Risk: Hardware

- Due to the lack of access to the appropriate tools and work space, we will no longer be working on a physical version of the power management system and servo motor driver. Instead we will be designing this board but not actually manufacturing it. Given that we were still on campus we would be able to manufacture and test this easily.
- To assess whether the motors are actually capable of meeting the speeds and torque that are advertised, they will be tested with a 3D printed pan and tilt mechanism.
- The previous risk of being able to properly use the battery will no longer be testable without PCB previously mentioned.

Risk: Integration

With all members of the team distributed across the Western hemisphere, integrating the project's varied subsystems becomes a major concern. For clarification, software integration is not the principal risk in this matter as this work can be completed remotely. With the delays posed by the acclimation process to COVID-19 changes, Demo #1 will be a time for this team to complete most independent software functionality, with Demo #2 focusing on integrating the different software subsystems together. The principal risk in this matter is integrating the software work into the hardware system. With only one team member, Ismael Mercier, having access to the resources required to operate a hardware project, a concerted effort to integrate the full team's work no longer becomes feasible. To address this risk, this team has coordinated a plan with the course administrators to develop and demonstrate the independent functionality of each subsystem with all points of integration clearly identified, yet without taking the final step of integration itself. Although this compromise is a disappointment to all team members, the reality of the COVID-19 landscape forces our hand to make these decisions and create the best product possible.

Risk: Long video storage requirements

This risk isn't the result of having to work remotely due to the COVID19 pandemic, but more so something that came up recently in our team discussions. The issue is that images with resolutions even as low as 480p take up about 1 MB when not compressed. As a result, if we need to save every frame in order to compile them into a video only when finished, we could quickly run out of local memory -- especially since the AI optimizations of TensorRT on the Jetson use up so much of its memory.

One way that we're thinking of achieving local storage is to use OpenCV's VideoWriter object. It supports only the *avi* video format and can only save videos of up to 2 GB. If long or high frame-rate videos at high resolutions are a must, we will have to look into different ways of compiling videos. It is important to note that OpenCV is mainly a computer vision library and there are bound to be improved video streams/codecs that can help save longer/higher quality videos.