Optimization of a Customized Mixed Model Assembly Using MATLAB/Simulink

R.B Kuriakose¹, H.J Vermaak²

¹,² Department of Electrical, Electronics and Computer Systems, Central University of Technology, Bloemfontein, Free State, 9301, South Africa,

Abstract. The relevance of assembly line balancing has never been more critical as it is now with the advent of Industry 4.0. This paper considers the case study of a Mixed Model Assembly Line in the form of a water bottling plant. The paper initially discusses the design consideration of the water bottling plant and how it was modelled in Simulink. Next the paper chooses an appropriate optimization technique and runs it on the model with real-time inputs. Finally, an analysis is done to test the impact of optimization and if it has improved the production time of the water bottling plant. The results of the study aim to add knowledge to the field of Mixed Model Assembly line balancing.

1. Introduction

Traditionally, industries have focused on producing limited variety of products. This is done by stocking raw materials ahead of demand and shipping finished products in accordance with demand. This is termed make-to-stock approach [1]. The risk in make-to-stock approach occurs when the product number is high and demand is stochastic [2]. In order to overcome the obvious disadvantages of make-to-stock approach, most industries now opt for the make-to-order approach [3]. A make-to-order assembly line is one which has the ability to fulfil [4] customer orders quickly as well as offer a variety of products. The major advantages of such an approach is that it eliminates finished inventories that may remain unsold, reduces financial risk and increases product variety [4].

Product variety [5] can be included to a product at various phases of production. It can be included [6] in the design, fabrication, assembly, sales or use phase of product manufacture. The assembly line however, provides one of the most cost effective approaches to high product variety [5]. However, as product variety increases, due to the shift from mass production to mass customization [6], assembly lines must also be designed to adapt accordingly and operate to meet the demands of product variety.

Product variety can be introduced to a manufacturing scene by using mixed model assembly lines. They can produce many different models of a product [7]–[10] on the same assembly line without changing the structure of the assembly line. However, mixed model assembly lines are complex in nature and contribute to the problem of assembly line balancing [11]. The challenge becomes exacerbated when considering a system which has customized inputs and requires real time optimization.

Real time optimization is critical for the successful planning of a diverse assembly line [12]. It would need to consider factors like production rate, production time, resource availability and stock update.
[13] among others to ensure quick decisions to aid a customized [14] approach. This is particularly significant in light of recent developments in the area of SMART manufacturing and Industry 4.0 [15].

This paper aims to conduct a real time optimization to reduce the production time by implementing a Simulink model that was designed to replicate the working of a water bottling plant. The water bottling plant, henceforward referred to as the plant, is taken as a case study for a customized mixed model assembly line. The plant can produce 500 ml and 750 ml bottles of water to the requirement specified by the users. The paper is structured such that initially the focus will be on the overview of the model and its design. Secondly, the focus will be on the optimization of the model and finally the results of the optimization, in the form of simulated graphs, will be discussed in detail.

2. Model Overview
A 3-D print model of the plant once completed is shown in Figure 1. As depicted in Figure 1, there are three major units; the source and tank unit (A), the bottle manufacturing and storage unit (B) and the water filling unit (C). As mentioned in the introduction, the aim of the plant is to bottle 500 ml and 750 ml bottles as per user defined input requirements.

![Figure 1. Top view of 3-D printed model of the completed plant](image-url)
A Simulink model was designed in order to replicate the 3D model in Figure 1. This is depicted in Figure 2. As seen from Figure 2, all units described in Figure 1 has been included as subsystems in the Simulink model. This section aims to explain each subsystem. This will form a foundation for the discussions on the model optimization in the next section.

2.1. Source subsystem
This is a masked subsystem which, as the name suggests, acts as the source of the purified water which will be used to fill the 500ml and 750 ml bottles. The subsystem consists of a constant block which defines the flow rate of the water, a saturation block, to control the upper and lower limits of the flow rate and an output block. The flow rate, defined as \( m_{source}(t) \), is a critical component in optimization as it acts a handle which can be varied to regulate the production time with respect to the constraints.

2.2. Tank with pump subsystem
This subsystem has continuous states, as it takes input from the source subsystem and stores it in the tank and pumps it out to the water filling subsystem. Continuous states refer to those variables whose value is determined by the numerical integration of its derivative with respect to time. This is described in Equation (1).

\[
m_{net}(t) = m_{source}(t) - m_{pump}(t)
\]  

(1)

The net volume of water in the tank is obtained by integrating the net water flow, this is described in Equation (2).

\[
m_{net}(t) = m_{net}(t) - u(t)
\]  

(2)

Here,

\( m_{net}(t) = \text{Net water stored in tank} \)
\( m_{source}(t) = \text{Water coming from source} \)
\( m_{pump}(t) = \text{Water coming from pump} \)
The amount of water remaining in the tank is calculated as a percentage. This is described in Equation (3) as the tank level. The tank level is a critical factor in optimization as it acts as one of the constraints and should not go below zero for the ideal functioning of the plant.

\[ Tank \ level \ = \ \frac{m_{net}(t)}{V_{max} \times 1000} \times 100 \]  

Here, 

\[ V_{max} = \text{Maximum volume of tank} \]

2.3. Bottle manufacturing subsystem
The bottle manufacturing and storage subsystem are independent of the source and tank with pump subsystem. The function of the bottle manufacturing subsystem is to manufacture and distinguish between the two bottle sizes necessary as per design requirements. This is achieved in Simulink by passing a sine wave through a compare with zero block. All outputs during the positive half cycle of the sine wave are considered to be 500ml bottles, while all bottles in the negative half cycle will be considered as 750 ml bottles. The output of this subsystem is connected to a multiplexer and provide as input to the storage subsystem.

2.4. Bottle storage subsystem
The bottle storage subsystem takes the output of the bottle manufacturing subsystem. Therefore, it will have two inputs, 500ml and 750 ml bottles. This subsystem acts as the second constraint in the optimization of the model as the number of the bottles should never go below zero as this would result in the plant ceasing to function.

This subsystem has three inputs; the first is, as mentioned previously, the input from the bottle manufacturing subsystem, the second is a count of the number of bottles that have been filled and finally, the initial count of the bottles, which will be defined under a subsystem mask.

2.5. Water filling subsystem
The outputs of the tank with pump subsystem and the bottle storage subsystem act as inputs to the water filling subsystem. The number of bottles that need to be filled will be determined by the order placed by the customers. The Customer order is recorded in MATLAB script along with the required date of delivery. The water filling subsystem needs to read this data and process the request accordingly. This processed data is placed onto a one dimensional lookup table and indexed. A trigger element is setup within the subsystem to ensure that once each requirement in the index has been met, it moves to the next. A distinction between the 500 ml and 750 ml bottles are made using function within the subsystem which checks the index where the data has been read from. The 500ml bottles, by virtue of being first, will be entered in the odd rows, while the 750ml bottles will be in the even rows.

3. Model Optimization
The aim of this section is to explain how the production time of the customized order can be optimized to find the least possible time for completing the orders. Optimization is defined [19] as the process of maximizing or minimizing an objective function with or without constraints with respect to certain input variables. The basic components of an optimization problem [20] are the following;

- Objective function – This is a function which expresses, in mathematical or non-mathematical form, the model which needs to be minimized or maximized. A model can have no objective functions or multiple objective functions. The event that there are no objective functions, is often referred to as a feasibility problem while multiple objectives are often formulated as single equation with weighted combinations of the objectives.
• Variables – These are a set of values which control the value of the objective function. Like objective function, a model can have single or multiple variables. However, variables are an essential component as without variables, the objective functions and constraints cannot be defined.

• Constraints – Also referred to as boundaries or limitations that have to be adhered to when minimizing or maximizing an objective function. Constraints can be equality or non-equality.

This specific model has initially considered two constraints being firstly the water level of the tank, defined in equation 3, and secondly the number of 500 ml and 750 ml bottles available in storage. The water level in the tank should never go below 0% and should create an alert when below 25%. The number of bottles should never go below zero as this would result in the system crashing in a physical setup. The pump flow rate from the tank with pump storage system acts as the handle which can be varied to meet the constraints. The pump flow rate is defined under a mask in the tank with pump subsystem as a variable ‘x’ so that it can be varied.

The MATLAB optimization function fmincon [17] is used to optimize the plant. The reason for choosing fmincon was because the function that needs to be optimized in the case of this plant is nonlinear and multivariable [18]. The fmincon function has the following structure.

\[
\begin{align*}
\min_{x} f(x) & \text{ such that } \\
& \begin{cases} 
  c(x) \leq 0 \\
  ceq(x) = 0 \\
  A.x \leq b \\
  Aeq.x = beq \\
  lb \leq x \leq ub
\end{cases}
\end{align*}
\]

Where,
- \(b\) and \(beq\) are vectors,
- \(A\) and \(Aeq\) are matrices,
- \(c(x)\) and \(ceq(x)\) are functions that return vectors
- \(lb\) = Lower boundary
- \(ub\) = Upper boundary
- \(f(x)\) = Function that returns a scalar

The specific syntax used in fmincon to achieve this optimization is described as follows;

\[
x = \text{fmincon}(\text{fun}, x0, A, b, Aeq, Beq, lb, ub)
\]

Where,
- \(x0\) = starting point of minimization
- \(fun\) = function to be minimized
- \(b\) and \(beq\) are vectors,
- \(A\) and \(Aeq\) are matrices
- \(lb\) = lower boundary
- \(up\) = upper boundary

Here, the syntax needs \(x0\), a starting point for the minimization. This is kept at 0.1. Next it needs a lower and upper boundary for the pump flow rate, which is the handle. This will ensure that the solution is always within the range of \(lb \leq x \leq ub\). The \(lb = 0\) and the \(ub = 1\). As there are no inequality constraints \(Aeq = []\), and \(beq = []\).

Therefore, the optimization should yield a pump flow rate between 0.1 and 1.0. Ideally the pump flow rate should be closer to 1.0, but that would drain the water in tank faster and use up bottles quicker, so the pump flow rate needs to varied to ensure both constraints are within bounds.
4. Results

The aim of this section is to explain and analyze the results of the optimization. This will be done by initially looking at the customer inputs and how they have been sorted according to the required delivery date which is an option provided to the customer. This is shown in Table 1.

| Customer  | Number of 500ml bottles | Number of 750ml bottles | Required Delivery Date | Expected Delivery Date | Expected Delivery Date-Optimized |
|-----------|-------------------------|-------------------------|------------------------|------------------------|---------------------------------|
| Customer A| 150                     | 100                     | 14-Jan-19              | N/A                    | N/A                             |
| Customer B| 200                     | 250                     | 13-Jan-19              | N/A                    | N/A                             |
| Customer C| 250                     | 150                     | 16-Jan-19              | N/A                    | N/A                             |

It can be seen here that the Table 1 that there are six columns, the first for the required number of 500 ml bottles, second for the number of 750 ml bottles, third for the required delivery date. Column 4 needs to output the expected delivery date of the order prior to optimization and Column 5 should give the expected delivery date post optimization. The assumption is that the post optimization delivery date will be shorter than the pre optimization delivery date. The expected delivery date is shown in Table 2.

| Customer  | Number of 500ml bottles | Number of 750ml bottles | Required Delivery Date | Expected Delivery Date | Expected Delivery Date-Optimized |
|-----------|-------------------------|-------------------------|------------------------|------------------------|---------------------------------|
| Customer B| 200                     | 250                     | 13-Jan-19              | 12-Jan-19 12:50        | N/A                             |
| Customer A| 150                     | 100                     | 14-Jan-19              | 12-Jan-19 13:25        | N/A                             |
| Customer C| 250                     | 150                     | 16-Jan-19              | 12-Jan-19 14:10        | N/A                             |

As seen from Table 2, the expected delivery date shows the possible date the orders can be completed without focusing on the level of water in the tank and the number of bottles in storage. The next section focusses on these aspects post optimization. The first of which is the level of water in the tank. This is depicted in Figure 3.
Figure 3. Tank level and pump flow rate after optimization

Figure 3 shows the tank level and pump flow rate after optimization. As can be seen, the tank level is standing above zero and the pump flow rate is fixed at just above 0.19. This means the optimized flow rate for this specific order is 0.19. The bottle storage level at the same time is shown in Figure 4.

Figure 4. Bottle usage during optimization

As depicted in Figure 4, the second constraint which was the usage of 500 ml and 750 ml bottles never goes below the zero. This means that both constraints set initially have been successfully met by the optimization. The only question that remains to be answered is if the optimization has resulted in reduced production time for the customer orders? The final table depicting the optimized delivery date is shown in Table 3.
Table 3: Customer requirements with expected deliver dates

| Customer  | Number of 500ml bottles | Number of 750ml bottles | Required Delivery Date | Expected Delivery Date | Expected Delivery Date-Optimized |
|-----------|-------------------------|-------------------------|------------------------|------------------------|----------------------------------|
| Customer B | 200                     | 250                     | 13-Jan-19              | 12-Jan-19 12:50        | 12-Jan-19 12:15                  |
| Customer A | 150                     | 100                     | 14-Jan-19              | 12-Jan-19 13:25        | 12-Jan-19 12:26                  |
| Customer C | 250                     | 150                     | 16-Jan-19              | 12-Jan-19 14:10        | 12-Jan-19 12:45                  |

As seen in Table 3, the optimization of the model has resulted in a reduced production time of the customer orders. This is solidified by a bar graph shown in Figure 5, which shows the difference between the optimized and non-optimized production time.

![Figure 5](image.png)

Figure 5. Difference between optimized and non-optimized production time. Usage during optimization

5. Conclusion
This study was necessitated due to the shift of industries from the traditional make-to-stock approach to the make-to-order approach. The make-to-order approach presents several opportunities and advantages for end users like reduced financial risk and increased product variety. Literature shows that product variety can be included at various stages of production, but including them in the assembly line phase of production is the most cost effective. In order to implement product variety, mixed model assembly lines need to be used. Mixed model assembly lines can be designed to have customized inputs to produce multiple variants of the same product without reconfiguring the assembly line.

However, simple mixed model assembly lines pose numerous challenges to assembly line balancing and when customized inputs are added will require real time optimization. Studies show that real time optimization is critical as the world moves towards SMART manufacturing.
In this paper a customized mixed model assembly line in the form of a water bottling plant is modelled and optimized using MATLAB and Simulink. Two factors, being the level of water in the tank and the number of bottles available in the storage, are used as constraints against the pump flow rate to optimize the model. The aim being to reduce the time taken for the production of a set of customized user inputs.

Table 1 and 2 show the order being captured by the model and expected delivery date determined prior to optimization. Figure 3 and 4 show the scope output of the two constraints after optimization. The water level in the tank should not go below zero in Figure 3 and the number of bottles in the storage should not go below zero in Figure 4. Both these constraints have been achieved. Table 3 and Figure 5 show the results of the optimization and they show that optimized model is able to reduce the production time.

6. References

[1] P. Kaminsky and O. Kaya, “Combined make-to-order / make-to-stock supply chains,” IIE Transactions., vol. 41, no. 1, pp. 103–119, 2009.
[2] H. Lee, “Effective Inventory and Service Management Through Product and Process Redesign,” Operational Research, vol. 44, pp. 1515–159, 1996.
[3] N. Kumar and D. Mahto, “Assembly Line Balancing: A Review of Developments and Trends in Approach to Industrial Application,” Global Journal of Research Engineering, vol. 13, no. 2, p. 23, 2013.
[4] D. Gupta and S. Benjaafar, “Make-to-order , Make-to-stock , or Delay Product Differentiation ? – A Common Framework for Modeling and Analysis,” IIE Transactions., vol. 36, no. 612, pp. 529–546, 2004.
[5] J. Um, A. Lyons, H. K. S. Lam, T. C. E. Cheng, and C. Dominguez-pery, “Product variety management and supply chain performance : A capability perspective on their relationships and competitiveness implications,” International Journal of Production Economics, vol. 187, no. June 2016, pp. 15–26, 2017.
[6] S. J. Hu et al., “Assembly system design and operations for product variety,” CIRP Annals-Manufacturing Technology, vol. 60, no. 2, pp. 715–733, 2011.
[7] M. Razali, F. Abdul Rashid, and M. Razif Abdullah, “Mathematical Modelling of Mixed-Model Assembly Line Balancing Problem with Resources Constraints,” in International Engineering Research and Innovation Symposium (IRIS), 2016.
[8] M. Jamil and N. M. Razali, “Simulation of Assembly Line Balancing in Automotive Component Manufacturing,” Material Science Engineering, vol. 114, no. 1, 2016.
[9] G. Reginato, M. J. Anzanello, A. Kahmann, and L. Schmidt, “Mixed assembly line balancing method in scenarios with different mix of products,” Gestão & Produção, vol. 23, no. 2, pp. 294–307, 2016.
[10] X. Zhu, S. J. Hu, Y. Koren, and S. P. Marin, “Modeling of Manufacturing Complexity in Mixed-Model Assembly Lines,” Journal of Manufacturing Science and Engineering, vol. 130, no. October, pp. 1–10, 2008.

[11] A. Gharehgozl, M. Rabbani, N. Zaerpour, and J. Razmi, “A comprehensive decision-making structure for acceptance/rejection of incoming orders in make-to-order environments,” International Journal of Advanced Manufacturing Technology, vol. 39, no. 9, pp. 1016–1032, 2008.

[12] P. Cortes, L. Onieva, and J. Guadix, “Optimizing and simulating the assembly line balancing problem in a motorcycle manufacturing company: a case study,” International Journal of Production Research, vol. 48, no. 12, pp. 3637–3656, 2011.

[13] B. L. Song, W. K. Wong, J. Fan, S. F. Chan, and A. Nomenclature, “An Adjusted Recursive Operator Allocation Optimization Algorithm for Line Balancing Control,” International Journal of Applied Mathematics, vol. 36, no. 1, 2007.

[14] B. Song, W. Wong, J. Fan, and S. Chan, “Real-time intelligent optimization decision support system for line-balancing control,” in 9th World Multi-conference on Systemics, Cybernetics and Informatics, 2005.

[15] V. Roblek, M. Meško, and A. Krapež, “A Complex View of Industry 4.0,” SAGE Open Journal, vol. 6, no. 2, 2016.

[16] R. B. Kuriakose and H. J. Vermaak, “Modelling and Simulating a Mixed Model Assembly Line : A Case Study Using a Water Bottling Plant,” Presented at the 2018 4th International Conference on Computer, Communication and Control, Krabi, Thailand. March 2018.

[17] “Mathworks documentation:fmincon.” [Online]. Available: https://www.mathworks.com/help/optim/ug/fmincon.html. [Accessed: 18-Jul-2018].

[18] R. Waltz, J. Morales, J. Nocedal, and D. Orban, “An interior algorithm for nonlinear optimization that combines line search and trust region steps,” Mathematical Programming, vol. 107, no. 3, pp. 391–408, 2006.

[19] N. Kumar, ”Optimization problem and Model formulation,”Optimization Methods. [Online]. Available: http://msulaiman.org/onewebmedia/Lecture 2 mphil.pdf. [Accessed: 28-Dec-2018].

[20] N. Kumar, “Introduction and Basic Concepts -Classification of Optimization Problems,” 2006.