A Preliminary Calculations

A.1 Power Usage Estimate

The following estimate assumes "typical maximum" operating conditions, as this system is capable of producing far greater control than is necessary for the DSFC:

- Thermoregulation at 25% power for holding temperature;
- LEDs set to 20% power (comparable to sunlight);

Estimate:

- Automation Computer (Raspberry Pi) - **5W Average**
- Diaphragm Pump - **24W**
- Lighting - 5 series x 3 LEDs/series x 9 boards/unit x 12 units x 3W/LED x 20% power (typ.) = **972W**
- All Thermoregulation Peltier Tiles - 4 tiles per unit x 2 units (1 water + 1 air) x 8.5A x 15V x 25% power (typ.) = **255W**
- Runoff Recycling Peristaltic Pump - **3W**
- All Sensors - **10W**
- All Fans - 10 fans (2+2 air thermo, 2 water thermo, 1+1 gas exchange, 2 humidification/dehumidification) x 2W per fan = **20W**
- Humidification Driver - **10W**
- Primary Aeroponic Solenoid - **5W**

A.2 Water Usage Estimate

Humidification: By using a mesh nebulizer to produce smaller and more consistent vapour, greater overall water consumption efficiency was achieved.

Aeroponics: Aeroponics by design uses far less water than traditional farming. In addition, higher quality nozzles with adjustable directionality allow for more of the water to be sprayed directly at the root zone and with better and more consistent droplet sizes for better uptake. Finally, by enclosing the root zone in a watertight container, no water escapes, and runoff water collected at the bottom of the container can be recycled. 5mL/sec/nozzle @80PSI x 2 nozzles per unit x 12 units x 10 seconds misting per hour

In calculation, it is assumed that all aeroponic water is consumed, as all runoff water is recycled.

A.3 Mass Estimate

By using smaller parts, power consumption and complexity was reduced along with volume and mass. PeaPod's mass was also optimized through minimizing density across components. Aluminum was chosen for the framing due to it's high strength to density ratio. For insulation, a less dense foam coated in mylar was used to maintain PeaPod's insulating capabilities while reducing mass.

The following are over-estimates:

Frame and Trays: **54kg** (4.5kg per frame unit) Insulation: **5kg** Control Module: **10kg**

A.4 Crew Time Estimate

Setup: Before plants are grown, crews must make sure solution containers are full, and UV sterilize the housing.

Harvesting: Crew members simply need to remove the trays from the unit to access the grow trays. This largely depends on the plant type, ranging from 5 minutes (picking fruit) to 15 minutes (harvesting root vegetables).

Output processing and storage: Processing will depend on the produce grown and desired end product, varying from boiling to frying to baking etc. average 30 minutes per meal.

A.5 Optimization Method

A common existing approach to plant optimization in academia is to have a simple neural network train on a dataset of **fixed/unchanging** environment parameters (input) and **final** plant metrics (output). This approach is severely limited, in that it does not account for changes over time. Environment parameters cannot be changed, and as such cannot target specific plant growth phases (i.e. specific lighting and nutrients for germination vs vegetation vs fruiting). In addition, the exclusion of plant metrics throughout the duration of plant growth means that a system has no way to respond to failing plant health, as it has no data for this.

Instead, a statistical model is trained on an ordinary differential representation of plant behaviour and phenology, which takes into account the cumulative property of growth: A plant's current state is not *immediately* related to it's surrounding environment (consider a "flash-freeze" step change), but instead a plant's state's rate of change (i.e growth rate and other processes) depend on both its previous instantaneous state and the current environment.

Assume a plant's growth rate (or "state change") is related to its current internal state $\vec{P} \in \mathbb{R}^n$ (for n plant metrics) and the environment conditions $\vec{E} \in \mathbb{R}^m$ (for m environment parameters). Let these both be functions $\vec{P}(t), \vec{E}(t)$ defined at each point in time t, where $t = 0$ indicates the time of planting. Assume that this relationship is constant for all members of a given species.

Define plant state change $\vec{P}'$:

$$\vec{P}'(t) = \frac{d}{dt}\vec{P}(t)$$

Define the plant-environment phenology mapping function Q, a re-definition of the state change in terms of plant and environment states:

$$Q(\vec{P}(t), \vec{E}(t)) = \vec{P}'(t)$$

Given the internal and external state changes, determine the plant's state change across time:

1. Set $\vec{E}_{set}(t)$ for each point in time, aka the program;
2. Record and smooth/filter $\vec{P}(t)$ (plant metrics) and $\vec{E}(t) \approx \vec{E}_{set}(t)$ (environment sensors) for each point in time;
3. Calculate $\vec{P}'(t)$ from $\vec{P}(t)$ for each point in time;
4. Fit Q via a statistical or machine learning model, Q_{model}, using $\vec{P}(t), \vec{E}(t)$ (input) and $\vec{P}'$ (output) at each point in time as our dataset;

The optimal form of Q_{model} is an ongoing topic of investigation.

By fitting Q as Q_{model}, we can predict $\vec{P}$ at any point in time for any program $\vec{E}$. For example:

$$\vec{P}(t_f) = \int_0^{t_f} Q_{model}(\vec{P}(t), \vec{E}(t))\, dt \quad \text{or a numerical method}$$