An Educational Platform for Small Satellite Development with Proximity Operation Capabilities

Ivan R. Bertaska and John Rakoczy
NASA Marshall Space Flight Center

August 4, 2018

AIAA/USU Small Sat Conference 2018

Logan, Utah, USA

Introduction

- Relevant NASA/MSFC Strategic Goals for 2018:
  - Technical:
    - 2.2: Conduct Exploration in Deep Space, Including to the Surface of the Moon.
  - Educational
    - 3.3: Inspire and Engage the Public in Aeronautics, Space, and Science.

- *Can we combine the above such that they work in conjunction rather than against each other?*
  - Engage the public through our internship programs, while still maintaining technical excellence through *technically interesting* projects.
  - Maintain a broad applicability to students of varying backgrounds (ME, EE, AE, CS, etc.) and levels of education (high school through graduate).

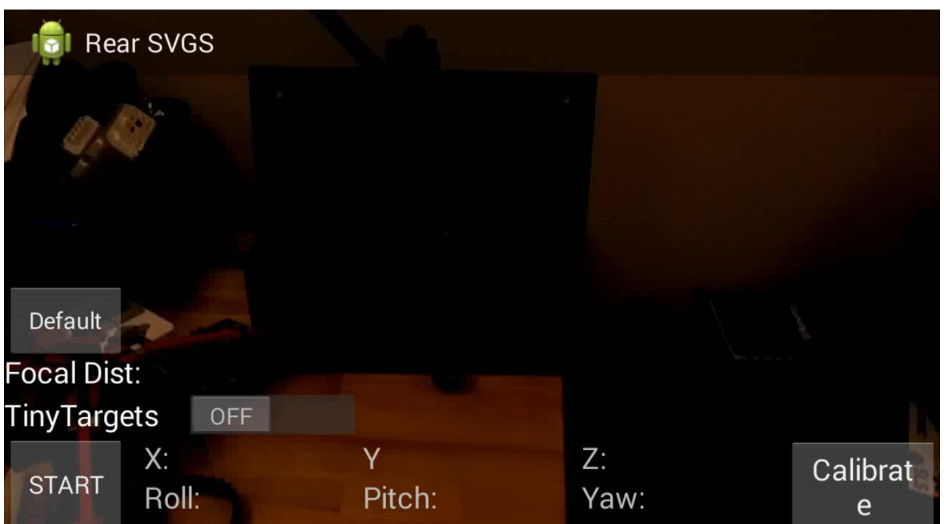
- Present students with concepts in a familiar package…

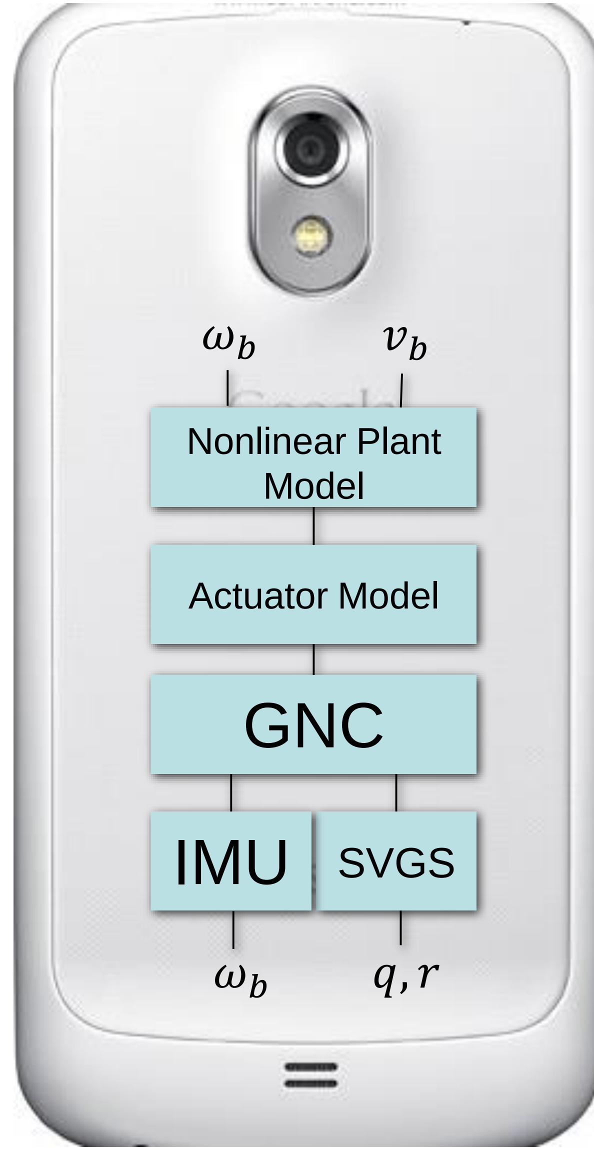
Cellphone

LEGO

Satellite Simulator App with Proximity Operations Capabilities

- *"One app that can simulate a full small satellite"*
  - Camera for relative position and orientation from a target – "SVGS"
  - Use onboard sensors (IMU) to measure rotation rates and
  - Run control system and plant dynamics in real-time.
  - Implemented in Java.

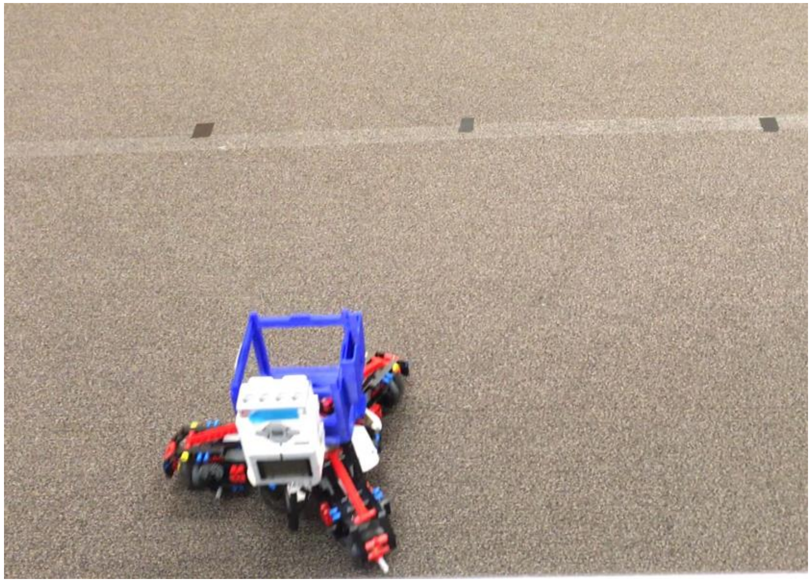




Agilis Omnidirectional Robot

- LEGO EV3 design for a wheeled robotic platform.
- "Agilis*" robot provides unconstrained, omnidirectional movement in longitudinal, lateral, and rotational directions.
- Rotacaster wheels on each leg allow for travel in both longitudinal and transverse directions.
  - Rotacaster wheels have rollers whose axes of rotation are orthogonal to that of the main wheel hub.
  - Allows for the wheel to "slip" in transverse direction.
- The Agilis can match any desired planar velocities (u, v) and rotation rate (ω_z).
- *=> Same degrees of freedom as satellite floating on a flat floor.*

*Agilis instructions found at https://makezine.com/projects/lego-holonomic-robot/



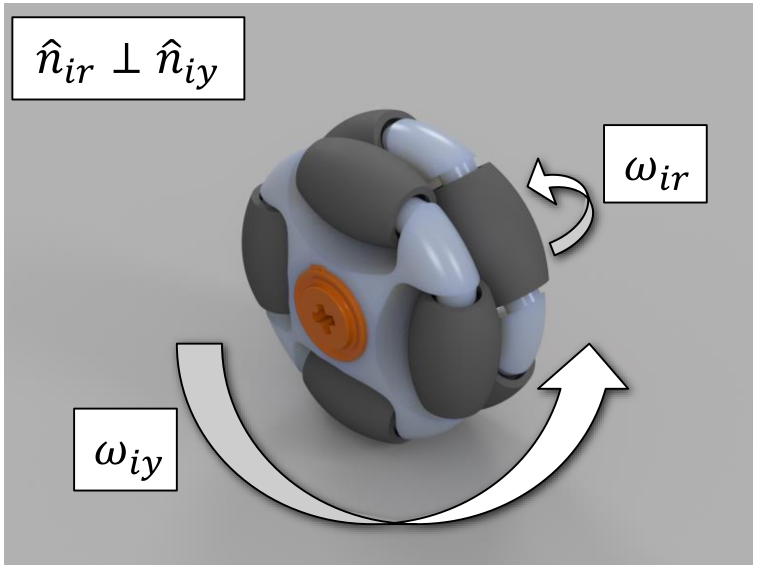
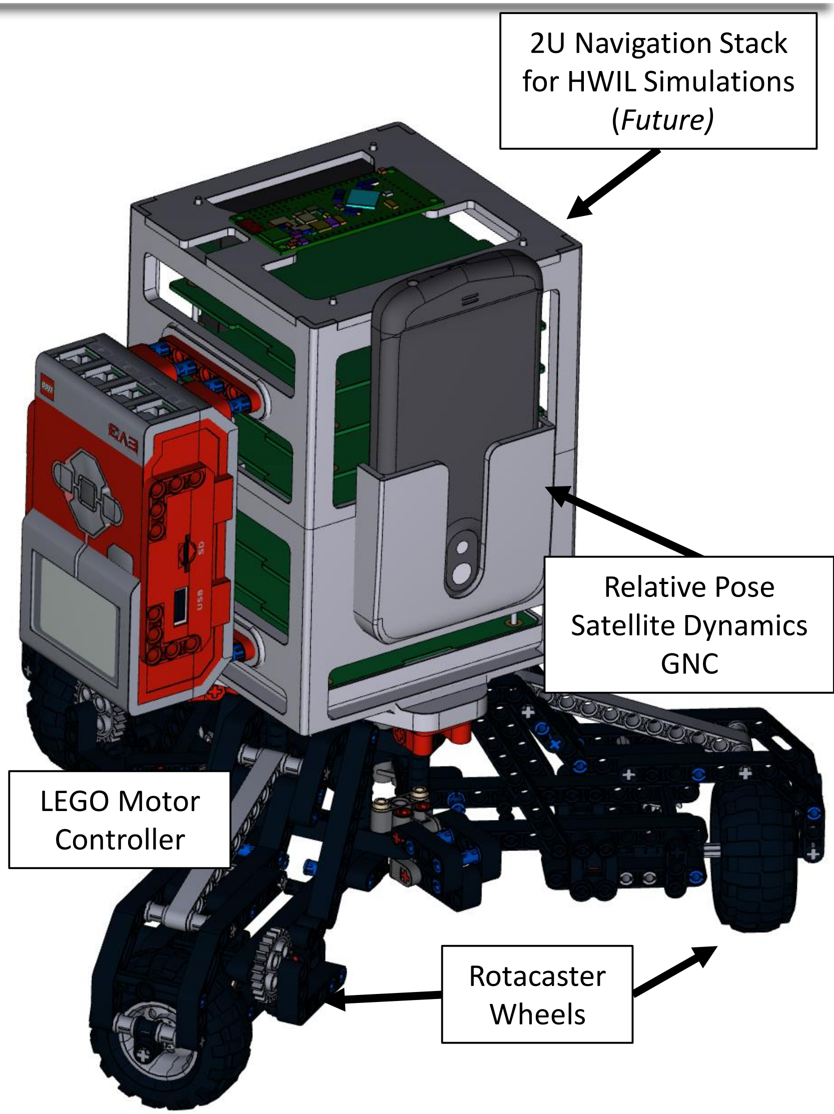


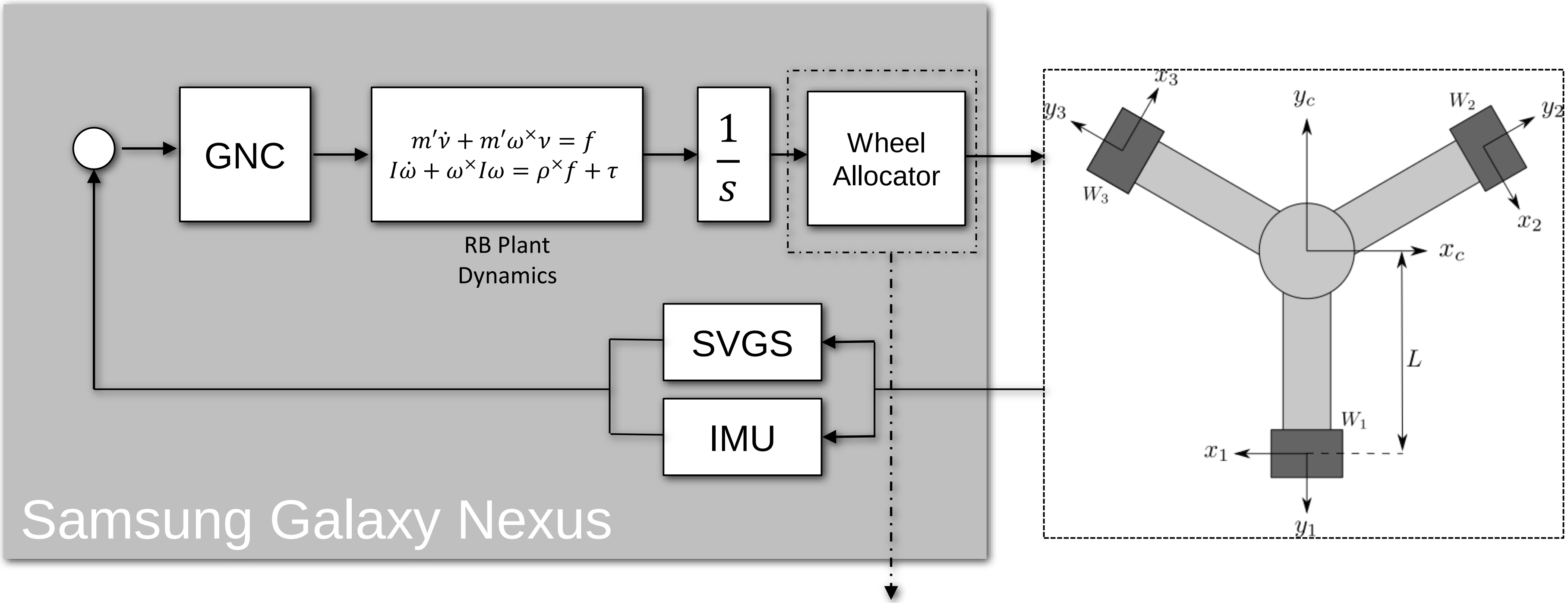
Photo courtesy of grabcad.com

Agilis Satellite Kinematic Simulator (ASKS) Platform

- “Close the loop” around the satellite simulator app using the Agilis platform.
  - Behaves much like a satellite floating on an air bearing, with the exception that dynamics are not closed loop – only the kinematics.
  - Satellite dynamics and actuators are simulated within app, velocities are output from phone to an inverse kinematic allocator, which transfers body velocities to wheel velocities.
- Although the ASKS provides only a partial physical realization of the system, it contains a number of advantages over a traditional air-bearing setup:
  - Motion of the platform is not restricted to flat floors
  - Not constrained by hoses are air tanks.
  - Less overhead – only the mission planning GNC, ADCS/navigation sensors are integrated.
  - GNC and mission logic can be rapidly reiterated and tested.
  - More familiar hardware for students.
  - Can be run with no expensive or ESD-sensitive hardware (just the phone and LEGO MCU).



ASKS Inverse Kinematics



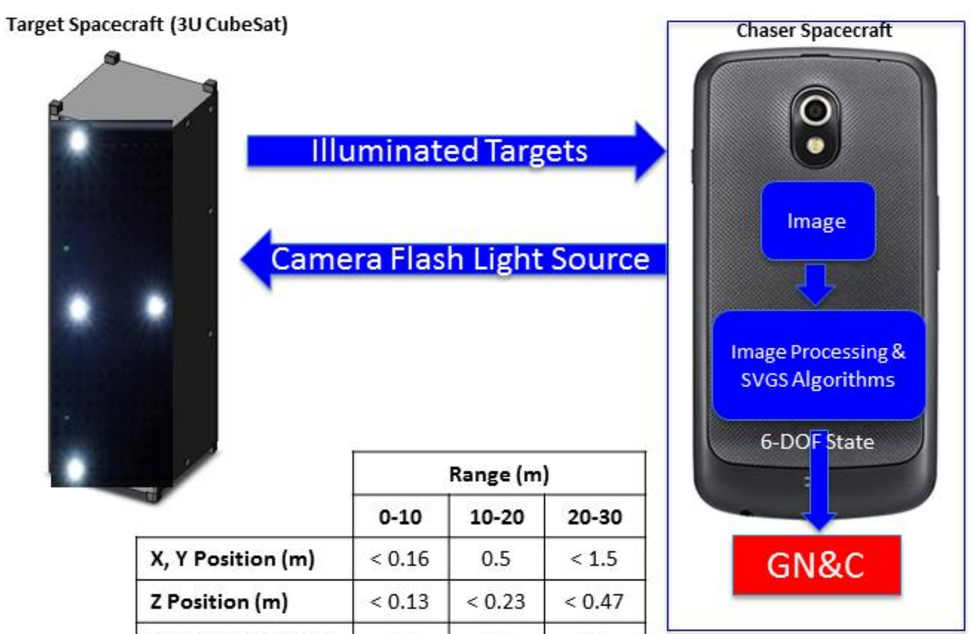
- Allocator for wheel velocities is derived in fully in paper, but becomes a simple transformation from 3DOF velocities to wheel speeds.
- Simulated 3DOF velocities are transformed into wheel speeds by,

$$\begin{bmatrix}\omega_{1y}\\ \omega_{2y}\\ \omega_{3y}\end{bmatrix} = \begin{bmatrix} -\frac{1}{R} & 0 & -\frac{L}{R} \\ \frac{1}{2R} & -\frac{\sqrt{3}}{2R} & -\frac{L}{R} \\ \frac{1}{2R} & \frac{\sqrt{3}}{2R} & -\frac{L}{R} \end{bmatrix} \begin{bmatrix} u \\ v \\ \omega_z \end{bmatrix}$$

*no dependence on roller velocity ω_{ir}

Smartphone Video Guidance Sensor (SVGS)

- The SVGS is a relative positon and orientation sensor based on MSFC technology with flight heritage, the Advanced Video Guidance System (AVGS).
  - DART (2005) and Orbital Express (2007) missions.
- SVGS performs the same functionality as the AVGS in the form factor of a "smartphone."
- A target consisting of retroreflective corner cubes or LEDS with a known configuration is mounted on the target spacecraft.
- Image of target is captured by camera and process.
- SVGS uses an inverse perspective algorithm with an adaptation of the collinearity equations to produce the 6DOF states (position x, y, z and 3-2-1 Euler angle rotation ϕ, θ, ψ) between the camera and target.



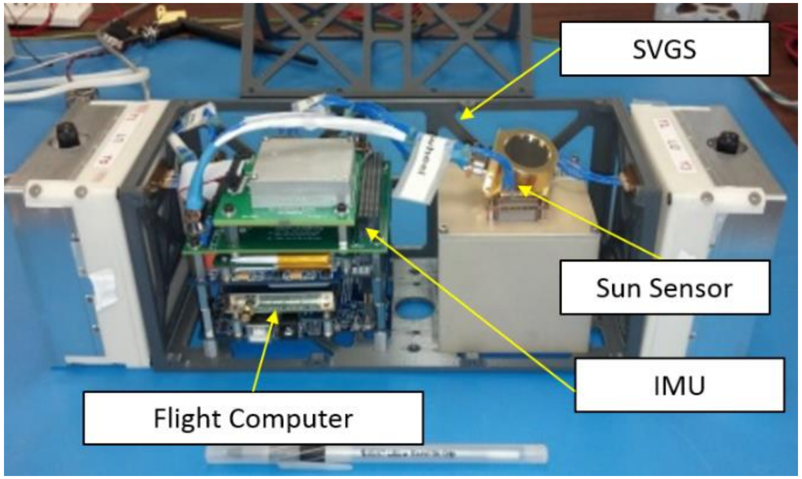
	Range (m)		
	0-10	10-20	20-30
X, Y Position (m)	< 0.16	0.5	< 1.5
Z Position (m)	< 0.13	< 0.23	< 0.47
RPY attitude (deg)	< 2	< 2	< 3

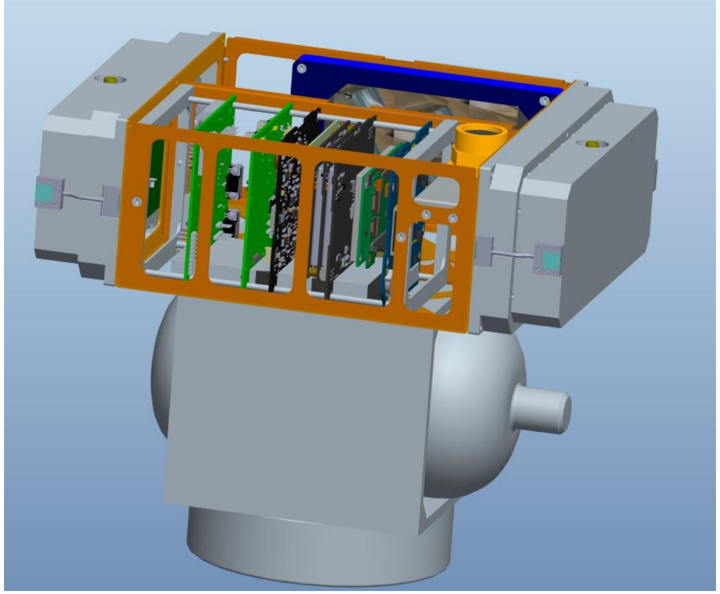
- SVGS has been ported to a variety of platforms (including non-smartphones):
  - Samsung Galaxy Nexus
  - Samsung Galaxy S8
  - Inforce 6501 (NASA Astrobee board)
  - Raspberry Pi (in development)
  - C implementation
  - Java implementation
  - Python implementation

3U and 6U CubeSat Model

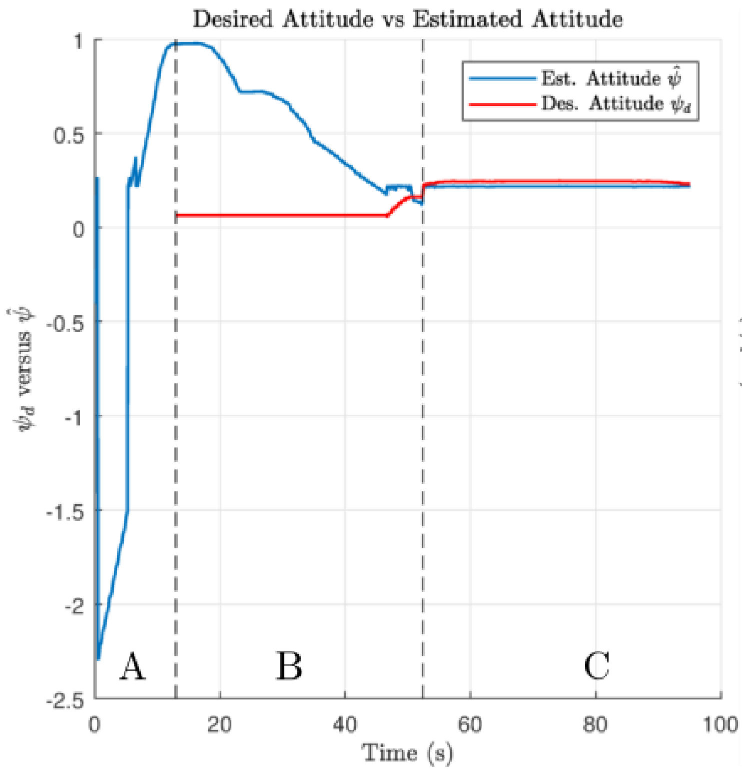
- Two models of “real” CubeSats are currently implemented in the satellite simulator app:
  1. “Underactuated” 3U CubeSat - only actuated in lateral direction.
     - MAI 10mN-m-s Reaction Wheel Assembly
     - ADIS16488 IMU
     - Sinclair Interplanetary Sun Sensor
     - Two 0.5U University of Arkansas green prop using 1,1,1-3,3,3-hexaflouropropane.
     - SVGS

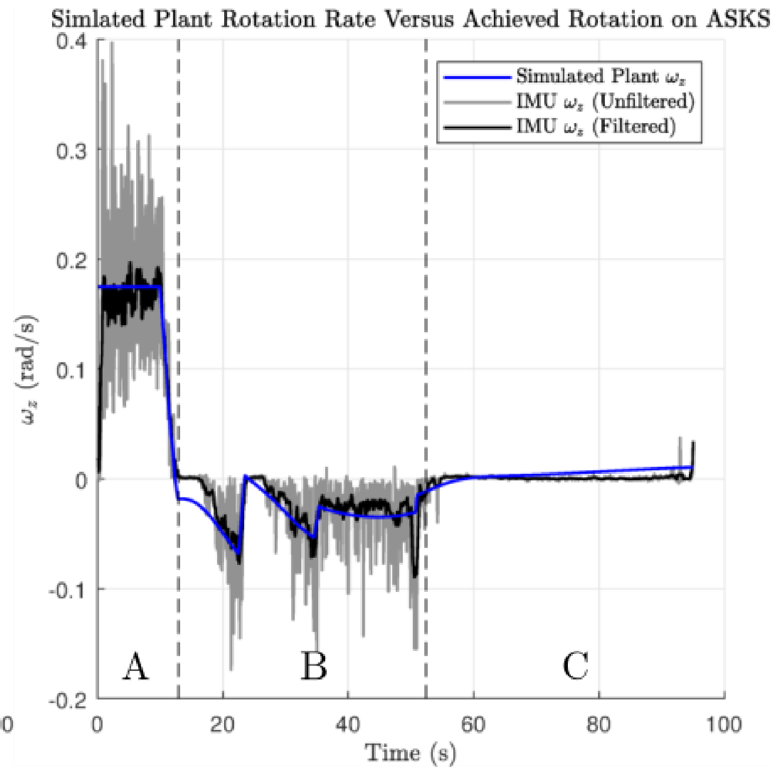
  2. Fully actuated 6U CubeSat – full actuation in 3DOF
     - Sinclair Interplanetary 30mN-m-s Reaction Wheel (single axis)
     - Sinclair Interplanetary Sun Sensor
     - ADIS16488 IMU
     - Modular Attitude Determination System (MADS)
       - MSFC-developed board that interfaces to sensors. Optionally capable of performing attitude and navigation filtering.
     - Two 1U University of Arkansas green prop using 1,1,1-3,3,3-hexaflouropropane.
     - SVGS





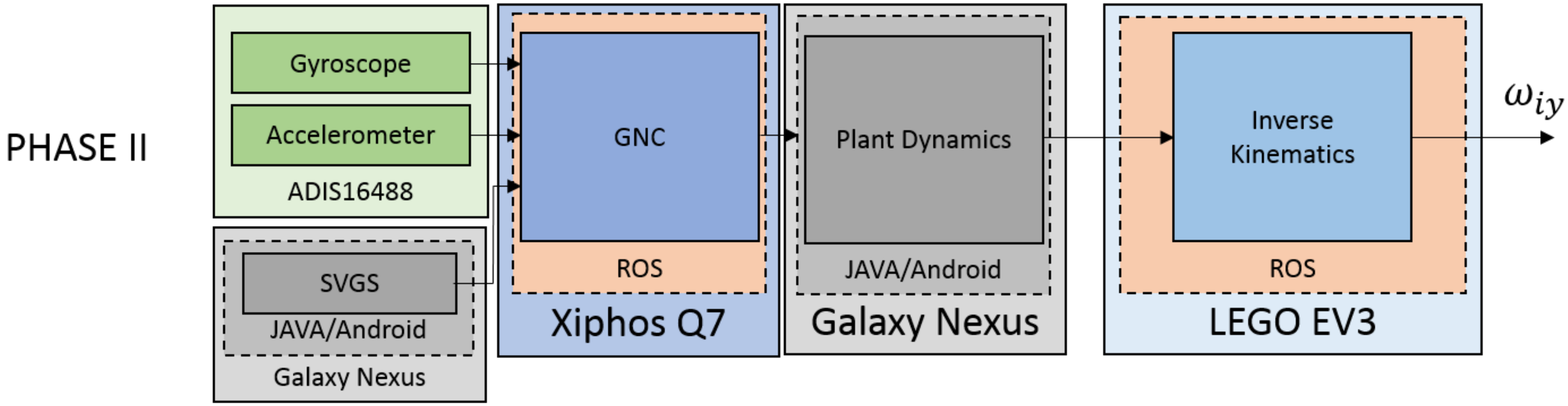
3U Detumble and Point Results





- Successfully replicated plant velocities with ASKS. Single DOF experiment.
  - A: “detumble” event from “tipoff” of 10dps
  - B: slew towards target
  - C: point at target (i.e., “science” portion of mission)
- Right: coplots of unfiltered IMU rate (gray), filtered IMU rate (black), and simulated plant rate (blue). For successful plant velocity reproduction blue must match black.
  - Small difference at ~80 seconds between simulated plant rate and sensed rate can be attributed to quantization in the wheel controller (minimum resolution for commanded wheel velocities is 1dps)
  - Future redesign of ASKS platform will increase step-down gear ratio to desensitize system to quantization effects.

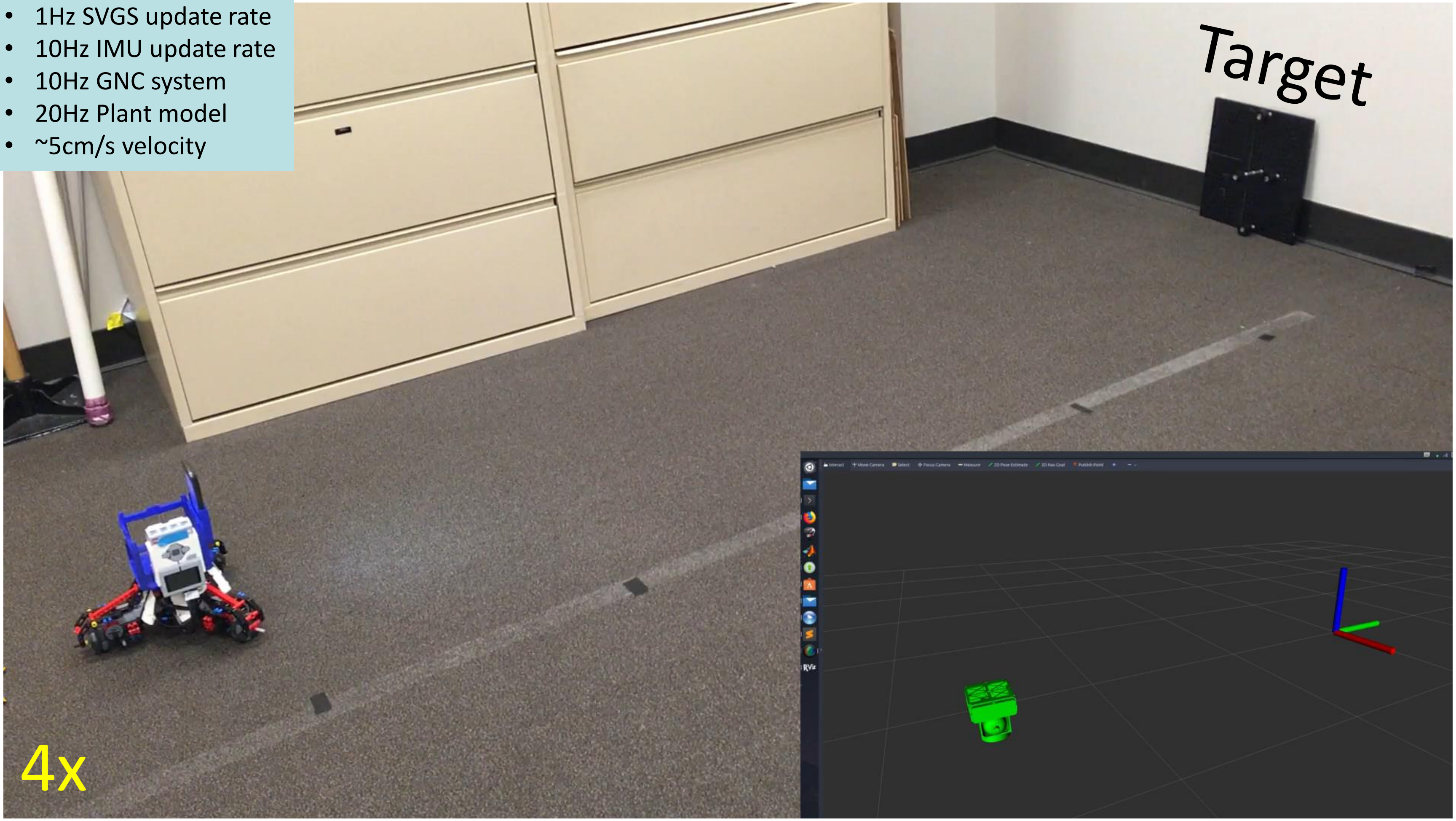
Future Developments



- Tailor project to leverage recent technological trends and intern/student knowledge and skills.
  - Learning curve for RTOS, transition system to Robot Operating System (ROS)
    - Using FreeRTOS framework, up to 6 weeks for interns to familiarize with architecture.
    - ROS, as little as two weeks experience needed.
  - Python is growing in popularity, while fewer students are entering program with C/C++ experience. Nearly all have some level of Python.
  - Embedded Linux-based SBCs, e.g. RPi, Beaglebone up
    - Q7 is a space-grade, Linux SBC with a bash shell and compatible with the opkg package manager
- Restructuring into ROS architecture.
  - Fully featured real-time and post-processed visualizations and off-the-shelf capabilities.
  - Well defined interfaces allow for easy implementation of Hardware Abstractions Layers (HAL) to swap between PIL, HWIL, and simulation.
- Allows for the integration of a 2U navigation stack to perform HWIL simulations with actual sensors (SVGS, IMU, etc.) and flight computer.

Future Developments - 6U 3DOF Control Results

- 1Hz SVGS update rate
- 10Hz IMU update rate
- 10Hz GNC system
- 20Hz Plant model
- ~5cm/s velocity



- 3DOF position and orientation control via SVGS
- Post-processed results gives insight into system. Visualization at bottom-right is generated using ROS RViz graphical tool.
- Full integration of ROS allows for real-time visualization and debugging support.

Conclusion and Acknowledgements

- An educational tool was developed to simulate satellite dynamics in planar 3DOF motion. It uses two main elements to “close the loop” around simulated satellite kinematics.
  1. Satellite simulator Android app – uses in-built IMU and camera (SVGS) to sense states, run GNC system, and plant dynamics.
  2. Agilis platform is an LEGO-based robot using rotacaster wheels for omnidirectional motion. Uses plant velocities from Satellite simulator app to reproduce 3DOF planar motion of a satellite (like in air bearing platform)
- Capability to experimentally validate proximity operations through SVGS.
  – ASKS platform hold both an SVGS-capable phone and target.
- Experiments demonstrate successful reproduction of plant velocities using the ASKS platform and full 3DOF motion control of the ASKS platform
- Future developments for the ASKS platform leverage technological trends to create a system that is intuitive to students, which can extend their capabilities rather than learn a new system wholesale.
  – Python implementation, ROS, use of embedded Linux SBC.
  – ROS allows for a rich graphical front end “out of the box” and simple implementation which reduces learning curve when compared to FreeRTOS-based architecture.
  – HWIL integration with 2U navigation stack.

- A special thanks goes out to interns that helped me with this project:
  – Zachary Prihoda, Elijah Johnson, Amarri Cole, Tevon Walker, Eric Bradshaw, Dylan Hunziker, Chris Carter, Logan Bolingbroke, Zakk Giacometti, Brian Adams, James Holt, Bryce Brown, Joshua Bryant, Robinson Czajkowksi, Jacob Jensen