Agile Trajectory Generation for Tensile Perching with Aerial Robots

Progress Update

- Explored Drake Simulation
  - Spend 2 days starting to learn how to use.
  - Conclusion: Drake may be too complex of a platform to learn and requires defining the majority of the physics of the simulation from scratch.
- Returned to PyBullet utilising demo trajectory
  - Demos
  - Overall seems to work reasonably well actually with 10 segments
  - Increasing the number of segments:

Increasing the number of segments:

- Does increase computational cost but not to an extreme level. (2x CPU, 3x Memory for 10x segments i.e. 100 segments)
- Current issue is with the type of segments - using a Point2Point connection between each segments.
  - This doesn't work very well since in Bullet this is done through applying a force to keep these points together.
  - There are other types of joints such as Spherical joints but they're not exposed on the python library. The python lib is a wrapper around the C++ lib.

Solution:

- Statically define it using a model file.
- Fork my own version of PyBullet and add this in programatically.

Plans Until Next

- Mainly focused on the simulation environment:
  - Attempt one of the above solutions.
  - Adjust dynamics of simulation to match previous environments.
  - Hopefully be able to show a full demonstration in simultaion.

Questions

- Demonstrations from previous project? Previous repo has:
  - Analytical solution path for wrapping (used above in simulation)
  - Data from the ML training process - could probably find a way to use this.
  - But no complete trajectories - is this available?