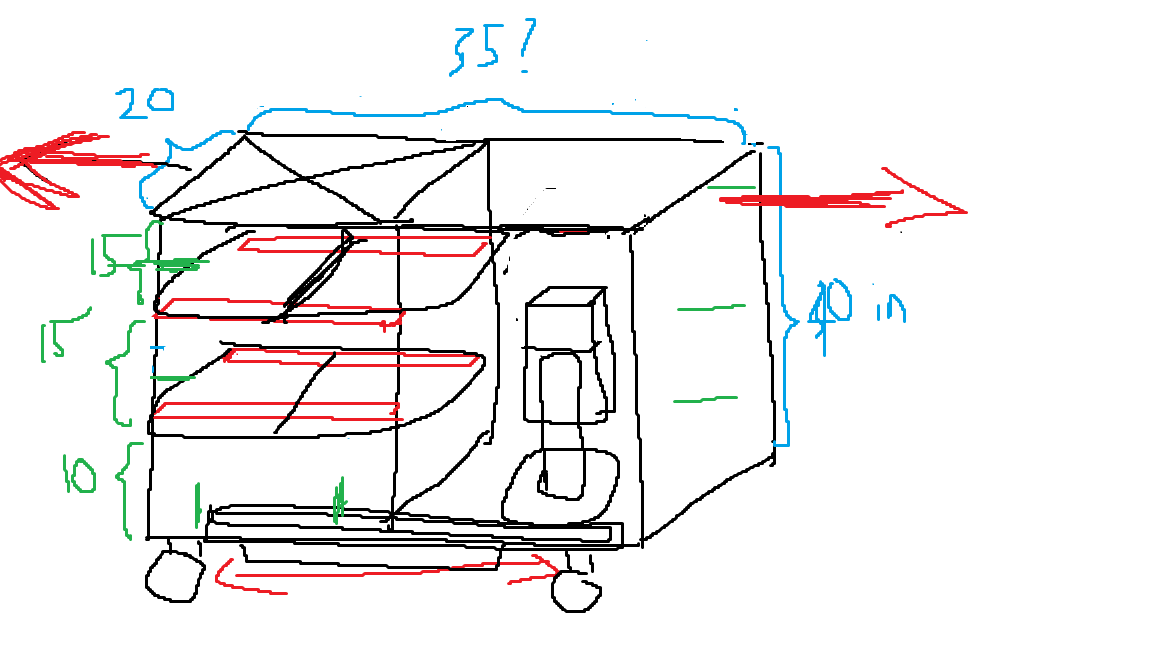
**Storage Sub-System Mathematical Formulation**

In this part, we are going to derive the mathematical formulation of the storage system based on the initial sketches of the sub-system below:



Now, due to the nature of the storage system, we can infer that a discrete space system would be a better way to represent our sub-system. Note that this discrete space and action represent a quantitative action and movement in real life. With that being said, here are the formulation:

1. **Action (Input)**
2. Tray Detection or Confirmation: This input should come from either:
   1. The lifter sub-system sending an output signal saying the tray has been released
   2. The storage sub-system detecting the tray (say by weight detecting sensor at the designated tray placement location)

This can be in the form of:

1. Cart Clearance: This input would be a direct user input indicating that the full storage cart has been cleared and is ready to be deployed again. This can come in the physical form of a button and mathematically defined as:

Thus, our action will have the form and have the action size .

1. **States**
2. Tray Storage Count: Assuming that we can store up to 4 trays at the most, this state value will have the following values:

The value for this state will indicate how many trays we have in the storage.

1. Non-Operational Indicator: This would be the state that would indicate that the storage system is not ready for operation. This could mean that the robot is moving, or tray placement is in progress.

In this case, 0 indicates that the storage is not operating and 1 indicates that the storage sub-system is ready to place the next tray.

Thus, our state would have the form of and the state space size of .

1. **Transition Probabilities**

Due to the nature of the robot, the transition probabilities will always take the value of either 1 or 0. In general, the transition probabilities will have the form of:

To illustrate how this will work, let’s give an example. Assume that we have an empty storage, and the system is ready to start collecting trays. Without any changes in the initial input here, the state will never change. In other words, we can generally define that:

A tray is placed on top of the designated location ().

Translating the above equation to English, we are saying that if a tray is placed on the designated initial position, we will update the number of trays inside the storage. This pattern is true up to when the 4th tray is placed.

Now, notice that in the last statement, . The reason that we are doing this is to indicate that the sub-system should not be operating. In other words, no matter what is, the state should not change value. This is just a special for the situation where . In general:

Where and can be any value within its respective space.

Now, let’s talk about From the previous part, we have mentioned that this input indicates that the storage is cleared by the user. In other words, it resets the state of the sub-system. In general, we can say that:

Translating this to English, for any current state we are in, if , then the state will reset the number of tray () to 0 and prepare the sub-system for operations.

1. **Output**

This needs to be figured out

1. **Mathematical Formulation to Simulation**

In this section, we will talk about the rough concept of translating the mathematical formulation into a simulation. First of all, we know that will not have any physical effect in the simulation as it resets the state of the sub-system, which is done internally. We also know that this is true for , which indicates that nothing should be physically moving. With that being said, let’s take a look at a transition:

In this case, both and are 0 and 1 respectively. Thus, the main algorithm would be something like if and , then the vertical lift tower would contract to the bottom most level, and then move to the left-most spot in that level, dropping the tray at the designated 3rd position storage. Then, the vertical tower will go right then up again to the initial position. Of course, all this motion can be parametrized by the dimension of the robot.