Virtual Testing Environment for Flexible Drone Flight Simulation and Design Analysis

Nicholas Rocchi, Computer Systems Engineering
Mentor: Wenlong Zhang, Associate Professor
The Polytechnic School / School of Manufacturing Systems and Networks

Research Question

- What improvements can be made to the data collection process for flexible drones with the use of testing in a virtual environment?
- What effect does using a virtual environment have on the testing of flight pathing and execution of a drone's hardware and software?

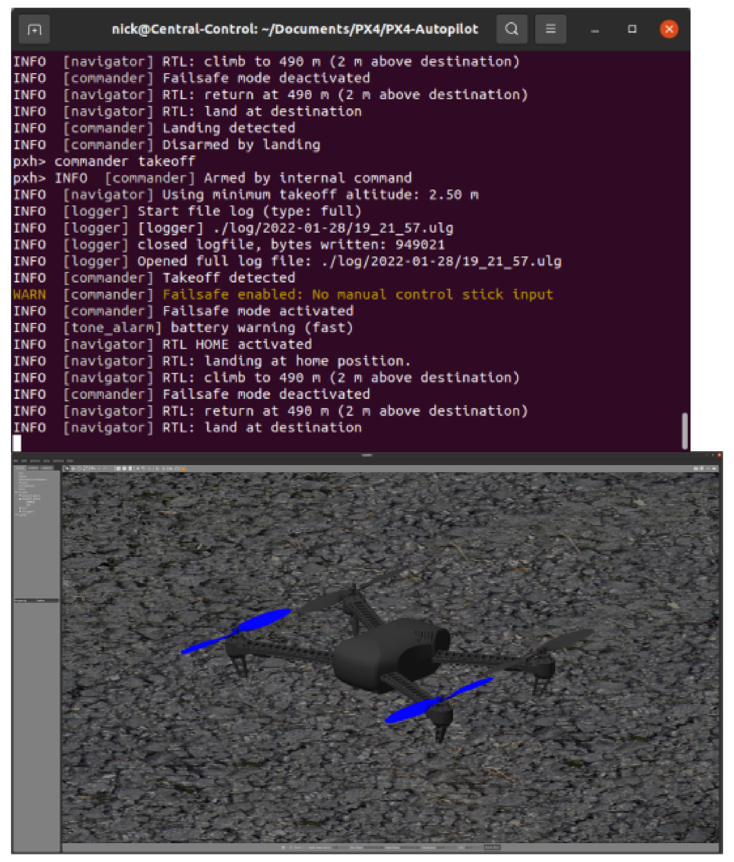
Research Progress

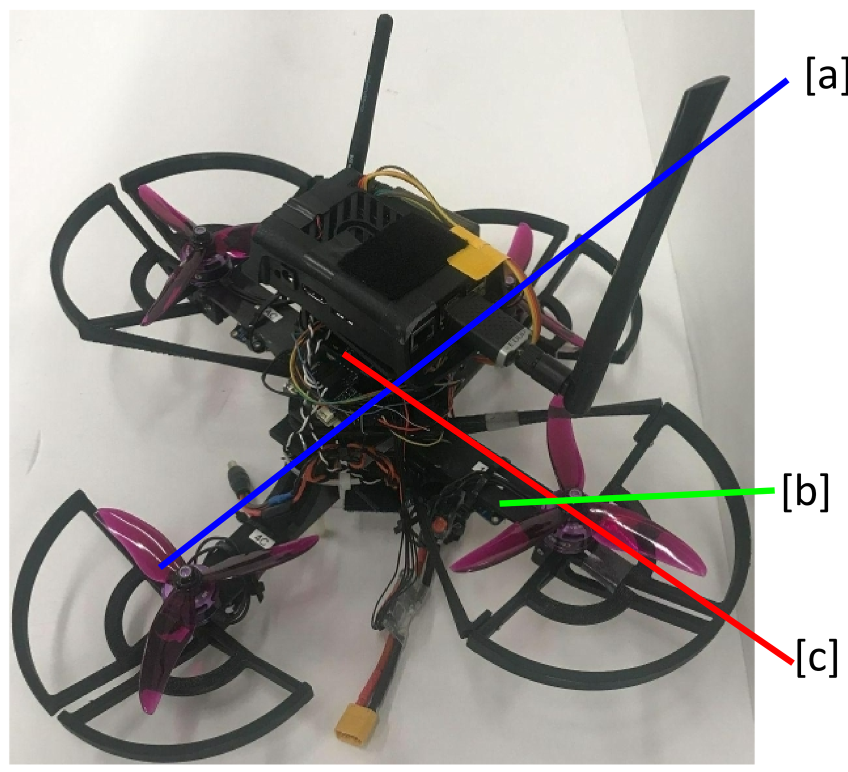
- Got set up with the software needed for the project
- Tested different methods/programs for modeling and data collection
- Created a 3d cad model prototype for the drone to be simulated
  - made of both modified existing models and custom made ones
- Implemented most parts- custom or modified- into the simulation individually- not combined into one drone yet

Future Plans

- Further improve the accuracy of the cad model until the model is close to 100% accurate
- Compare the manual computations & model analysis data to improve computer analysis methods
- Implement the drone fully into the simulation to allow for flight tests and not just physical analysis

Drone Solidworks Model





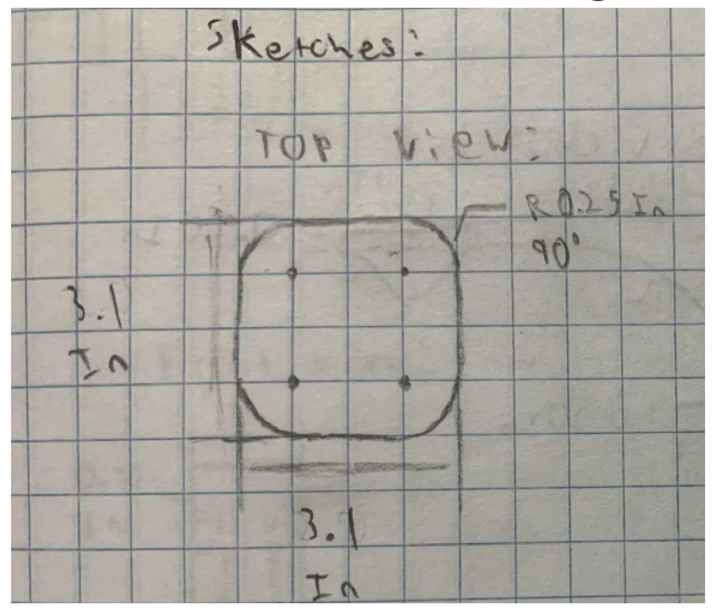
[a] Custom model for the GemFan Hurricane, modeled using Solidworks 2021, using official specifications, and a modified PolyCarb material file from Solidworks, custom made for greater accuracy than publicly available models could provide.

[b] Rigid Arm for the Squeeze drone propellor arms, ported from a .stl mesh file using Solidworks 2021, using a custom material for PLA plastic, edited for software performance in Solidworks 2021.

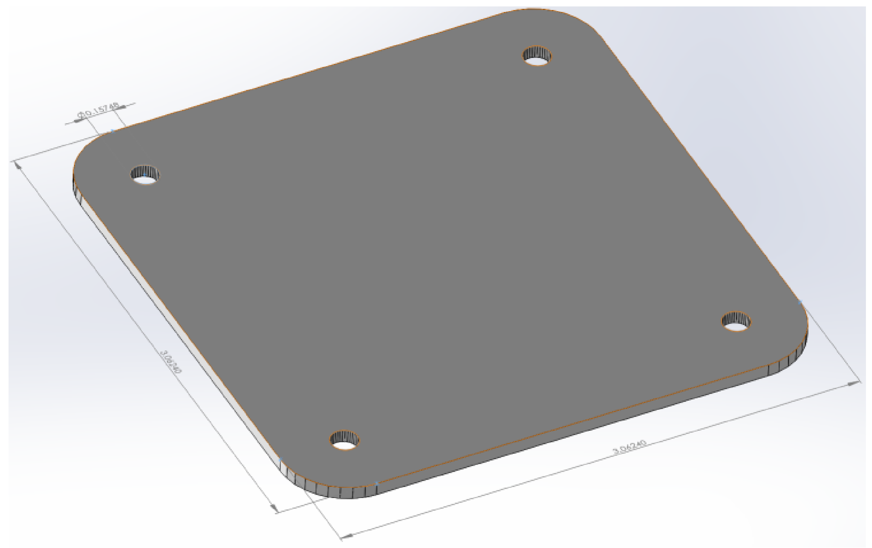
[c] Adafruit BNO055, made in Solidworks 2021 by Pamir on GrabCad, resized using Solidworks 2021, custom metal material made in solidworks for density and material properties accuracy.

Modeling and Data Analysis of Parts

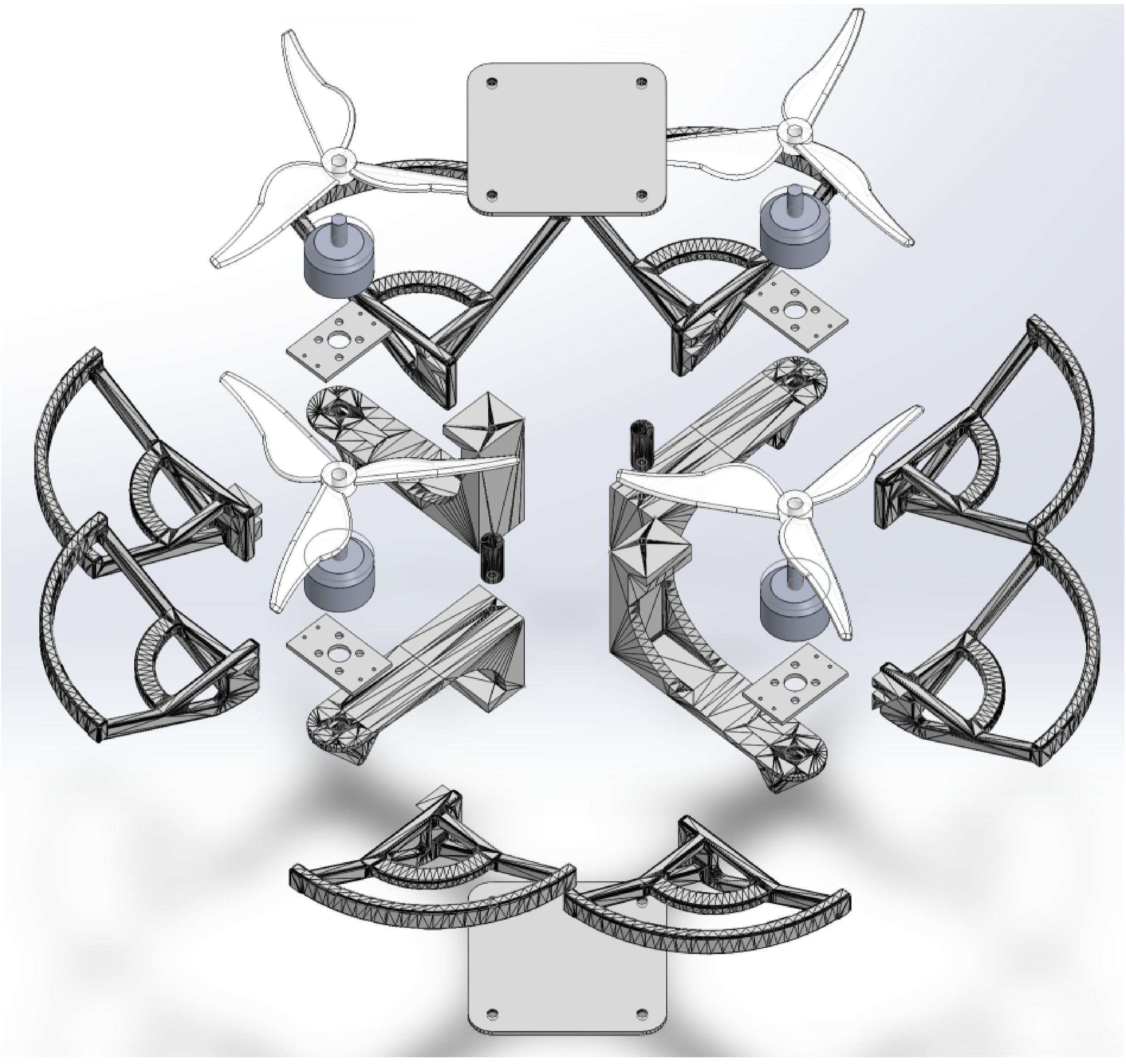
Sketch and dimensioning



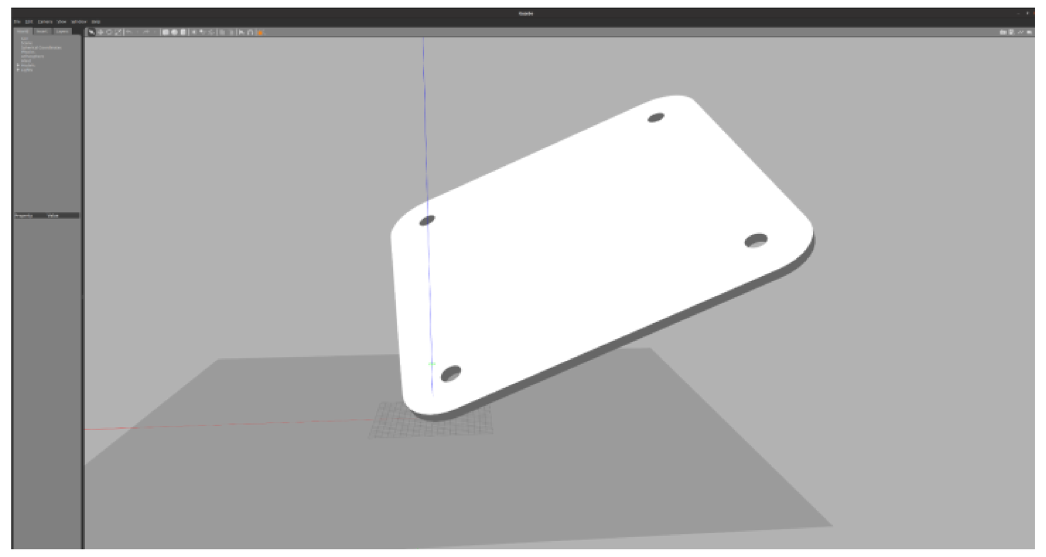
Modeling and cross-referencing



Exploded View of Prototype Drone Model



Incorporating into the Simulation



Data comparison: Manual computation vs Computer Analysis

Equations (for manual calculations)

J_{xx}	$\frac{2}{5}MR^2 + M(cg_y^2 + cg_z^2) + 2m(cg_y^2 + cg_z^2) + m((-l\sin\beta_1 - cg_y)^2 + (h - cg_z)^2) + m((l\sin\beta_2 - cg_y)^2 + (h - cg_z)^2)$
J_{yy}	$\frac{2}{5}MR^2 + M(cg_x^2 + cg_z^2) + m((l - cg_x)^2 + cg_z^2) + m((-l - cg_x)^2 + cg_z^2) + m((-l\cos\beta_1 - cg_x)^2 + (h - cg_z)^2)$ $m((-l\cos\beta_2 - cg_x)^2 + (h - cg_z)^2)$
J_{zz}	$\frac{2}{5}MR^2 + M(cg_x^2 + cg_y^2) + m(cg_y^2 + (l - cg_x)^2) + m(cg_y^2 + (-l - cg_x)^2) + m((-l\sin\beta_1 - cg_y)^2 + (-l\cos\beta_1 - cg_x)^2)$ $+m((l\sin\beta_2 - cg_y)^2 + (-l\cos\beta_2 - cg_x)^2)$
$J_{xy} = J_{yx}$	$Mcg_ycg_x - mcg_y(l - cg_x) - mcg_y(-l - cg_x) + m(-l\sin\beta_1 - cg_y)(-l\cos\beta_1 - cg_x)$
$J_{yz} = J_{zy}$	$Mcg_ycg_z + 2mcg_ycg_z + m(-l\sin\beta_1 - cg_y)(h - cg_z) + m(l\sin\beta_2 - cg_y)(h - cg_z)$
$J_{zx} = J_{xz}$	$Mcg_xcg_z - m(l - cg_x)cg_z - m(-l - cg_x)cg_z + m(-l\cos\beta_1 - cg_x)(h - cg_z) + m(-l\cos\beta_2 - cg_x)(h - cg_z)$
cg_{xyz}	$(-Cl(\sin(B_1)-\sin(B_2)),\ -Cl(\cos(B_1)+\cos(B_2)),\ 2Ch)$

Constants

B_1, B_2	0 Degrees
m	$95*10^{-3}$Kg
M	$710*10^{-3}$Kg
h	$-3*10^{-2}$m
R	$5*10^{-2}$m
l	$12.5*10^{-2}$m
C	$8.71*10^{-2}$

Results of calculations

Method of calculating	Results (in Kg * m^2) $J_{xx}, J_{yy}, J_{zz}, J_{xy}, J_{zy}, J_{xz}$
Manual Calculation	0.0037,0.0037,0.0067,0.0000,0.0000,0.0000
Machine calculation	0.0041,0.0041,0.0077,0.0000,0.0000,0.0000

Note:
The Machine calculation is a black box, it does not explicitly follow the formulas listed above

Acknowledgement and References

I thank Karishma Patnaik, Dr. Wenlong Zhang, and Jonathan Bush for their guidance and/or advice on this project
Models of the Intel Upboard, FrSky D4R, PX4 PixRacer, and Adafruit BNO055 courtesy of Mehrad Ghanbari, Chris, Sergey Torshin, and Pamir on Grabcad respectively

[1] K. Patnaik, S. Mishra, S. M. R. Sorkhabadi and W. Zhang, "Design and Control of SQUEEZE: A Spring-augmented QUadrotor for intEractions with the Environment to squeeZE-and-fly," 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020, pp. 1364-1370, doi: 10.1109/IROS45743.2020.9341730.