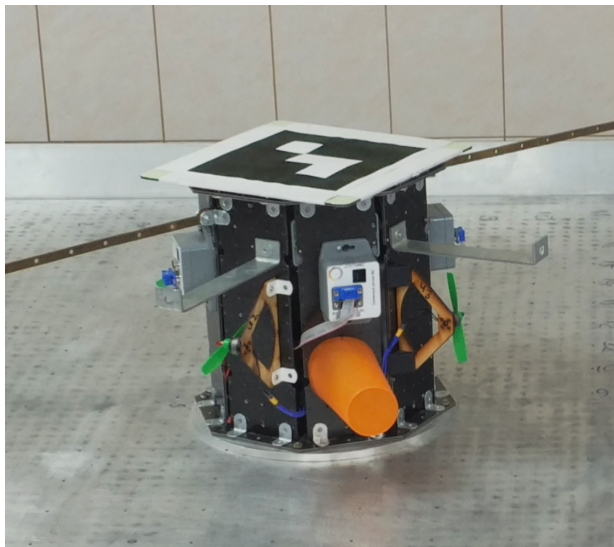


Шачков М.О., Коптев М.Д.,
Иванов Д.С., Овчинников М.Ю.

Моделирование движения микроспутников на стенде с
аэродинамическим столом.

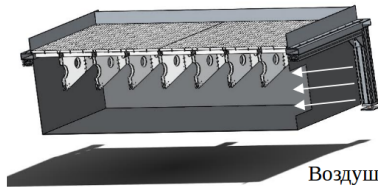
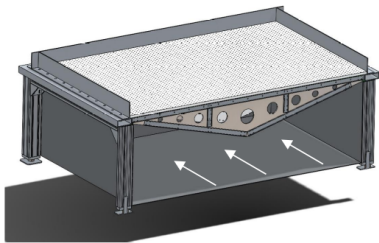


- Макет - модель малого КА (микроспутник)
- Бортовой компьютер - Raspberry Pi 2
- Бортовые сенсоры: инерциальные блоки - акселерометр и ДУС, магнетометр, оптические сенсоры имитирующие звёздный датчик, видеокамера
- Управляющие механизмы: маховик и 4 бесколлекторных двигателя с пропеллерами
- Связь с макетом осуществляется через WiFi соединение



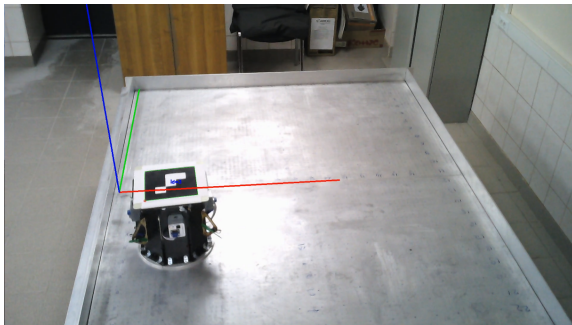
- Практическое изучение математических моделей движения твёрдого тела
- Структурная и параметрическая идентификация математических моделей
- Постановка и решение задач управления, синтез управляющих законов
- Построение и реализация наблюдателей для оценки вектора состояния системы
- Непосредственная реализация управления в виде ПО
- ...

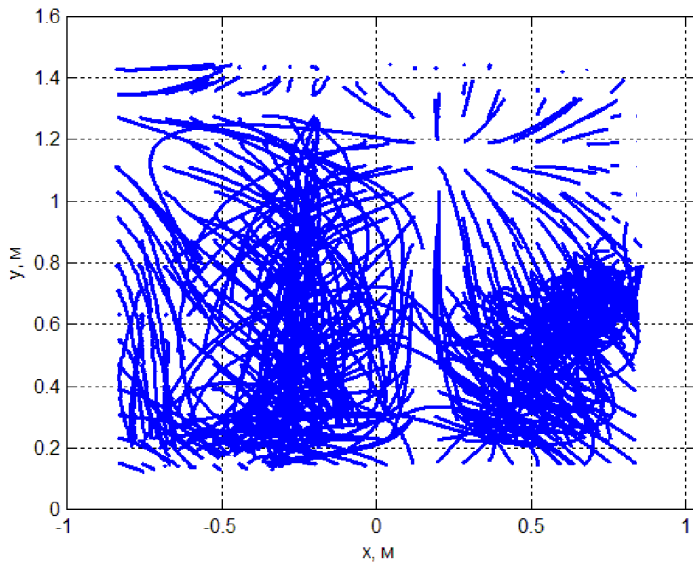
В качестве примера рассмотрим задачу идентификации математической модели движения макета по столу. Простой эксперимент показывает несостоятельность модели равномерного и прямолинейного движения в отсутствие управления.

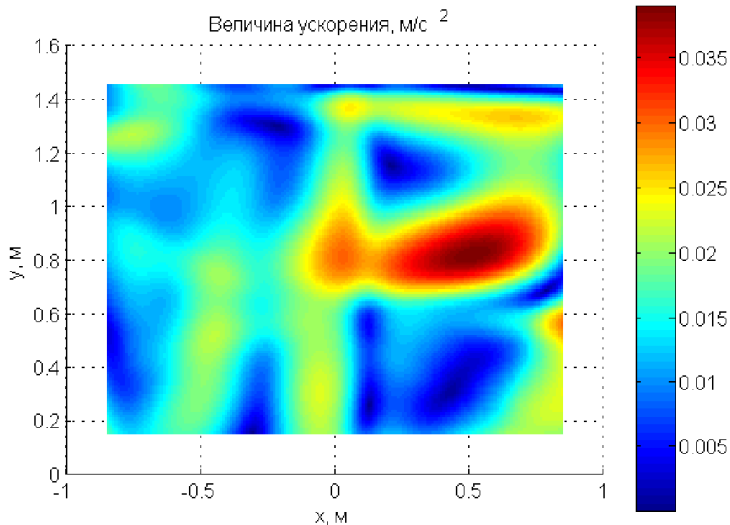


Воздушный
поток

Для идентификации модели была проведена серия замеров, с целью экспериментального определения поля сил и моментов в предположении стационарности этих сил.







Решая очередную задачу пользователю требовалось полностью реализовывать ПО исполняемое на борту макета (за исключением базовых функций работы с сенсорами и управляющими механизмами).

Однако подавляющее большинство задач укладываются в типовую последовательность периодически выполняемых функций:

- Обработка вектора состояния
- Генерирование высокоуровневого командного сигнала (траектории, ориентации и т.д.)
- Вычисление управляющих воздействий в терминах сил прикладываемых к макету
- Перевод управляющих сил в непосредственные сигналы управления

Для упрощения взаимодействия пользователя со стендом, не сужая класс решаемых задач, реализовать комплекс ПО включающий в себя бортовой модуль (ядро) и модуль станции управления (станция).

C++, pthreads. Для учёта пользовательской реализации основных функций потока управления движением ядро пересобирается перед каждым запуском. Алгоритм работы ядра:

- Инициализация всех необходимых систем
- Установка соединения со станцией, получение параметров миссии
- Создание и конфигурирование потоков считывания показаний датчиков и управления движением

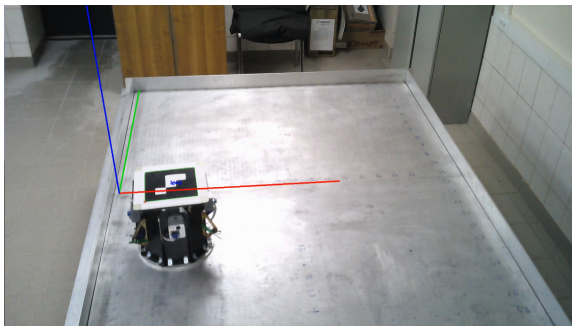
C++, QT для gui и работы с сетью, OpenCV (Aruco) для определения координат. Основным понятием является “миссия” - набор параметров описывающих решение текущей задачи. Станция выступает в качестве как конфигуратора, так и системы мониторинга и управления текущей миссией.

Сценарий работы со станцией



- Доработать графический интерфейс для более удобного взаимодействия со стендом
- Предусмотреть возможность реализации функций потока управления на языках программирования отличных от C++

Стыковка после гашения колебаний



Работа выполнена при поддержке грантов РФФИ
№ 17-01-00449 и 16-01-00739.