначенная специально для вывода DeltaGlider.
14. Разработан и имплементирован (с помощью C# и библиотеки OLink) SOMA.
 - a. Выполнены виртуальные тестовые последовательности манёвров.
15. Разработан порядок выполнения манёвров для достижения целей миссии.
16. С помощью астродинамической библиотеки на Python рассчитаны запланированные орбитальные манёвры.
17. С помощью GMAT проверены результаты расчётов, выстроена полётная миссия.
18. Изучены различия в алгоритмах расчётов траекторий в GMAT и Orbiter 2016
 - a. Определены причины возникновения ошибок интегрирования на стороне Orbiter на длинных промежутках времени.
 - b. Определены настройки Orbiter, необходимые для точного вычисления траекторий.
19. Вычислены параметры окон запуска.
20. Подготовлены сценарии для Orbiter, моделирующие запуск RAMU и DeltaGlider.
21. Анной Апальковой подготовлена эмблема миссии.
22. На основе OrangePi PC2, USB-адаптера WiFi и Web-камеры создан упрощённый эмулятор БЦВМ Formosat-5 (на ОС Armbian; в качестве имитатора приложения для обработки изображений использовался сервер видеонаблюдения ZoneMinder).

Полёт

Полётная задача

Тайваньский спутник Formosat-5 с сентября 2017 года работает в нештатном режиме, из-за проблем с ПО научной ПН. Требуется перехватить его, сблизиться на минимальное безопасное расстояние, физически подключиться к сервисным разъёмам его БЦВМ, и устранить неполадки, обновив прошивку ПО.

Решение

Поскольку требуется непосредственное подключение к ЭВМ спутника, необходима ВКД. Значит, требуется пилотируемый полёт, и необходимо использовать DeltaGlider.

Поскольку Formosat-5 вращается на относительно высокой орбите, выйти на неё сразу не удастся, потребуется переход с НОО на орбиту спутника. Для этого у доступного нам DeltaGlider не хватит запаса dV . Также, для обеспечения безопасности миссии требуется иметь на борту DeltaGlider «неприкосновенный» запас топлива, предназначенный для схода с орбиты и атмосферных манёвров на финальном участке спуска, над аэродромом посадки. Значит, необходимо дозаправить КА на орбите.

Перехват должен осуществляться КА-заправщиком в автоматическом режиме, с помощью автопилота SOMA. На это уйдёт определённое количество топлива из баков заправщика, также следует предусмотреть небольшой остаток топлива для сведения заправщика с орбиты, чтобы предотвратить засорение НОО. Запаса топлива в баках одного заправщика, по расчётам, будет недостаточно, следовательно, потребуется два заправщика.

Также, для экономии топлива требуется обеспечить точное выведение всех КА на заданные орбиты.

Далее, после дозаправки, DeltaGlider сможет перехватить Formosat-5, экипаж выполнит ремонтные работы на спутнике, и вернётся на один из космодромов, оборудованных полосой, способной принять DeltaGlider.

План

- 1) В расчётное время запустить КА RAMU1, RAMU2 и DeltaGlider на орбиту с наклоном 98.24° и высотой апоаписа 300.0 км.
- 2) В автоматическом режиме осуществить перехват DeltaGlider аппаратом RAMU1.
- 3) Выполнить стыковку DeltaGlider с RAMU1.
- 4) Выполнить дозаправку DeltaGlider.
- 5) Отстыковать RAMU1, в автоматическом режиме выполнить сведение его с орбиты.
- 6) В автоматическом режиме осуществить перехват DeltaGlider аппаратом RAMU2.
- 7) Выполнить стыковку DeltaGlider с RAMU2.
- 8) Выполнить дозаправку DeltaGlider.
- 9) Отстыковать RAMU2, в автоматическом режиме выполнить сведение его с орбиты.
- 10) Выполнить перехват КА Formosat-5 аппаратом DeltaGlider.
- 11) Сблизиться с Formosat-5 на расстояние 50 м.
- 12) Выполнить ВКД:
 - a. С помощью MMU сблизиться с Formosat-5 и закрепиться на нём.
 - b. Подключиться к ЭВМ спутника.
 - c. Обновить прошивку ЭВМ.
 - d. Установить софт для захвата и обработки изображения
 - e. С помощью MMU вернуться к DeltaGlider.
- 13) Перевести DeltaGlider на НОО высотой 300 км.
- 14) Выполнить reentry.
- 15) Выполнить посадку DeltaGlider на выбранном аэродроме.

Исполнение

Инициализация.

RAMU1, RAMU2 и DeltaGlider были выведены на заданные орбиты в расчётное время, 20, 21 и 22 мая 2022 года, в штатном режиме.

В процессе вывода на орбиту DeltaGlider, отделение второй ступени РН произошло недостаточно плавно, из-за чего КА резко отклонился вправо, но, прежде чем мы как-то отреагировали на ситуацию, КА самостоятельно вернулся на расчётный курс.

Интервал между каждым из запусков составил приблизительно 1 сидерические сутки. Отклонение от требуемых параметров орбиты составило:

по наклонению:	+/- 0.1°
по высоте:	+15.0 км / -5.0 км
по RAAN:	+/- 0.02°

После вывода на орбиту, запасы топлива КА составляли:

КА	Запас топлива маршевой Д.У.	Запас топлива RCS
RAMU1	7 т	0.75 т
RAMU2	7 т	0.75 т
DeltaGlider	3.02 т	0.35 т

Фаза 1.

23 мая 2022 года в 00:00:00 была инициирована последовательность перехвата DeltaGlider аппаратом RAMU1. КА, под управлением SOMA, выполнил 4 манёвра (выравнивание плоскости орбиты по RAAN, по INC, и двухимпульсный перехват КА-цели за 8 витков, что обеспечивало минимальный расход dV).

Заправщик перехватил DeltaGlider 23 мая в 15:09:41 над Карским морем, у северной оконечности Новой Земли, выйдя из манёвра в 64-х километрах от аппарата-цели. Несмотря на относительно большое расстояние, скорости аппаратов были почти равны (разница скоростей составляла 7-9 м/с), поэтому мной было принято решение идти на сближение и стыковку.

В течение 40 минут DeltaGlider сблизился с RAMU1, и, под моим управлением, со второй попытки выполнил в ручном режиме стыковку с заправщиком. Расход топлива на сближение оказался неожиданно велик, и к моменту стыковки было израсходовано почти 2 т топлива из баков маршевой Д.У., а в баках RCS DeltaGlider осталось всего 3 кг топлива.

В течение часа из баков RAMU1 было принято более 5 т топлива, так что после первой дозаправки запас топлива в баках DeltaGlider составил 5300 кг в баках маршевой Д.У. и 600 кг (полная заправка) в баках RCS.

Далее, RAMU1 был отстыкован от DeltaGlider и в автоматическом режиме сведён с орбиты.

Процесс выполнения орбитальных манёвров оказался исключительно длительным и практически не требовавшим нашего внимания – автопилот выполнял все манёвры с высокой точностью, обеспечивая почти абсолютное соответствие между траекторией, вычисленной в GMAT и траекторией, моделируемой в Orbiter. Поскольку средства ускорения прокрутки времени в симуляторе вызывали ошибки интегрирования векторов состояния, а сидение в течение многих часов возле компьютера не представлялось рациональным, было решено длительные последовательности манёвров просчитывать только в GMAT, и затем просто переносить в Orbiter вектора состояния маневрирующих КА.

Фаза 2.

После этого была начата вторая фаза миссии. Теперь на перехват DeltaGlider пошёл RAMU2. В расчёт его траектории я ввёл дополнительное условие, повышающее точность перехвата ($RMAG_{RAMU2} = RMAG_{DeltaGlider}$, Tolerance=5.0 км) Выполнив последовательность манёвров, аналогичную манёврам RAMU1, RAMU2 перехватил DeltaGlider над северной частью острова Врангеля, почти на линии терминатора, 24 мая в 09:28:34. После выполнения манёвра расстояние между аппаратами составило 19.3 км, а разница скоростей составила приблизительно 4 м/с.

Вновь принято было решение сближаться и идти на стыковку. Однако это оказалось сложнее чем ожидалось! DeltaGlider, став почти на 2.5 т. тяжелее, чем был перед стыковкой с RAMU1, расходовал топливо из баков RCS значительно быстрее. В результате, несмотря на запас в 600 кг., топливо в баках RCS закончилось, когда DeltaGlider находился примерно в 120 м. от RAMU2, имея относительную скорость с заправщиком около 0.2 м/с.

Ситуация была близка к критической, однако решение было найдено довольно быстро. ReFuel MFD позволял гибко управлять перекачкой топлива между баками КА, а двигатели RCS использовали ту же топливную пару, что маршевая Д.У. Быстро настроив перекачку из баков маршевой Д.У. в баки RCS, удалось закачать в баки RCS около 300 кг, топлива, что позволило завершить стыковку. Процесс стыковки занял около часа.

В течение следующих полутора часов в баки DeltaGlider было заправлено ещё около 6 т топлива, так, что общий запас топлива в баках маршевой Д.У. составил 9.4 т, а в баках RCS – 0.6 т. После завершения дозаправки, RAMU2 был отстыкован от DeltaGlider и в автоматическом режиме сведён с орбиты.

Фаза 3.

После дозаправки была начата третья фаза миссии. Сначала был оценён полный запас топлива. Его оказалось несколько меньше, чем планировалось. В случае, если бы мы попытались выполнить перехват по схеме из шести манёвров (трансфер на орбиту Formosat-5, выравнивание плоскости орбиты и двухимпульсный манёвр перехвата), топлива на дальнейшее возвращение на низкую орбиту и reentry хватало в обрез, без малейшего запаса. Поэтому решено было использовать более эффективную схему перехвата, путём двухимпульсного выравнивания плоскости и трёхимпульсного манёвра перехвата (по смыслу напоминающего биэллиптический манёвр). Такая схема сэкономила около 500 кг топлива, и, более того, баллистические условия позволяли выполнить её в один виток, сильно сокращая время миссии. Воспользовавшись данной схемой, мы стартовали на перехват

спутника, и 24 мая в 15:45:50 перехватили Formosat-5, сблизившись с ним на расстояние 600 м, с относительной скоростью 2 м/с.

Медленное, осторожное сближение со спутником заняло более часа. В 17:25 DeltaGlider, под моим управлением, сблизился с Formosat-ом на расстояние менее 50 м, и Анна, готовившаяся к ВКД, вышла через шлюз в открытый космос.

Сблизиться со спутником оказалось несложно, но далее начались сложности. Подключиться к операционной системе эмулятора БЦВМ Formosat-5 удалось только через полчаса возни с мелкой электроникой в толстых перчатках и бесконечных перезагрузок. Радиосвязь, эмулируемая через LinPhone и SIP-коммутатор, то и дело прерывалась сильными помехами. Выполнить перепрошивку и начальную настройку удалось без проблем, однако настроить приложение, имитирующее ПО для обработки изображений, никак не удавалось. Сначала Анна действовала по предварительно разработанной инструкции. Потом, убедившись что инструкция не помогает, она консультировалась со мной, пытаясь определить правильность пунктов инструкции. Затем, полагаясь на мой опыт системного администратора, пыталась выполнять мои инструкции. Я всё это время контролировал относительное положение КА, удерживая DeltaGlider на фиксированной дистанции от Formosat-5.

Мы пытались наладить ПО виртуального спутника в течение почти 4-х часов (вместо запланированных двух), однако все попытки оказались безуспешны. В результате я принял решение демонтировать модуль памяти эмулятора спутника, отформатировать его, накатить чистую прошивку, и наладить систему удалённого управления, чтобы обеспечить специалистам заказчика возможность самостоятельно сделать то, что не удалось сделать нам. Анна, демонтировав карту памяти эмулятора, вернулась на борт, мы установили свежую прошивку, и затем Анна, вернувшись к спутнику, установила карту памяти на место. После перезапуска и инициализации, спутник заработал. Проверка показала, что устойчивая связь с его БЦВМ есть. В итоге, работы по ремонту, пуску и наладке Formosat-5 заняли почти 5 часов.



Рисунок 13 - Formosat-5 после завершения ремонта

Фаза 4.

Закрыв люки, мы приступили к выполнению 4-й фазы миссии. Израсходовав приблизительно 780 кг топлива, 24 мая в 23:04:00 мы спустились на парковочную орбиту с высотой апоаписа около 373 км и эксцентриситетом 0,0023. Исходя из параметров этой орбиты, я запустил приложение для расчёта reentry, указав ему в качестве аэродрома-цели Kennedy Space Center. Однако, приложение не нашло прохождений КА над указанной точкой в ближайшие трое суток. Тогда, я изменил цель на аэродром авиабазы Эдвардс, штат Калифорния, и приложение быстро определило, что для посадки там нам нужно выдать тормозной импульс 26 мая, в 04:43:59. Предполагая, что проверка расчёта не требуется, мы завершили расчёт, и отправились отдыхать.

На следующий день, мы заранее настроили автопилот, и, в заданное время, начали reentry.

После тормозного импульса был активирован Reentry autopilot и установлен угол атаки $0,7^\circ$. В расчётное время, на высоте 120 км, DeltaGlider нырнул в атмосферу. Понемногу начала расти перегрузка.

Однако здесь что-то пошло не так. Вероятно, я задал некорректную массу корабля при расчётах, указав несколько большее значение, чем реальное бывшее при входе в атмосферу. DeltaGlider тормозился несколько быстрее чем я ожидал. Разница в скорости была несколько десятков метров в секунду - горизонтальная скорость несколько ниже расчётной, а вертикальная несколько выше. Погружение в атмосферу шло быстрее, и КА нырнул гораздо глубже, чем я ожидал – минимальная высота при первом входе в атмосферу составила около 47 км вместо ожидаемых 60. Я не придавал этому значения, тем более что инерция и подъёмная сила быстро выбросили КА вверх. Однако при повторном входе в атмосферу скорость ещё более уменьшилась относительно расчётной. Первый вход в атмосферу был над Средним Западом США, второй – над Аляской, третий – над севером Восточной Сибири, и в этот момент стало понятно, что скорости отчётливо не хватает. Я отключил автопилот, взяв управление на себя и всеми силами стараясь минимизировать потерю скорости. DeltaGlider продолжал мчаться по границе атмосферы на гиперзвуковой скорости, мы миновали Монголию и Гималаи. Над южной частью Индии скорость упала до 5 км/с, КА тормозился значительно быстрее расчётного. Стало понятно, что до авиабазы Эдвардс мы не дотянем, а впереди был только Индийский океан. Посадка на воду где-то в Индийском океане означала провал всей миссии.

Траектория орбиты шла дальше, через Антарктиду и западную часть Южной Америки, однако у меня были сомнения в том, что DeltaGlider сможет долететь до американского континента. Единственной сушей впереди была Антарктида, однако, на Map MFD в Orbiter там не было обозначено полярных станций.

По моей команде, Анна стала искать информацию о полярных станциях в локальной копии Википедии. Однако российские полярные станции Мирный и Молодёжная находились слишком далеко от нашей траектории. Скорость продолжала падать. На Map MFD были указаны отметки аэронавигационных маяков на острове Короля Георга. Где-то в том районе, насколько я знал, находились чилийские антарктические станции, но до них тоже было далеко...

Я двинул РУД вперёд, стараясь энергией остатка топлива изменить ситуацию в нашу пользу, однако это было бесполезно. Всё что мне удалось – лишь слегка увеличить высоту, израсходовав около 200 кг

топлива. Расходовать оставшееся сейчас было бессмысленно – выгоднее было поберечь его до момента посадки, чтобы иметь возможность скомпенсировать потерю скорости или перетянуть через препятствие.

Мы нырнули на ночную сторону над побережьем Антарктиды. Высота полёта упала до 50 км, скорость – до 3.8 км/с. Вокруг был непроглядный мрак. Я продолжал напряженно бороться с тормозящей наш полёт атмосферой, пытаюсь удержать в воздухе неуклонно замедляющийся DeltaGlider, Анна, перегнувшись через моё плечо, меняла фокусировку Map MFD, пытаюсь найти хоть какую-то отметку полярной станции, возле которой мы могли бы сесть. Однако на карте не было ни одной отметки, кроме отметки Южного Полюса.

Внезапно, я догадался, что это может быть нашим спасением. Из научно-популярных книг и сериалов Discovery про исследования Антарктиды, я знал, что точно на Южном Полюсе расположена американская полярная станция, оборудованная отличным ледовым аэродромом с очень длинной полосой, способной принимать сверхтяжёлые грузовые самолёты, типа Galaxy C-5, Ан-124 или Ил-76. Если такой полосы хватает для трёхсоттонной Galaxy, то её хватит и для DeltaGlider-а! Я приказал Анне срочно найти информацию об этом месте, и это оказалось правдой – точно на географическом Южном Полюсе находится полярная станция Амундсен-Скотт, а рядом, менее чем в километре – аэродром Jack. F. Paulus Skiway.

Быстро выяснились и другие параметры. Высота аэродрома составляла 2835 м, направление захода – 02/20, т.е., 20° или 200°. До Южного полюса оставалось около пятисот миль, он был слева от нас. Я положил DeltaGlider в крутой левый крен, заставляя его траекторию искривляться в направлении базы Амундсен-Скотт. Скорость упала ещё на 1 км/с, до 2.8 км/с, однако траектория медленно развернулась в сторону Южного Полюса. До базы оставалось 400 миль, потом 300, 200... Скорость падала, высота уменьшилась до 30 км, затем до 20...

Симулятор не предоставлял радионавигационного маяка на базе Амундсен-Скотт, поэтому я включил индикацию дополнительных маркеров на экране. Однако это было излишним, ночная тьма сработала нам на руку – возле лиловой метки, обозначающей Южный Полюс, была видна небольшая горсточка желтоватых огней.

DeltaGlider прошёл над полюсом на высоте около 5 км со скоростью около 420 м/с. Я включил посадочные фары и заложил вираж, пытаюсь развернуться на курс 200, не потерять из виду огоньки базы и определить местоположение ВПП. Посадочной полосы видно не было – область Южного Полюса в Orbiter 2016 не детализирована, поэтому всё, что удалось различить – это идущую в приблизительно правильном направлении мутную желтоватую линию из неровных пикселей, несколько отличающуюся по цвету от окружающего голубовато-белого фона. Накренив машину вправо, я направил DeltaGlider туда.

Речевой информатор начал отсчитывать высоту. Я выровнял аппарат, заводя его на полосу. На барометрической высоте около 3000 м, примерно в двухстах метрах от поверхности, DeltaGlider внезапно попал в мощный порыв ветра, скорость просела, машину накренило влево. Я толкнул вперёд РУД, компенсируя джойстиком крен, глайдер повело вправо, но скорость резко увеличилась, не давая машине «провалиться». Я вновь выровнял машину над поверхностью, осторожно «притирая» её к полосе. 20 метров, 10... Я убрал газ, аппарат довольно сильно тряхнуло, он слегка

клюнул носом и заскользил по снежной полосе – я, как новичок, забыл выпустить шасси. DeltaGlider проскользил по снегу метров 300, затем, очевидно, задев сугроб правым крылом, рыскнул вправо, развернулся влево, проскользил ещё какое-то расстояние хвостом вперёд и, слегка накренившись вправо, остановился. Над неровно освещённой левым посадочным прожектором полосой расплывалось бурое облако гептильного дыма, мешающееся с паром от испарённого реактивной струёй снега.

Результаты моделирования

Миссия провалена.

- Основные задачи выполнены частично.
 - Ремонтные работы на спутнике Formosat-5 выполнены не в запланированном объёме.
 - Посадка произведена в нерасчётном районе.
- Эвакуация персонала и оборудования (12.5 тонн негабарита, с Южного Полюса, в условиях полярной ночи!) затруднена.
- Ценные для науки льды вокруг полярной станции Амундсен-Скотт переизгажены гептилом.
- Снежная ВПП аэродрома Jack. F. Paulus Skiway проплавлена реактивным выхлопом на протяжении порядка 400 м.
- Экипаж и матчасть не пострадали. (Посадка на брюхо, по чистой случайности, оказалась наилучшим решением. ВПП на базе Амундсен-Скотт сделана из утрамбованного снега, посадка на колёсном шасси, тем более зимой, могла окончиться сломанными стойками и капотированием. А так пиковая перегрузка при посадке не превысила 3G, повреждений DeltaGlider Orbiter не зарегистрировал.)

Практические результаты

В процессе выполнения моделирования миссии я:

1. Углубил свои знания астродинамики и навыки практического применения, в частности:
 - a. Верифицировал результаты работы алгоритмов из самодельной астродинамической библиотеки на Python.
 - b. Разработал алгоритмы для вычисления параметров манёвров схода с орбиты и перехода к reentry.
2. Изучил некоторые аспекты динамики РН на компьютерных моделях.
3. Разработал ПО, способное управлять компьютерной моделью РН на участке вывода КА на орбиту.
4. Изучил некоторые аспекты динамики атмосферного reentry.
5. Разработал алгоритмы для вычисления различных параметров reentry.
6. Разработал ПО, способное управлять компьютерной моделью КА на участке reentry.
7. Обновил и углубил свои знания ТАУ и САУ.
8. Расширил и отрефакторил прикладные библиотеки на C#, реализующие различные фильтры и PID-регуляторы.
9. Изучил ЯП LUA.
10. Улучшил свои навыки использования GMAT.

11. Обновил свои знания в области принципов, технологий и протоколов сетей обмена данными.
12. Улучшил практические навыки в области soft skills, в частности:
 - a. Улучшил стрессоустойчивость.
 - b. Улучшил свои навыки планирования.
 - c. Улучшил навыки тайм-менеджмента и распределения внимания в стрессовых ситуациях.
 - d. Изучил основы Crew Resource Management.
 - e. Улучшил свои навыки в области оперативного руководства персоналом.

В процессе моделирования миссии Аня Апалькова:

1. Углубила свои знания о космонавтике и аэрокосмической технике.
2. Изучила основы механики космического полёта.
3. Изучила базовую теорию ВКД.
4. Углубила свои знания в области прикладных компьютерных систем (Linux) и компьютерных сетей.
5. Закрепила практические навыки дизайна логотипов и фирменного стиля.
6. Улучшила навыки работы с информацией и технической документацией.
7. Улучшила практические навыки в области soft skills, в частности:
 - a. Улучшила стрессоустойчивость.
 - b. Получила опыт работы в ситуациях с неполной исходной информацией.

(Дополнительные подробности она расскажет сама, если захочет).

Дополнительная информация

Подготовка миссии выполнялась с 22.03.2022 по 18.05.2022, вечерами, в свободное от работы время.

Виртуальная миссия была выполнена вечерами 20-22.05.2022 и 25-26.05.2022.

Кот Василий, сопровождавший нас в виртуальном циркумполярном перелёте, и оказывавший посильную моральную поддержку в процессе подготовки миссии, в симуляции миссии не участвовал, предпочтя поддерживать нас удалённо с облюбованной им тумбочки в прихожей.