Suggested System Improvements

1. Hovering in place still needs to be tested without the test platform, which will also require tuning the x,y,z PD gains. I’m hesitant to test this now, because the attitude control still seems pretty shaky. The next two points are possibly ways to improve the attitude control performance. Once either or both of these make the attitude very stable, we can test the drone off the platform.
2. The sensors are pretty noisy. Re-implementing some version of the filters may improve performance.
3. Currently, the control loop updates at around 80Hz, but it’s noisy, and sometimes it takes longer. The suggested rate is 100-200Hz (by word of mouth). The control loop would be faster and more consistent if it was implemented on a microcontroller/microprocessor that’s supervised by the RPI.
4. To implement this, you would essentially implement everything we have now on a microcontroller, with IMU input and PWM outputs to the motors. You’d keep the relay and voltage measurement connections with the RPI, as well as the Vicon measurements, since those all operate on a slower time scale (or in the case of the relay, it’s just an interrupt). Then you’d connect the RPI to the microcontroller over I2C or SPI connection. In the microcontroller firmware, you would set things like the PID gains, the battery voltage, position setpoints, etc, to variables that will be set by the RPI. That way, the attitude control can run multiple loops before the RPI interrupts to update some slower-changing parameters.
5. Communication between the RPI and microcontroller still needs to be treated carefully, or the RPI will just interrupt and slow down the microcontroller. The best way to do this looks like it will be by using Direct Memory Access (DMA) or possibly Dual Core operation, although you might get by using priority interrupts and buffers or queues
6. For choice of microcontroller,
   * 1. STM32 will probably give the best performance, though it may be the most difficult to implement. Some models have DMA
     2. ESP32 may be a little easier to work with, still with ok performance. Some models have Dual Core operation.
     3. Arduino would be much simpler but probably wouldn’t give good enough performance.
7. This will be a pretty substantial undertaking. It would probably be best to keep the current drone intact and attempt this on a new drone build. I’d recommend using ChatGPT heavily to help develop this system.
8. Once the attitude control and x,y,z control are established, we can implement some sort of trajectory tracking and path planning. Or multiagent control with multiple drones. Or model-based or data-based controller designs. At this point, it’s whatever is most useful for the actual research.