Project Deliverable D

Conceptual Design

Marie Levin, Leila Smaili, Rafiullah Hamdard, Hanna Paik, and

Yifei Li

February 12, 2023

Abstract

This document contains Project Deliverable D. The purpose of this document is to show the Conceptual Design of the product. A set of conceptual designs were developed related to the problem statement, based on previous user/technical benchmarking and the list of prioritized design criteria. Moreover, these design concepts were discussed, analyzed, and evaluated in the team and the concept of the combination was chosen.

Table Of Contents

1.0 Introduction

Robert Ritchie has tasked our design team to create a product that displays the specific conveyor speeds that optimize the yield of beer from his manufacturing line. Following the initial client meeting, our team created the following problem statement.

“A need exists for Robert Ritchie and his fellow supervisors to find the optimal speeds of their beer packaging process to ensure it is “always at top efficiency (Project Background, Brightspace)”. The solution must have an attractive and straightforward interface that uses a flexible algorithm based on the V-Curve Theory to report optimized speeds of each unit and make recommendations on how to achieve them.”

After developing design criteria and metrics that will be used to measure our prototype’s ability to solve our client’s problem, each team member has created a basic design for each subsystem.

This report documents each member's design concept, subsystem concepts and the final conceptual design which combines all ideas.

2.0 Subsystems

1. User interface

This subsystem is the part of the software that interacts with the user. It consists of the code that displays the application to the user. It will ask for the filler speed input and show the output to the user. It should be possible to run multiple interfaces for different simulations. The user interface should also be able to store previous simulations. The design must be intuitive for the client.

2. Algorithms

This subsystem will consist of the functions that will take the filler speed input and compute the output values to be displayed on the user interface. The V-curve theory will need to be utilized to compute the conveyor speeds and other functions will calculate the optimal yield. This algorithm will need to be highly customizable to meet the client’s need for this design to be future proof.

Here is the logic of algorithms in diagram.

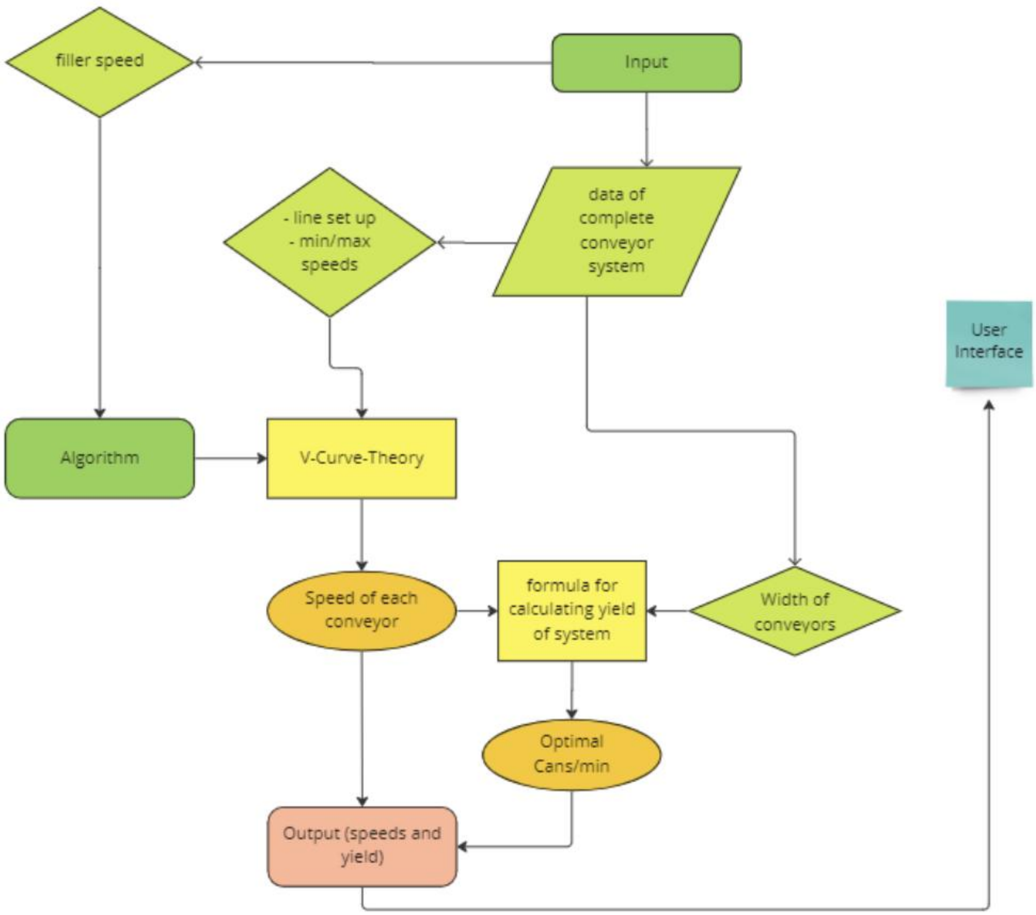


FIgure 1. Logic of Algorithms

3. Hardware

This subsystem will be responsible for running the software, it must be a device that is secure from privacy breaches. This is a vague definition as a variety of devices are being compared. Depending on the device chosen the type of software used to build the application will change.

3.0 subsystem concepts

User interface:

Table 1 Subsystem concepts

This table outlines basic designs for each subsystem.

Subsystems		
User interface	**Algorithm (The math function used to compute the output cannot be changed but the language to write the functions will)**	**Hardware**
Textually	Octave (matlab) Math-based/matrix coding language	Windows computer (display and keyboard)
Graphically	C++, java or python (Object oriented programming language)	Raspberry pi device Device purely for this application (LCD display with button pad and switches)
Visually	Excel formulas	Android phone/tablet (handheld touch screen)

3.1 Interfaces

Three possible interfaces will be provided in order to meet different needs and demands.

3.1.1 Textually displayed interface

This interface will list all data out like excel. Users are able to see data clearly. However, users might take longer to get familiar with the system display than other two visual interfaces.

Figure 1 is an example of the list on a textually displayed interface. Users are able to edit and simulate the data (Figure 2).

Line A:	
Current speed: XX	Optimal Field: XX
Line B:	
Current speed:XX	Optimal Field: XX

Figure 1. Textually displayed interface

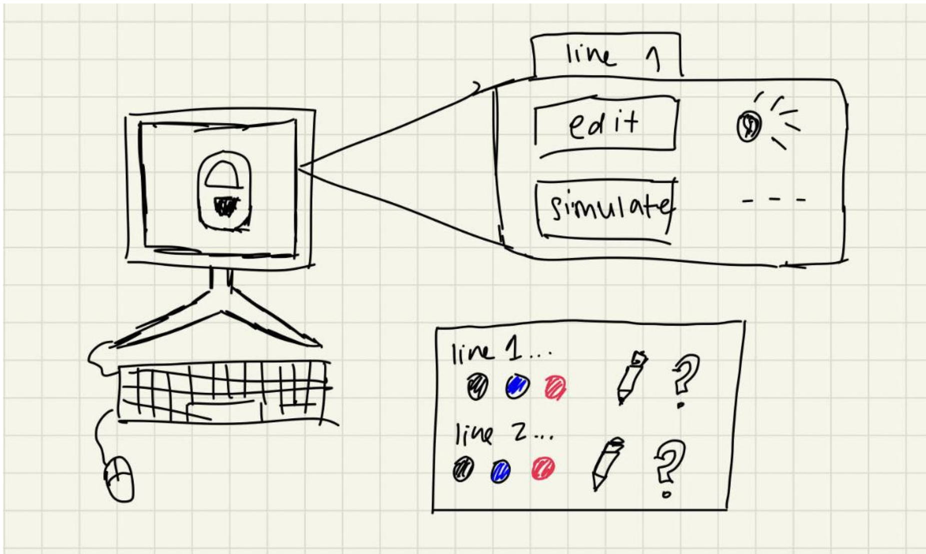


Figure 2. Edit and Simulate Button

- Input: Filler speed
- Outputs: Current speed of the conveyors, optimal yield (cans/min)

3.1.2 Graphically displayed interface

This interface is designed based on the idea that optimal speeds will be calculated according to V-curve Theory (Figure 2), so that users can easily have a graphical view of the relationships between speeds of each belt. However, the data and graph might be hard for users who lack knowledge of V-curve theory.

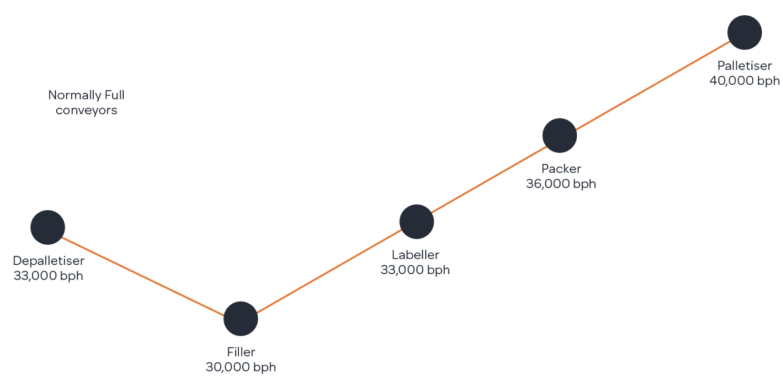


Figure 3. V-curve of a typical product line (LineView Admin, 2022).

A graphically displayed interface (Figure 4) is designed based on the trend of the V-curve theory.

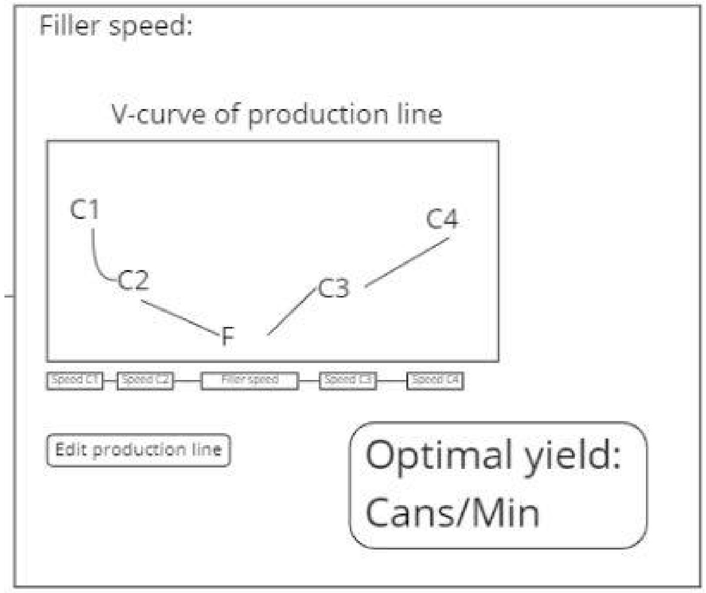


Figure 4. Graph of v-curve theory interface

- Input: Filler speed
- Output: Current speed of the conveyors, optimal yield (cans/min)

The current speed is able to be edited by users. Thus the optimal yield will also be changed based on the new current speed.

3.1.3 Visually displayed interface

This interface is designed based on the visual view of the conveyors, so that users can easily recognize the system. However, the interface might look too overwhelmed since there are a bunch of icons on.

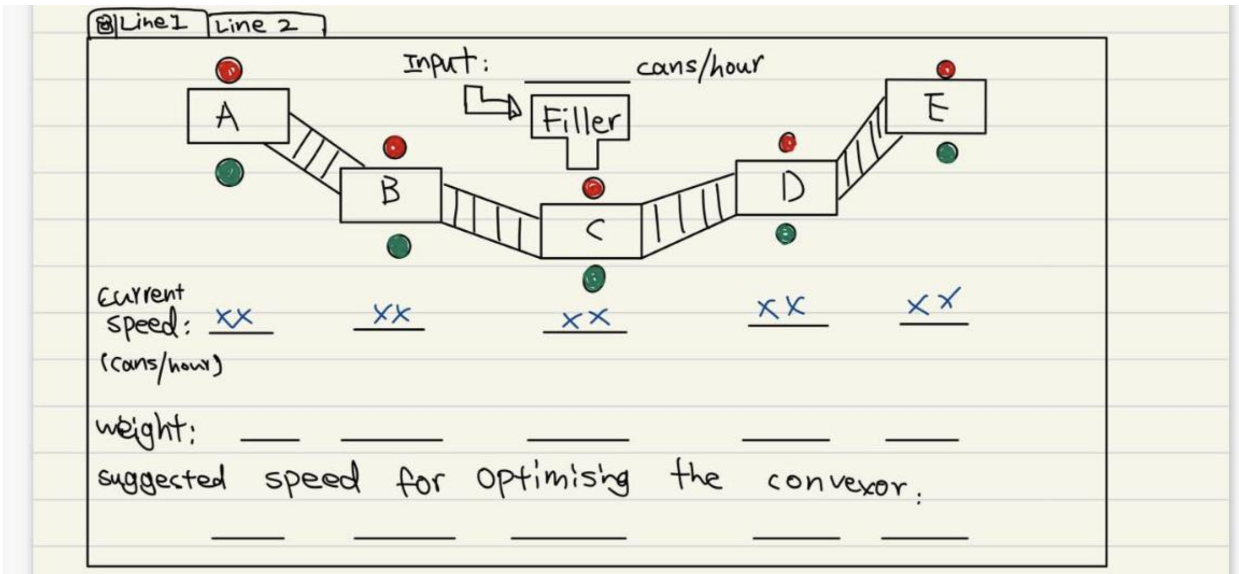


Figure 5. Visually displayed interface

- Input: Filler speed
- Output: current speeds, weight of the conveyors, optimal field.

Red and green lights are available for reminding users of the conditions of each belt.

3.2 Algorithm Language

Algorithms are the major part of this project, and the final function is constant throughout the whole project. In order to suit different demands and requirements, three different coding language choices will be provided to users.

3.2.1 Octave

Octave is a programming language used for solving math problems. It perfectly fits the project.

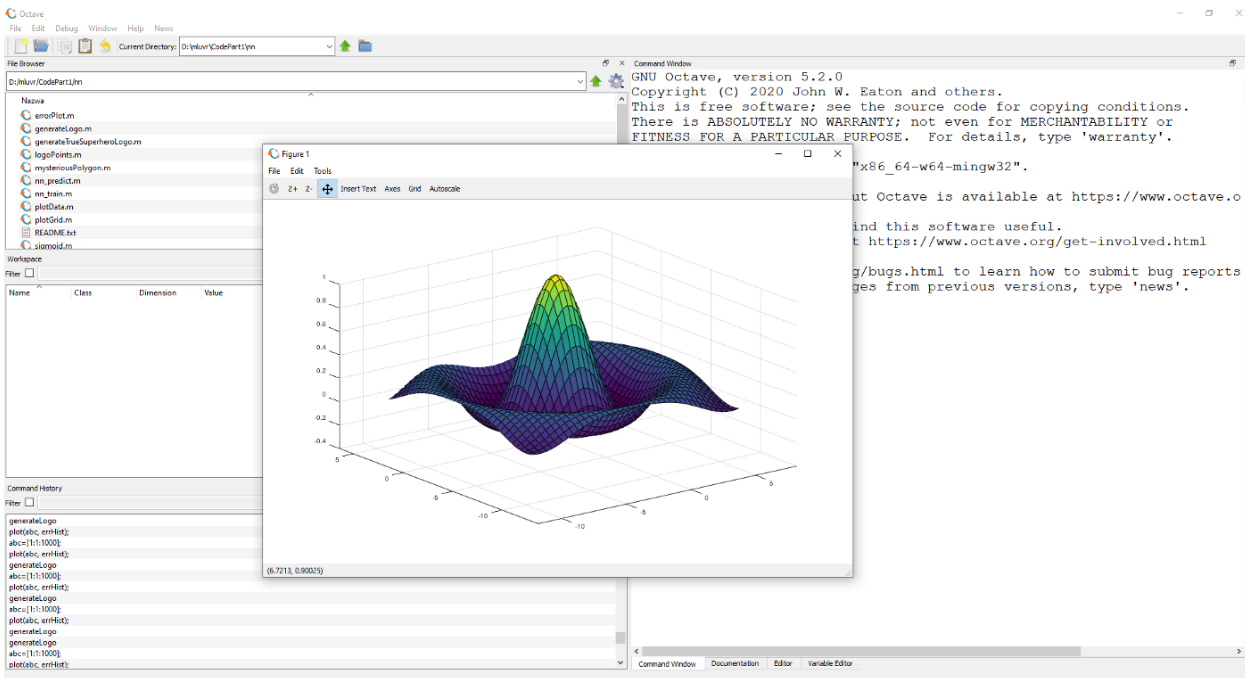


Figure 6. Screenshot of Octave (Gebel, 2020).

Octave has built-in visualization tools, easy-understanding logic and syntax. However, the application executes relatively slower than other alternative applications.

3.2.2 C++, java or python

Object oriented programming language is a typical method of creating interfaces. Majority of programmers are concentrated in these three programming languages. However, Object oriented programming languages are time-consuming for beginners.

3.2.3 Excel formulas

Excel is the most basic tool to compute functions. It is easy to handle and very stable. It has an advantage in showing digits in lists. However, it lacks control and security.

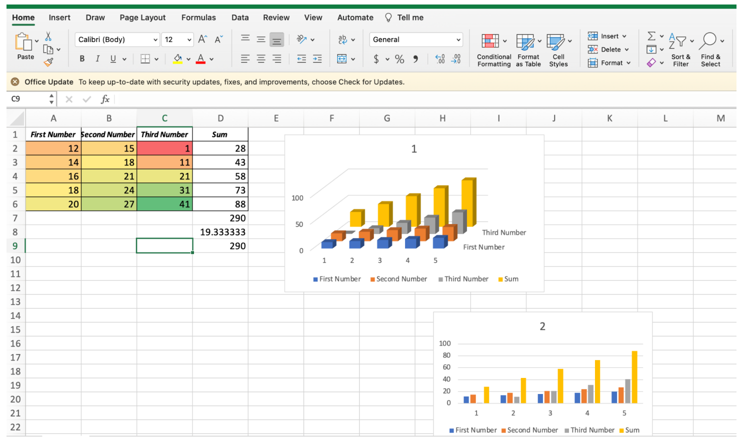


Figure 7. screenshot of Excel

3.3 Hardware

Three possible choices of hardware devices will be provided to users in order to fit different requirements and needs.

3.3.1 Windows computer

Windows is the most popular operating system in the world. After downloading the application into users' computers (only available for windows computer), users can access the interface with a mouse and keyboard, but NO touchable screen is available.

3.3.2 Raspberry pi device

Users might be looking for individually portable devices. In this case, a team-made device will be provided to users. Figure 7 is a component of the devices, switches, LCD display will be added after consulting the user needs. However, it relatively costs more than other possible alternative devices (Purchasing new components).

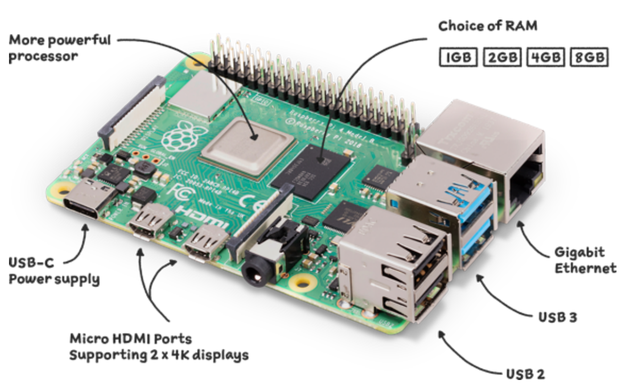


Figure 6. Raspberry pi 4 model B (Raspberry Pi, 2023).

3.3.2 Android phone/tablet

After downloading the application into users' phone/tablet, users are able to access the interface through their Android phone/tablet (only available for Android Phone/tablet). Touch screen is available. However, no external connecting equipment allowed (mouse/keyboard).

4.0 Concept Design

Three possible concept Designs.

Concept 1:
The first concept our team developed would use Octave, a free alternative to matlab, to create a desktop application. The user interface would display the production line information in a table format as seen in 3.1.1 . The pros to such an application is that it can run offline and locally on a windows computer and Octave would be useful for writing our computing algorithm in a matrix format. The cons of this application is that it would not be so intuitive as the display would be a series of numbers in tables that may be hard to understand at first glance. This design also assumes the client has a secure workplace computer to run this application on.

Concept 2:
Our team would use Python for our second concept. As a general-purpose programming language, Python is mainly used to create applications. The interface will stick to a V-Curve Theory diagram as seen in 3.1.2, so that users are able to see the speed relationship among different belts. This is easier for users to predict the entire change without actually simulating or changing the speed. However, users who have no knowledge in V-curve Theory might need to take extra time to figure out the meaning of the V-Curve graph in order to utilize the graph efficiently. In addition, this application will be run on a team-made raspberry pi device. Users are able to customize their own devices based on their needs. However, since the entire device is new, its relative costs will be more expensive than other possible alternative devices (Purchasing new components).

Concept 3:
Our third concept would be an android application created with the android development software. The algorithm would have to be implemented using Java, an object oriented programming language. The mobile application will display a visual diagram of the conveyor system and display the recommended speeds of each conveyor as seen in 3.1.3.. This concept will also take input on the real speeds of the conveyors and indicate which conveyor is not optimal. The pros of this concept is that the hardware would be a handheld device that would be used only to run our software. Cons of this device is that the android table would add extra cost for the company but the software can be developed by us for free as a proof of concept. The visual display of the conveyor system is very intuitive, but creating a diagram that can be customizable for different production lines may be complicated.

4.1 Official Concept Design

After evaluating the pros and cons of each subsystem and derived concepts we concluded on the following elements to create our design solution:

4.1.0 User interface

Our team has chosen a combination of the V-curve graphical display and the visual representation of the conveyor line. We believe this would be the most intuitive way of seeing the relationship between the speeds of each conveyor and visualizing the conveyors themselves. The customizability of the algorithm will be displayed upon clicking the edit button as a table view if the production line and properties of each conveyor will be displayed in a different window to be edited. The interface should also have different tabs that simulate different production lines. Octave will be useful for this as it has a wide variety of display tools to create an appealing diagram of the production line.

4.1.1 Algorithms

The process of the algorithm is shown in a flowchart as seen in the figure below. Upon receiving the input from the user interface the algorithm should use the V-Curve theory to calculate the recommended speed of each conveyor based on the filler speed. The algorithm will also check if the optimal speeds are within the bounds of each conveyor's max speeds and if given realtime conveyor speeds the algorithm should compare if the conveyors are too fast or too slow for the desired yield. The Algorithm will be coded using Octave. Octave is a very adaptable language as it can be used to create both an android or desktop application. It is also a more intuitive code for creating optimizing algorithms as it is a math oriented programming language that uses matrices.

4.1.2 Hardware

Our team has chosen Windows computers as the hardware because the majority of clients' equipment have Windows OS. Users are able to access our interface after downloading the application to their Windows equipment (computer). There is no extra need for users to purchase new devices. In addition, users are able to connect external equipment to the computer, so that the devices can be modified. However, Windows computers usually have poor security compared to other alternative OS computers. Users need to especially pay attention to virus susceptibility. In addition, It is possible for us to change to Android devices based on user needs.

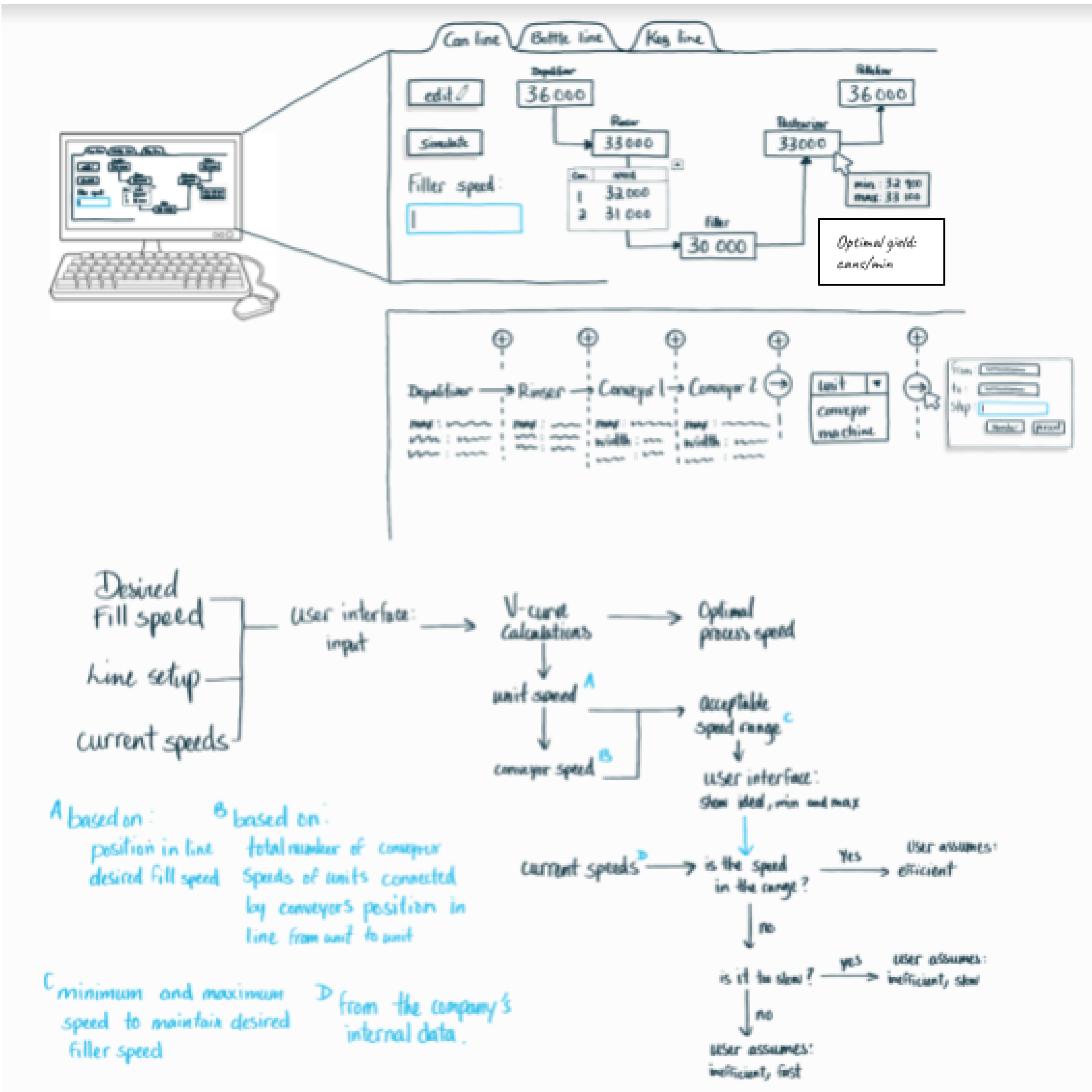


Figure 7. Official Concept Design

Priority	Design Specifications	Relation =,<, or >	Value	Units	Should our concept perform?
Functional Requirements					
5	Calculate the optimised yield (within a margin of error TDB)	=	TBD	Kegs/h and Bottles/ min	Yes
5	Easy to use(number of actions to achieve desired result)	<=	2	Number	Yes
5	Display the calculated optimal speeds of each unit	=	TBD	Kegs/h and Bottles/ min	Yes
4	Identify units which are not running at the optimised rates	=	TBD	Kegs/h and Bottles/ min	Yes
4	Can implement any machinery involved and any order of units in the process.	=	Yes	N/A	Yes
Non- Functional Requirements					
5	Safety (data provided is proprietary information and thus must be stored securely)	=	Yes	N/A	Depends on security of workplace computer (maybe)
4	Convenient (small handheld device or desktop app)	=	Yes	N/ A	Could be android (maybe)
3	Aesthetically pleasing user interface	=	Yes	N/A	yes
Constraints					
-	Cost	<=	100	$	Yes
-	Recommendations cannot exceed the max speed of each conveyor	<	Max speed	Kegs/h, Bottles/ min, Cans/min	Yes
-	Algorithm needs to be based off of the speed of the filler station	=	Input	Filler speed	Yes

Priority	Design Specifications	Relation =,<, or >	Value	Units	Should our concept perform?
Functional Requirements					
5	Calculate the optimised yield (within a margin of error TDB)	=	TBD	Kegs/h and Bottles/ min	Yes
5	Easy to use(number of actions to achieve desired result)	<=	2	Number	Yes
5	Display the calculated optimal speeds of each unit	=	TBD	Kegs/h and Bottles/ min	Yes
-	Recommendations are based off the data provided by the client	=	Yes	N/A	Could be implemented

Wrike Snapshot

https://www.wrike.com/frontend/ganttchart/index.html?snapshotId=lg28phhtWdRs3ZKibxK3Z7wCjbiTMBYu%7CIE2DSNZVHA2DELSTGIYA

4.0 References

Gebel, L. (2020). Towards data science. Retrieved February 12, 2023, from https://towardsdatascience.com/octave-schttps://towardsdatascience.com/octave-scientific-programming-language-crash-course-2ab8d864a01dientific-programming-language-crash-course-2ab8d864a01d

LineView Admin. (2022). *Is the V-curve the best option for setting accumulation line speed?* Blog, News & Events - LineView Solutions. Retrieved February 12, 2023, from https://news.lineview.com/is-the-v-curve-theory-the-best-option-for-setting-accumulation-line-speed

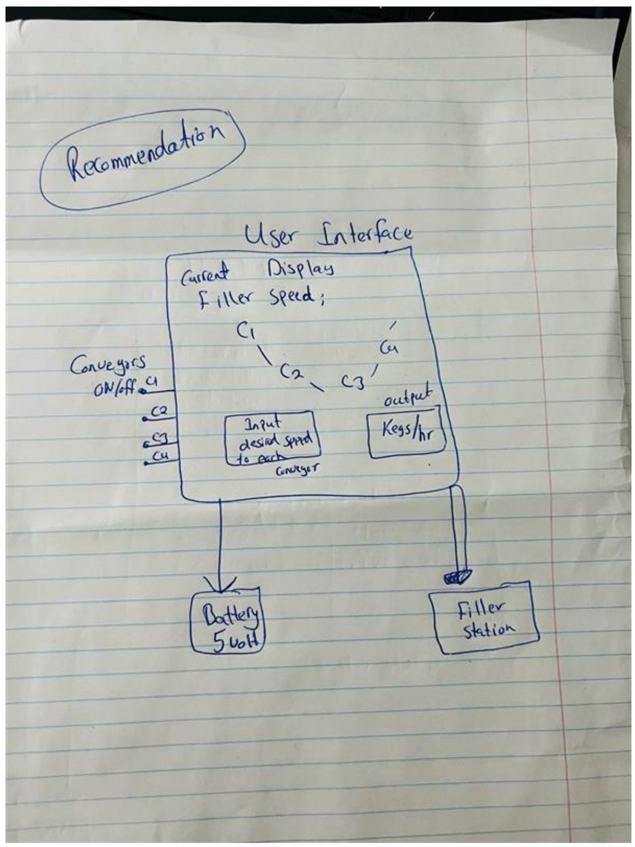
Raspberry Pi. (2023). *Buy A raspberry pi 4 model B*. Raspberry Pi. Retrieved February 12, 2023, from https://www.raspberrypi.com/products/raspberry-pi-4-model-b/

Appendix A

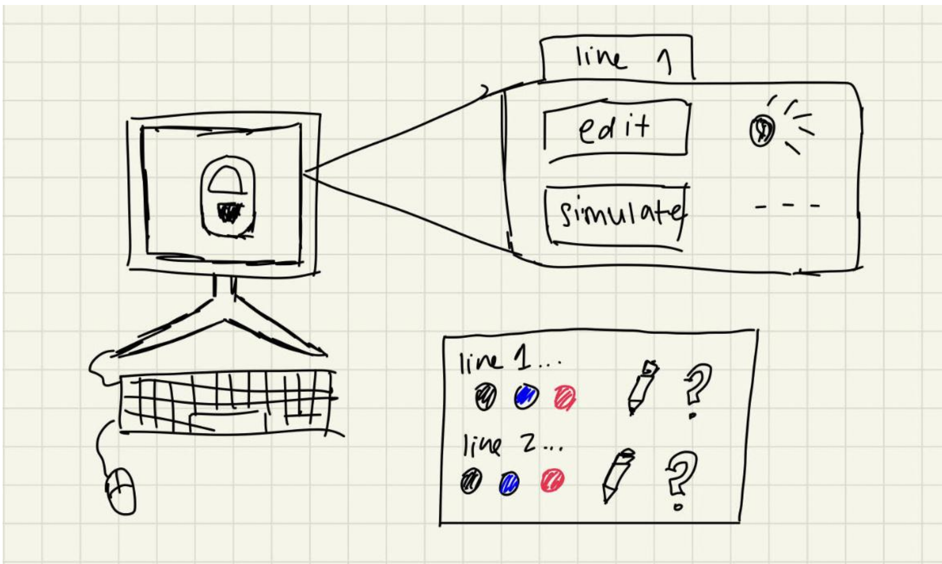
Our individual Design Concepts

Rafiullah:

- V-curve display
- Raspberry pi device connected to battery
- Switches
- Connected directly to conveyors with sensors that can be shut on/off

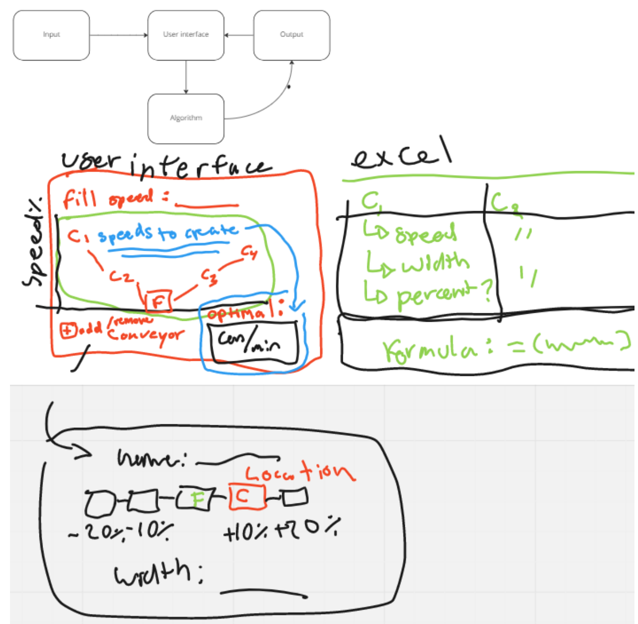


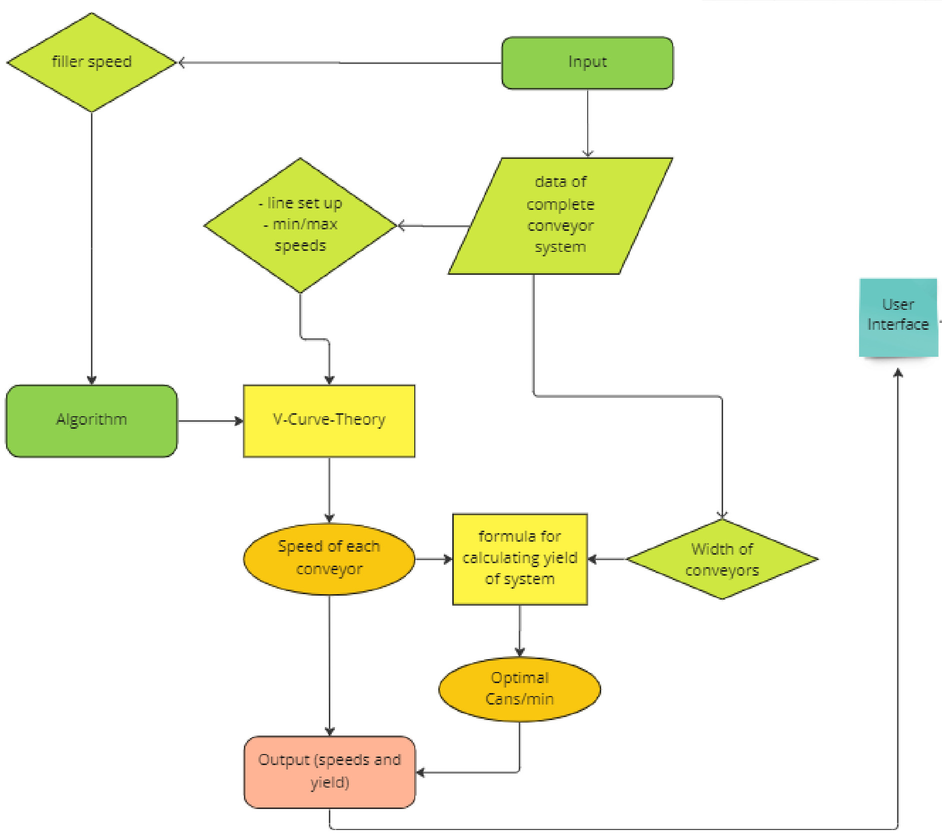
Marie Levin:

- Security
- Windows computer
- Simulation possible
- Current speed can be edit
- Multiple simulations
- 

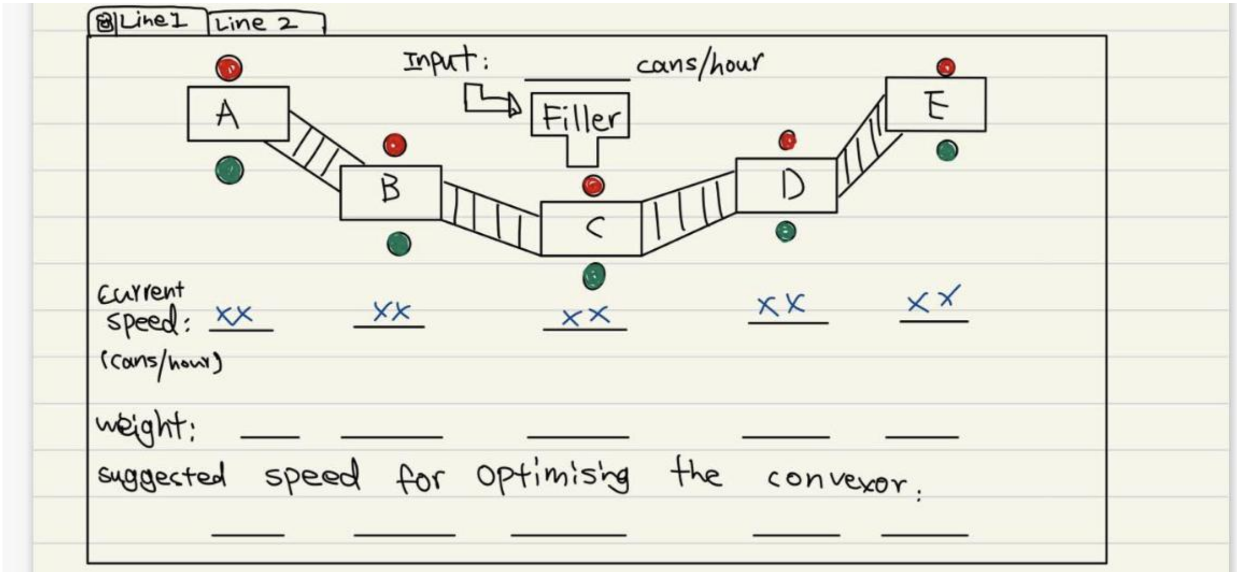
Hanna Paik:

- Flowchart of algorithm
- V-curve display of conveyor speeds
- Displays optimal yield
- Adjust algorithm with separate window
- Uses excel for calculations



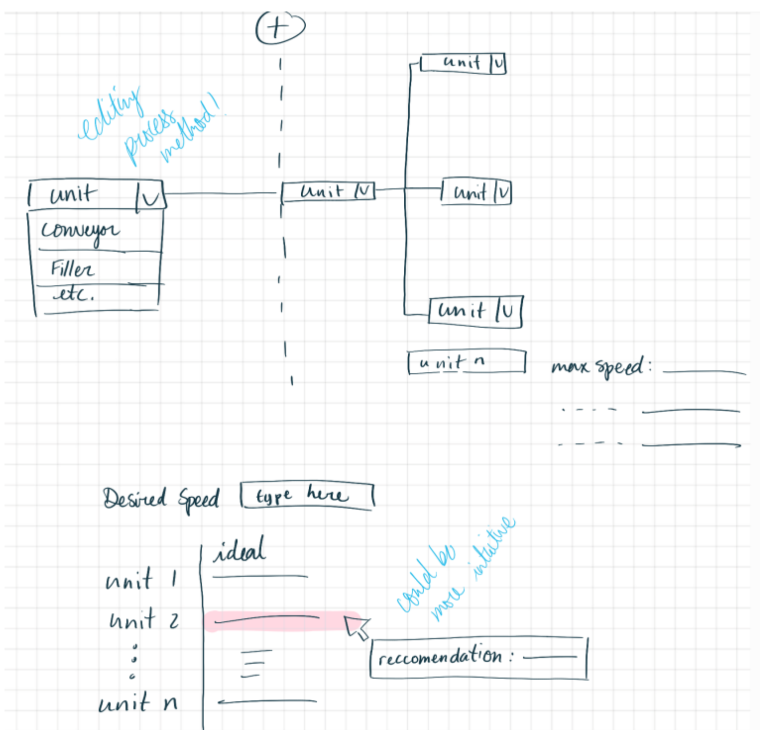


Yifei Li:

- Red/Green light indicate system situation
- Conveyor graph
- 

Leila:

- Table view of information
- Editable algorithm



o