

# SIMULATION AND CONTROL SOFTWARE OF A BEER BOTTLING PLANT

Author: Jorge Verdeguer Villar  
Supervisor: Marcin Morawski  
Cracow University of Technology  
Academic year: 2017/2018



## **Appreciations**

*I would like to thank to my project supervisor Marcin Morawski because of his time and help. It has been indispensable for the realization of this project.*

*Also, I want to thank to Cracow University of Technology for hosting me as an Erasmus student and to my home university: Universitat Politècnica de València (Escuela Técnica Superior de Ingenieros Industriales).*

# Index

<i>Abstract – aim of the project.....</i>	5
<i>1. Introduction.....</i>	6
<i>1.1 Simulation of production processes .....</i>	6
<i>1.2 Different types of simulation software .....</i>	6
<i>1.2.1 Software for queue management: .....</i>	6
<i>1.2.2 Software for inventory simulation.....</i>	7
<i>1.3 Software used in the thesis .....</i>	8
<i>1.3.1 Proficy Machine Edition.....</i>	8
<i>1.3.2 Arena Software .....</i>	9
<i>1.4 About SCADA.....</i>	11
<i>2. The problem.....</i>	12
<i>3. The bottling plant .....</i>	13
<i>3.1 Empty bottles warehouse.....</i>	15
<i>3.2 Cleaning empty bottles station .....</i>	16
<i>3.3 Filling station .....</i>	20
<i>3.4 Labelling station .....</i>	22
<i>3.5 Putting Caps Station .....</i>	24
<i>3.6 Beer, labels and caps warehouse.....</i>	25
<i>3.7 Quality control section.....</i>	30
<i>4. Conclusions .....</i>	33
<i>4.1 Influence of cart's and conveyor's speed .....</i>	33
<i>4.2 Knowledge acquired .....</i>	37
<i>5. List of figures.....</i>	38
<i>6. Bibliography and sources.....</i>	40

# **Abstract – aim of the project**

This project is in the area of automatics. More precisely, in the field of PLC and industrial control.

Specifically, the automatic program that controls a beer bottling plant has been designed and implemented.

The aim of this control software is to manage bottles flow along the production line as well as keeping stocked all the machines involved in the process of filling with beer, labelling and putting caps to bottles.

SCADA (Supervisory Control And Data Acquisition) has been used. This means that the control software has also a friendly HMI (Human Machine Interface) that allows its user not only to be updated about the state of each part of the plant but also to change different parameters of the machines working in the production line.

The control software and the HMI have been designed using Proficy Machine Edition software in order to work in a touch screen panel made by General Electric.

Also, a simulation of the bottling plant has been designed and implemented in the Arena Software so that all the parts of the control program can be tested.

From this simulation, we can take some interesting conclusions about the mean time that takes to produce a bottle of beer and the influence of different parameters such as carts speeds or conveyor speeds on this time.

# **1. Introduction**

## **1.1 Simulation of production processes**

Simulate allows to predict the behaviour of production systems under various real or foreseeable conditions. Using the simulation, it is possible to contemplate multiple scenarios and know how the production capacity will react to different anomalous situations. It also provides the ability to analyze the possible alternatives to optimize a production plant without having to physically alter it. The impact of these modifications will be known in advance.

A simulation can be also used to test how a control program works, know its failures and fix them before using it in the reality. This will reduce the costs and time necessary to make it properly work in the real production plant. [1].

## **1.2 Different types of simulation software [2]**

Among the most common simulation software, we can find the following examples:

### **1.2.1 Software for queue management:**

- ProModel

This software is a simulator for personal computers. Almost every system can be modeled with ProModel.

Some of its advantages are:

- Ease and flexibility to create models.
- Possibility to import the layout of the plant from a CAD file or from a digital picture.

- OR Brainware Decision Tools

The aim of this software is to help to small and medium companies to study queues in their production systems in an easy way.

This program also helps the user to take the decisions in terms of optimal production batch size.

- Arena

It is a software that allows to have a deeper understanding of a system. Is formed by different modules of SIMAN language. Arena is a very versatile software.

This is the program used in this project to do the simulation.

### **1.2.2 Software for inventory simulation**

- Dynamo

It has been the predecessor of many other software. The vast majority of the dynamic systems models of the last ten years have used the language of Dynamo.

Nevertheless, as it does not run on Windows, nowadays other software is more used than Dynamo.

- Stella and I'Think

These are probably the most common systems used for inventory simulation.

Both have the same computer development but prepared for different working environments. Stella is designed for scientific and social sciences uses and I'Think is used to give support to applications of the business world.

## **1.3 Software used in the thesis**

### **1.3.1 Proficy Machine Edition**

The HMI and the software used to control the simulation were created with the education version of the program named PME (*Proficy Machine Edition*). This software is developed by *GE automation* for configuration and programming of various types of GE's devices such as: PLCs (Programmable Logic Controllers), PACs (Programmable Automation Controller), Remote I/Os, Soft PLCs, HMI panels and other. [3].

The part of the software that is used to create the HMI is very versatile.

It is possible to introduce different elements from its library that can be configured in many ways. It is possible to insert buttons that change the value of variables or configure numeric inputs in order to set values of variables.

You can also change colours of elements, text shown by any element, disable the visibility of elements or even make elements blink according to the value of any variable. It can also display values of variables.

There are also some other more complex elements that can be configured in order to show in a graphic way the value of an integer variable.

In this software also exists the possibility of plotting the value of any variable along the time. These are usually, so called, real time trends.

When creating the control program, you can use many languages such as LD (Ladder Diagram) or SFC (Sequential Function Charts) which are the ones used in this project.

In order to design the control software for the plant and the HMI, PME was installed in a computer with Windows 7 operating system. The

developed HMI application itself can be transferred to a touch screen panel with Windows CE as well as run on the development computer in part of the PME software called *viewer* so that HMI can be used in a more convenient way.

For this purpose, 12" *QuickPanel View & QuickPanel Control* shown in figure 1 was used. This is a touch screen panel also made by *GE Automation*.



Fig. 1-12" QuickPanel View & QuickPanel Control

### 1.3.2 Arena Software

The simulation controlled by the program was done in *Arena Software*. It runs in a computer connected to the touch screen panel via Ethernet.

This is a really versatile software.

Arena is a discrete entities simulation software used to simulate and optimize complex processes.

It allows to analyze how a process or systems works.

Different scenarios can be created in order to check the behaviour of the system when they occur. With this, the solution to problems without a clear cause can be solved.

With this tool, it is also possible to identify bottle necks or oversized resources. [4].

It is based on entities that follow a path set by the user. These entities pass through boxes of different types that are put along the path, representing various actions on the production line.

All the entities have to be created with a “create” box and destroyed by a “dispose” box at the end of the path.

There are “delay” boxes, that hold entities one by one for a time set by the user.

Entities can be held in “hold” boxes until a condition configured by the user is true.

It is possible to set the value of any variable to any value when an entity passes through an “assign” box.

In order to prevent that more than one entity gets into one or more boxes at the same time, you can put before the first box a “seize” box and a “release” box after the last box. When an entity passes through the “seize” box means that a resource is being used by this entity, so the resource will not be used by more entities until this entity passes through the “release” box.

In order to do packs of entities and unpack them, “batch” and “unbatch” boxes can be used.

There are also decision boxes. They choose the way that an entity will follow depending on the value of a variable.

In this software, carts and conveyors can be simulated. When configuring them, you can choose their speed, path or distance from one station to another.

In this project, the education version of Arena software was used. The difference with the full version is that the education version has some limitations.

With the education version, the simulation stops when five minutes in real time have elapsed since it started.

Also, there are limitations in terms of number of entities used. It is not possible to use more than one hundred and fifty entities.

Other limitation is the number of variables that can be connected to another program. The maximum number in the education version is fifty variables.

## **1.4 About SCADA**

SCADA, acronym for Supervisory Control And Data Acquisition, is used to make a software for computers that allows to control and supervise industrial processes. It provides feedback with devices involved in the production plant (sensors and actuators), and controls the process automatically. It provides also all the information that is generated in the production process and allows its management and intervention with a friendly HMI.

## **2. The problem**

Nowadays, with the high competition level existing in every sector of the industry, it is crucial to optimize every resource used by companies in their factories in view of reducing costs involved in production processes.

One of the most important changes that there has been in factories the last sixty years is the introduction and development of automatic systems. They allow companies not only to produce at high rate but also to lower production costs.

This is why automation has this high importance these days and is of vital importance for companies in order to survive in the actual hostile industrial environment.

The problem posed and fulfilled by this project is to design a simulation of a beer bottling plant and create a control software able to manage this industrial plant and give a feedback to the user of its HMI. Also, allow this user to change different parameters of this plant.

### 3. The bottling plant

The beer bottling plant can be divided into eight different parts:

- Empty bottles warehouse
- Beer, labels and caps warehouse
- Cleaning empty bottles station
- Filling bottles of beer station
- Labelling bottles station
- Putting caps to the bottles station
- Quality control station
- Packing station

In the simulation done in Arena software, you can find all these sections and an additional one: process state update section.

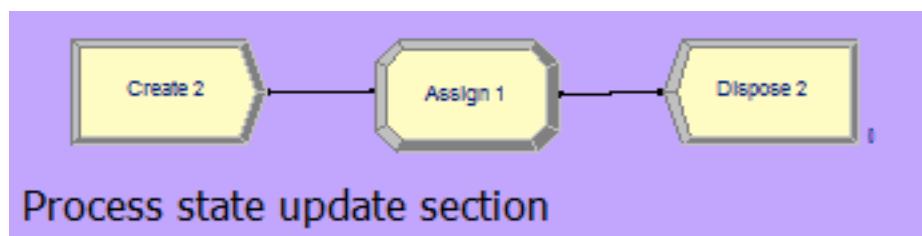


Fig. 2-Process state update section in Arena

It is formed by the three different boxes shown in figure 2. One create box, where entities are created, one dispose box, where entities are destroyed and a box between these two boxes: an assign box. This assign box is necessary in order to set the value of the following variables to their correct current value each time an entity passes through it:

- warehouse\_queue, which contains the number of packs inside the empty bottles warehouse
- cleaner1in, which contains the number of packs inside cleaner 1
- cleaner2in, which contains the number of packs inside cleaner 1
- beerlevel, which contains the current level of the beer tank

HMI, that is shown in figure 3, is divided in seven parts:

- Empty bottles warehouse control
- Cleaners control
- Beer pouring control
- Labelling control
- Putting caps control
- Quality control results
- Information of number of packs produced



Fig. 3-HMI

### 3.1 Empty bottles warehouse

When the simulation in Arena software starts, bottles are created and batched into ten bottles packs with a batch box. After being batched, they go to the empty bottles warehouse as figure 4 shows.

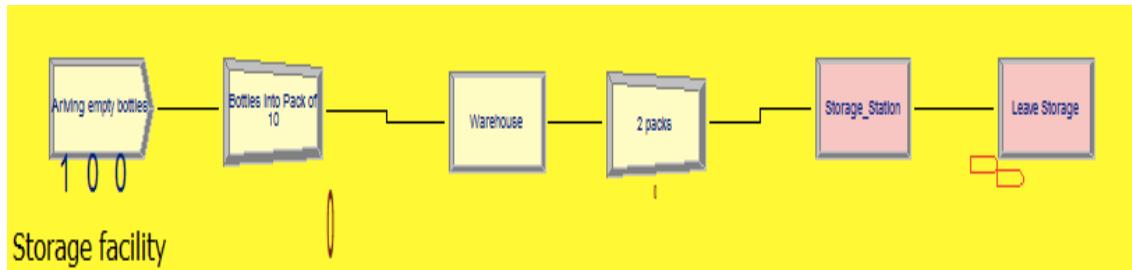


Fig. 4-Storage facility in Arena

In the empty bottles warehouse, packs of 10 bottles wait until the value of a variable called “leave\_warehouse” is changed from the HMI.

When the value of this variable is one, ten packs of bottles leave the warehouse. If this variable has the value of zero again, empty bottles packs wait in the warehouse until the value of this variable is changed again.

This variable is controlled manually by the user of the HMI with a button. If this button is pressed, the value of the variable “leave\_warehouse” changes from zero to one or from one to zero depending on its current value.

The message and colour that this button has, also change depending on the value of “leave\_warehouse”. If its zero, the message that it shows is “Stay in warehouse” and the colour of the button is red, while if the value of it is one, the colour changes to green and the message shown is “Leave warehouse”.

This mechanism was implemented just by configuring the button. When it is pressed, the value of the variable changes. When the value of the variable changes, the colour and the message shown change.

In this section of the HMI, you can also find a number which informs to the user about the number of packs waiting in the warehouse. This number shows the current value of the variable “warehouse\_queue”.

When packs leave warehouse, they are batched again into 2 packs of 10 bottles each, in order to be transported by a cart from this station to the next station: cleaning empty bottles station.

## 3.2 Cleaning empty bottles station

Here the bottles are prepared to be filled with beer.

As figure 5 shows, this beer bottling plant has two cleaning machines. Each one cleans a pack of two packs of ten empty bottles each.

It takes forty seconds to clean a ten bottles pack if this process takes place in Cleaner 1. It takes fifty seconds if the pack is cleaned in Cleaner 2.

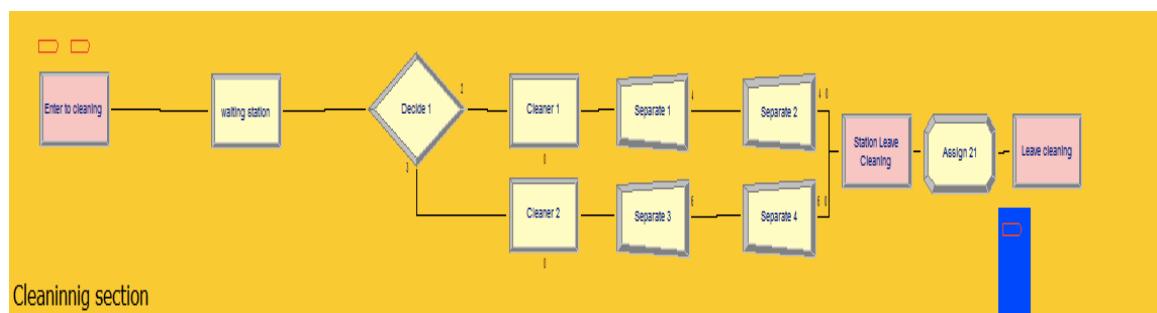


Fig. 5-Cleaning section in Arena

In order to optimize the cleaning process, they can work simultaneously.

The working mechanism is the following:

If both cleaners are available or only Cleaner 1 is available, the pack goes to Cleaner 1.

If only Cleaner 2 is available, the pack goes to Cleaner 2.

If both cleaners are not available, the pack waits in the cleaners waiting station until one of them is available.

A cleaner is available only when these two conditions are met: it is not busy cleaning a pack and it is turned on.

It is possible to turn off and on each cleaner by pressing a button in the HMI.

This algorithm was implemented by using ladder instructions.

Figure 6 is given to the reader in order to ease the understanding of the ladder program explanation:

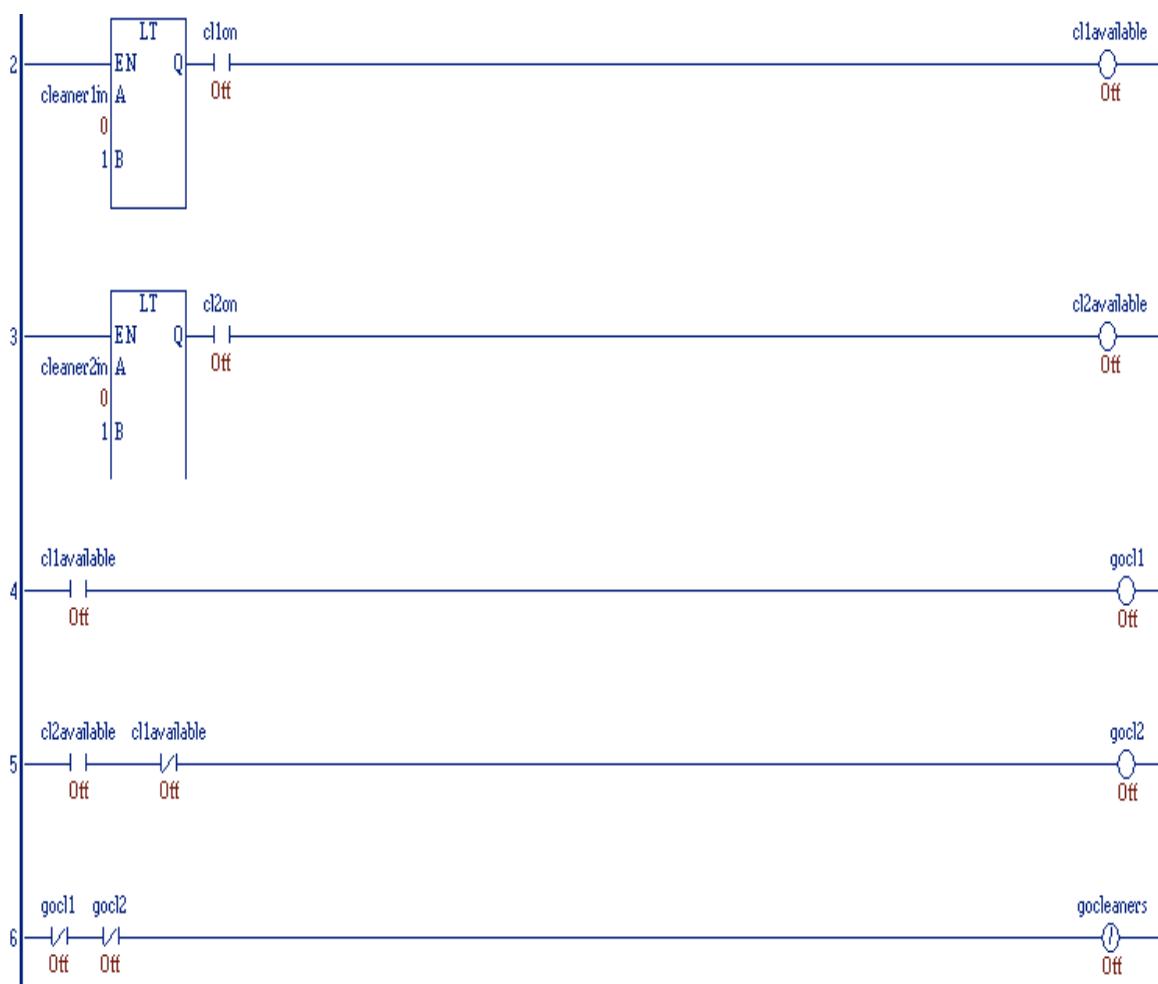


Fig. 6-Ladder program of cleaners section

Row 2:

When the number of entities inside Cleaner 1 is less than 1 (this means, when no entity is inside Cleaner 1) and the variable “cl1on” is one, the variable “cl1available” is one.

The value of the variable “cl1on” can be changed from zero to one or from one to zero by pressing a button in the HMI (toggle button).

In all the other situations, the value of the variable “cl1available” is zero.

Row 3:

The same happens with Cleaner 2, “cl2available” variable and “cl2on” variable.

Row 4:

When variable “cl1available” is one, the variable “gocl1” is one. In the other case, “gocl1” is zero.

Row 5

When variable “cl1available” is zero and “cl2available” is one, the variable “gocl2” is one. In all the other cases, it is zero.

Row 6

When both variables “gocl1” and “gocl2” have the value of zero, the variable “gocleaners” has the value of zero. In all the other cases the value of this variable is one.

This variable is the one that controls whether packs go to the cleaners or they wait in the cleaner's waiting station. When its value is one, packs go to cleaners. On the other hand, when its value is zero, they wait in the cleaners waiting station.

In the simulation done in Arena Software, this was implemented by using a hold box. When “gocleaners” equals to zero, packs wait there until this condition is not true.

In order to send the packs to de correct cleaner, a decision box was used in Arena Software. When the value of the variable “gocl1” is one, the packs that arrive to de decision box go to Cleaner 1. On the other hand, when the value of this variable is zero, they go to Cleaner 2.

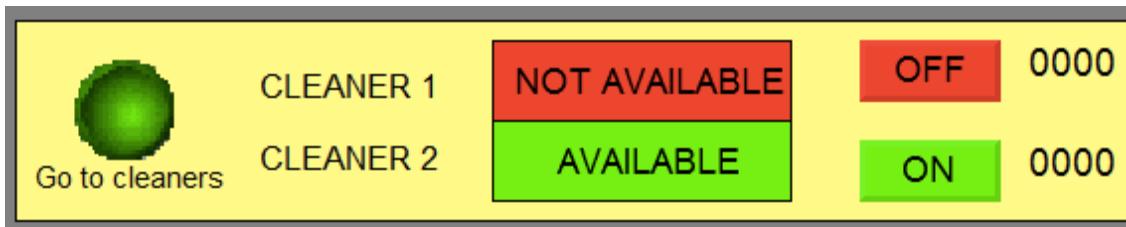


Fig. 7 Cleaners section of the HMI

The figure 3 above shows the cleaners section of the HMI. In this case, Cleaner 1 is not available because it is turned off. Cleaner 2 is available because it is turned on and it is not busy. The variable “gocleaners” is one so the light that indicates the value of this variable is green.

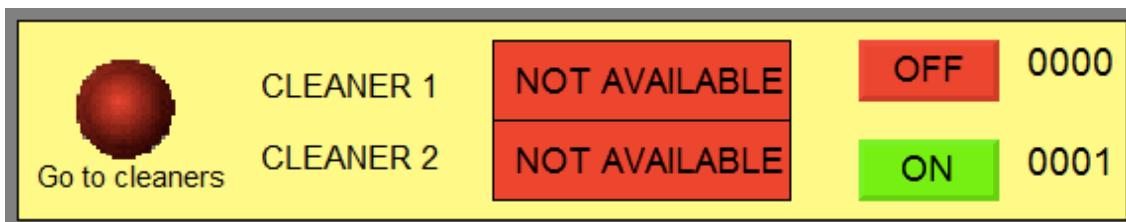


Fig. 8-Cleaners section of the HMI

In the figure 3 above, Cleaner 1 is turned off and Cleaner 2 is turned on but it is busy (there is a pack being cleaned in it, as you can see in the number next to the ON button) so it is also not available.

Because both cleaners are not available, the variable “gocleaners” has the value of zero and the light that indicates its value is red. In this case, packs would wait in the cleaners waiting station until it is green.

Once bottles are cleaned, they are ready to be filled with beer. Packs are unbatched and bottles enter in a three-meter conveyor that leads them to the filling station.

### **3.3 Filling station**

The filling station has a machine that takes beer from a one-hundred-litre tank and pours it into the bottles.

The quantity of beer poured into the bottles follows a Normal distribution with mean in 0.5 litres and a standard deviation of 0.05 litres.

This section in Arena Software is made using flow process boxes. All bottles pass through the following boxes:

- Three boxes necessary for the conveyor system
- A seize regulator box and a release regulator box.

Between these two boxes is where all the other boxes of the filling station are placed. When a bottle passes through the seize regulator box, all the other bottles are queuing in this box until the bottle passes through the release regulator box. Regulator is the resource that controls the flow.

- A flow box, where beer is poured at a rate of 5 litres (on average) per minute of the simulation.
- An assign box, that changes the image of entities from an empty bottle to a filled bottle.

Apart from these boxes, the flow system has also a tank box, where the capacity and the initial level of the tank are defined.

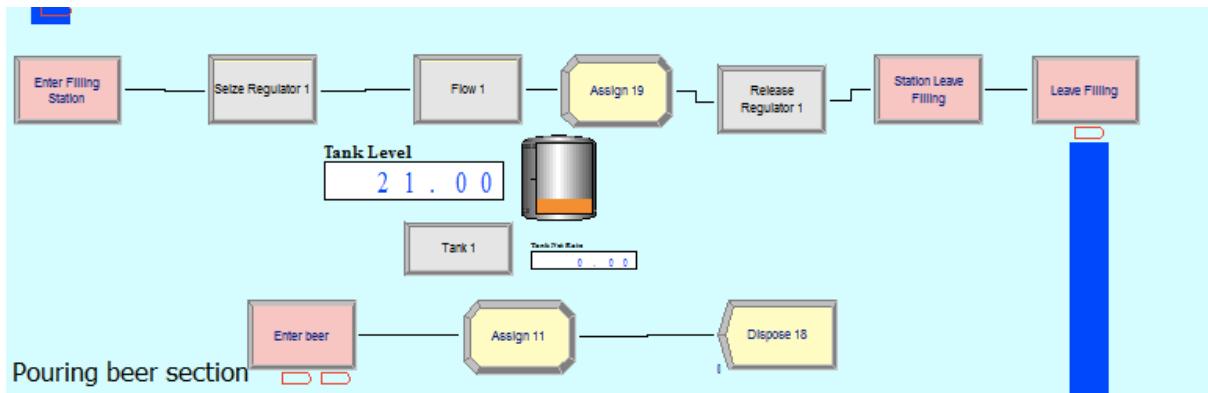


Fig. 9-Pouring beer section in Arena

The part of HMI dedicated to this section has a bar graph that shows the current level of the beer tank. It is also displayed numerically next to the graph.

If the level of the tank is over 20 litres, a message that says “BEER LEVEL OVER 20 LITRES” is displayed in a green background.

When the beer level is below 20 litres, an alarm is activated. The message changes to “BEER LEVEL UNDER 20 LITRES”. This message is on a background that changes its colour from orange to red and from red to orange every second (Fig. 4).

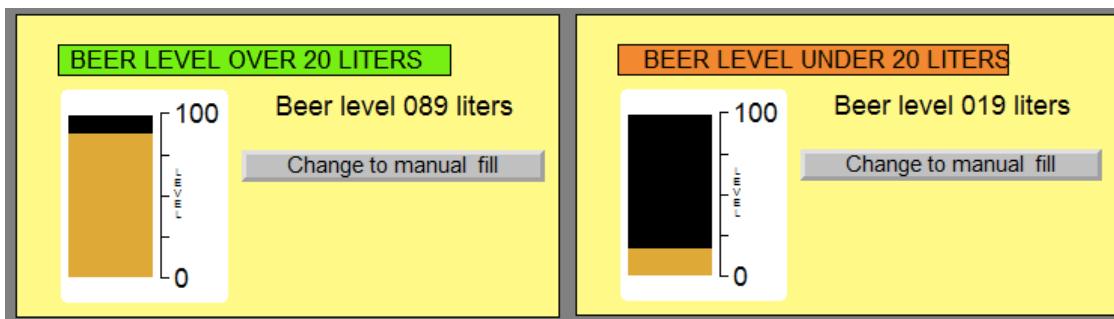


Fig. 10-Beer tank section of the HMI

After bottles are filled, they get into a four-meter conveyor that takes them to the labelling station.

### 3.4 Labelling station

Bottles are being labelled in this section of the plant.

This section in Arena Software is formed by eight boxes:

- Three boxes necessary for the conveyor system.
- A hold box, that prevents bottles from getting in the labelling machine if the number of labels left is lower than ten. This box hold bottles if the value of the variable “labelsstop” equals to one.
- A seize box.
- An assign box in which every time that a bottle is inside, the variable that contains the number of labels left in the machine decreases by one.
- A delay box, that simulates the delay that a labelling machine would cause in a production line.
- A release box.

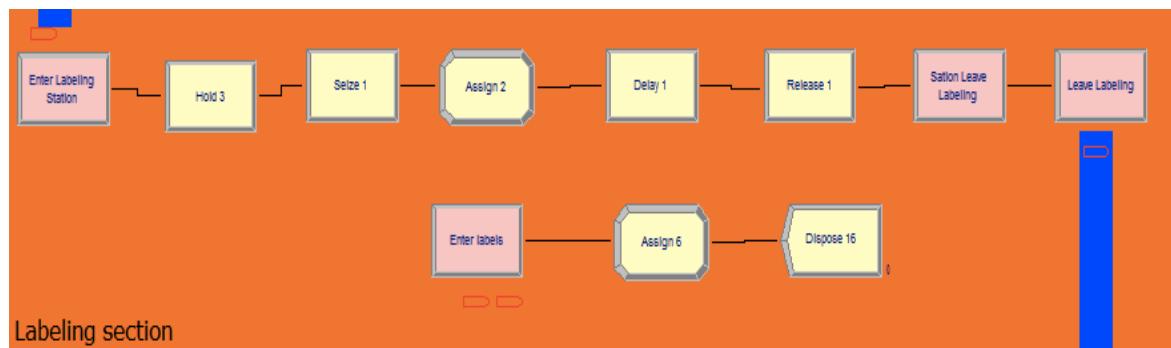


Fig. 11-Labeling section in Arena

The area of the HMI window dedicated to this section has an animated roll of labels that is being unrolled if the machine is working (Fig. 5).

This animation was done by changing from one of these pictures to the next one every seven hundred milliseconds:

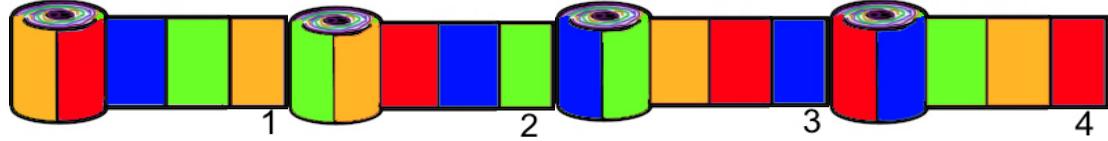


Fig. 12-Pictures of the labelling animation

This was done configuring each picture so that it is shown just when a variable for each one is activated. The variable that shows each picture is activated by an action of a four actions SFC. The transition between each action is done seven hundred milliseconds after the action was activated.

In the labelling section of the HMI, is shown also the number of labels left in the machine. If the number is lower than one hundred, an alarm similar to the beer alarm is activated.

If the number of labels left in the machine is lower than ten, the value of the variable “labelsstop” is set to one and bottles wait in the hold box before labelling process.. Also, the animation shown in the HMI is stopped and a red background in the animation appears (Fig. 6).

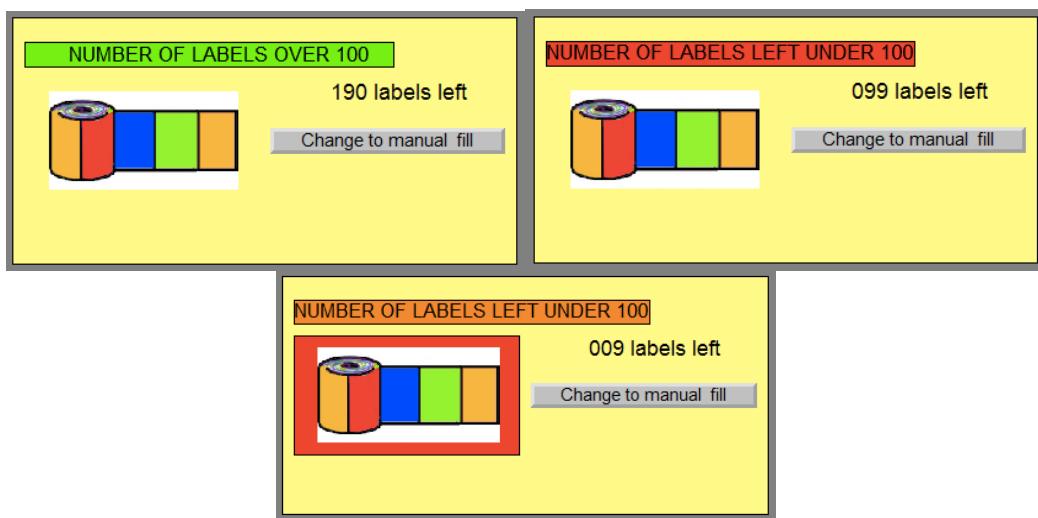


Fig. 13-Pictures of the labelling section of the HMI

After being labelled, bottles go to the putting caps station through a two-meter conveyor.

### 3.5 Putting Caps Station

This section is exactly the same as the labelling section but with changed names of blocks, variables and the animation shown in the figure 7.

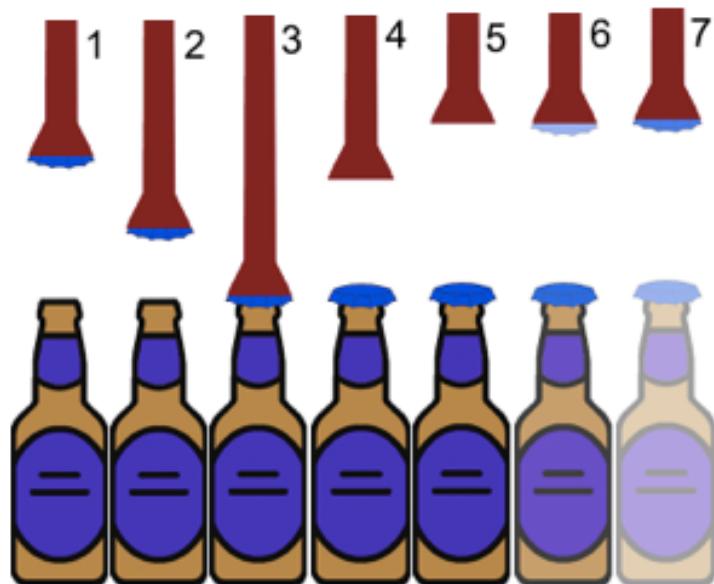


Fig. 14-Pictures of the animation of the putting caps section of the HMI

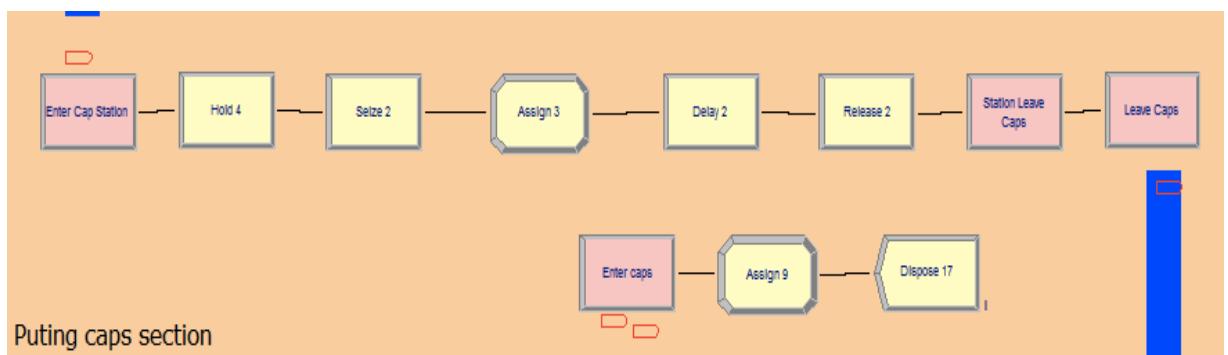


Fig. 15-Putting caps section in Arena

Once bottles have their caps, they are sent to the quality control station.

### 3.6 Beer, labels and caps warehouse

This is where beer, labels and caps are stored until they are needed.

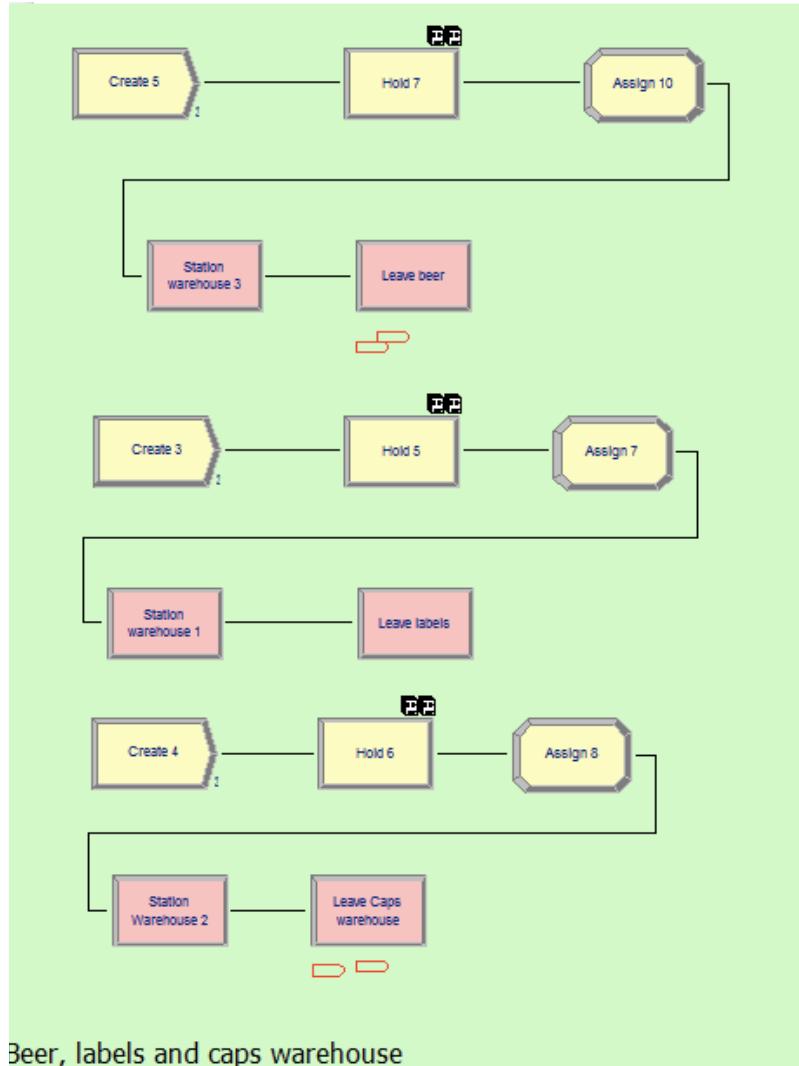


Fig. 16-Beer, labels and caps warehouse

When any of the machines which is using them is running out of them, an alarm is activated, as it is explained in the beer, labels and caps chapters.

When this alarm is activated, a cart goes from the warehouse to the machine to fill it again with beer, labels or caps.

From the HMI is possible to choose whether the system is working automatically or is being controlled by a user in the manual mode.

## Automatic mode:

In this mode, when the alarm is activated, a cart goes to the machine with an amount of beer, labels or caps that is enough to turn off the alarm and which guarantees the correct functioning of the machine.

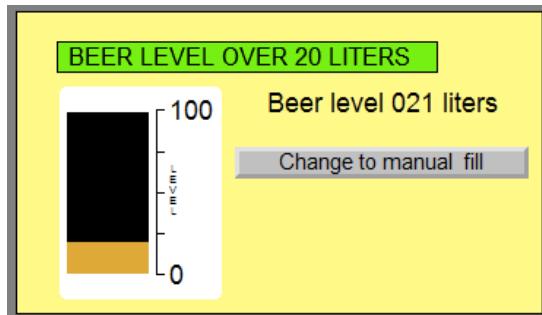


Fig. 17-Beer tank section of the HMI working in automatic mode

## Manual mode:

When using manual mode, you can choose the amount of beer, labels or caps that will be sent to the machine when the alarm will be activated.

User has to be aware that if the amount of beer, labels or caps send to the machine is not enough for turning off the alarm, although the alarm is still activated, not more carts will be sent to the machine automatically until it is turned off and on again.

In order to fix this problem, it is possible to send manually a cart with the desired amount of beer, labels or caps by pressing a button.

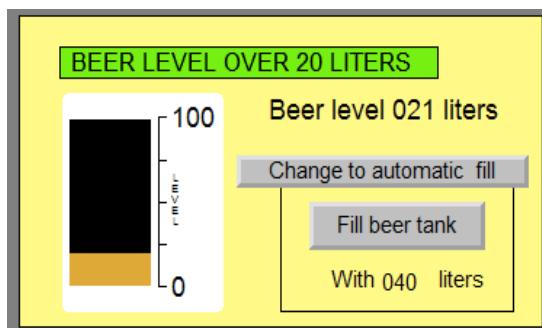


Fig. 18- Beer tank section of the HMI working in manual mode

This mechanism was implemented for the beer section with the following program:

When the beer alarm is activated (variable “levelalarm” has the value of one), the variable “alarmbeeractivated” has the value of one for half a second.



Fig. 19-Ladder program of the filling tank process

When “alarmbeeractivated” has the value of one, the action “hold\_beer1” of the following SFC is activated:

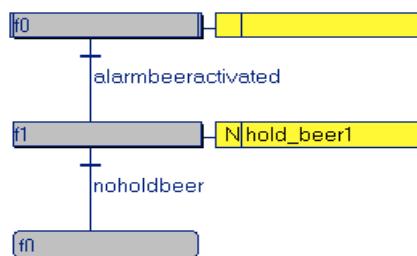


Fig. 20-Sequential function chart of the filling tank process

When it is activated, the value of the variable “holdbeer” is one.

When this happens, a beer pack that was created with other five more packs at the beginning of the simulation is released from a hold box in the beer warehouse done in Arena Software. It passes immediately through an assign box that changes the value of “holdbeer” to zero again, so not more beer packs are released.

When the value of this variable is zero again, the SFC activates the first action waiting until “alarmbeeractivated” has the value of one.

The beer pack released by the hold box, is carried by a cart to a station next to the beer tank. This station is connected to an assign box, so that when the pack arrives to this box, the beer level of the

tank is set to the value that results of the sum of the current tank level plus the litres of beer that the beer pack is supposed to have inside.

The quantity of beer that the beer pack is supposed to have inside is saved in the variable “beernecesary”.

If the program is working in the manual refilling mode, this value is set by the user of the HMI.

On the other hand, if it is working in automatic mode, the value of this variable is the number resulting of subtracting the current beer level from ninety-nine, so the beer level after the process has finished will be always ninety-nine litres.

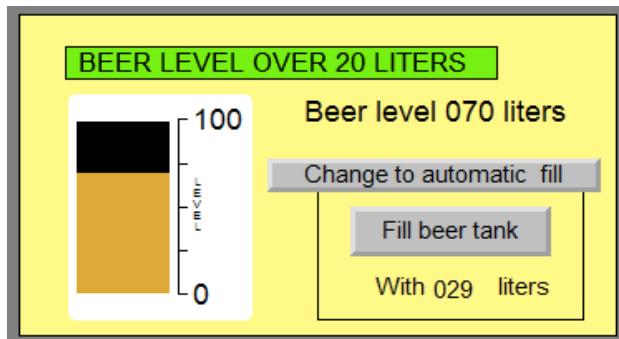


Fig. 21- Beer tank section of the HMI working in manual mode

This was implemented with the ladder program shown in figure 21:

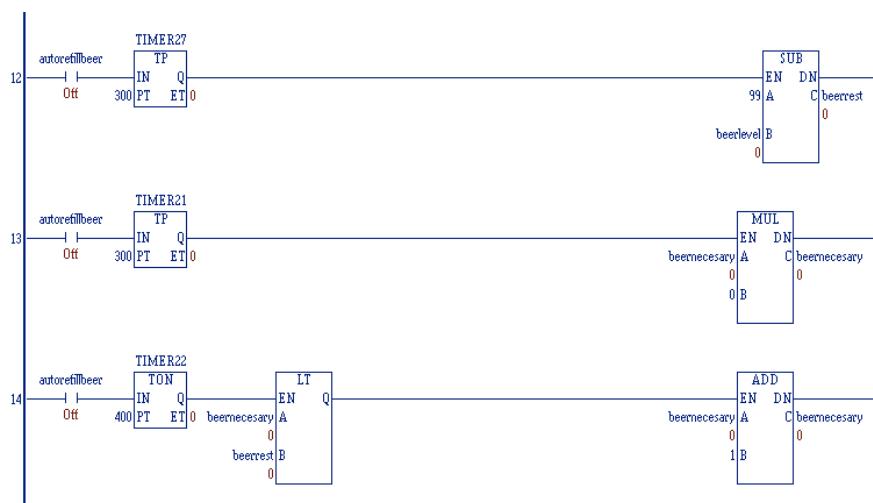


Fig. 22- Ladder program of the filling tank process

When using automatic mode, variable “autorefillbeer” is turned on.

When “autorefillbeer” is turned on:

Row 12:

During three hundred milliseconds, “beerrest” is set to the result of the subtraction of 99 and “beerlevel”.

Row 13:

During three hundred milliseconds, “beernecessary” is set to zero.

Row 14:

After four hundred milliseconds, “beernecessary” increases its value until it is equal to “beerrest”.

For the labelling and putting caps section, the same mechanism was used but with different variables and with the value of 190 labels or caps.

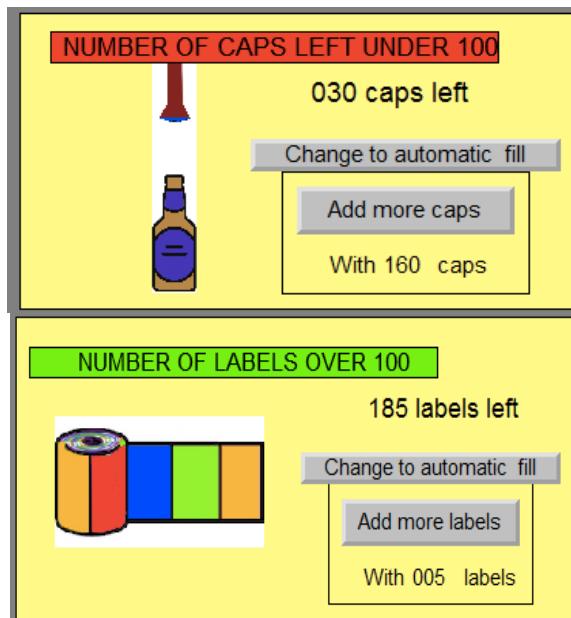


Fig. 23-Putting caps and labelling section of the HMI

### 3.7 Quality control section

Once bottles have their labels and caps, they are transferred to the quality control section.

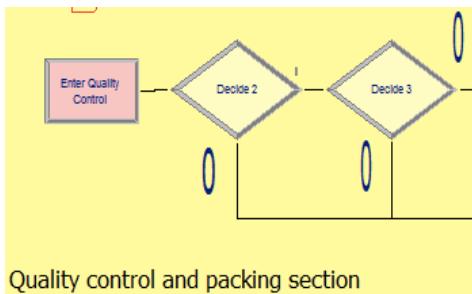


Fig. 24-Quality control and packing section

Bottles that have too much or not enough beer, are refused in this section.

The correct amount of beer that should be poured into each bottle is half a litter. Nevertheless, as said before, the pouring machine is not perfect. This means that sometimes it does not pour exactly half a litter of beer.

If the beer poured into a bottle is more than 0.43 litters and less than 0.57 litters, it is accepted. If the beer quantity inside the bottle is out of this interval, the bottle is automatically refused.

In the quality control section of the HMI, user have information about how many bottles have been accepted and refused (Fig. 15).

In this part of the interface you can find also a button. With it, you can force the quality control machine to refuse bottles even if they have the correct amount of beer.



Fig. 25-Quality control section working on “Refusing” mode

If “Accepting” mode is enabled, only beer bottles with the correct amount of beer will be accepted. On the other hand, when running in “Refusing” mode, all the bottles will be refused.

To change from one mode to the other, this button has to be pressed.

If the system is on “Accepting” mode, the button is green and it is written “Accepting” on it. When the button is pressed, the colour changes to red and the message displayed on it is “Refusing”.

This was implemented by configuring a button in Proficy Machine Edition. When it is pressed, it changes the value of the variable “refuse” from zero to one or from one to zero depending on its current value. When this value changes, the colour and the text displayed change.

In Arena Software, two decide boxes where used in order to implement this system. These boxes are in series.

The first one is used to force the rejection of bottles. Bottles are refused only when the value of the variable “refuse” is one.

If the value of this variable is zero, bottles go to the next decide box.

This one was configured so that all the bottles which do not meet the following condition are refused: the absolute value of the difference between beer quantity inside the bottle and 0.5 litters is less than 0.07 litters (Fig. 25).

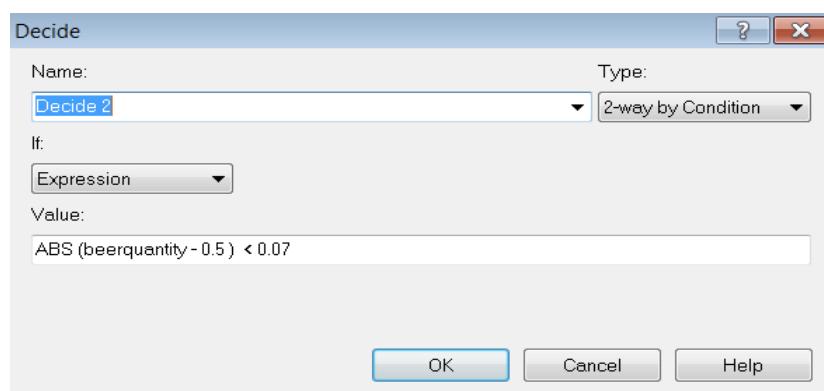


Fig. 26-Configuration of the second decide box in the quality control section

When a bottle is accepted, the value of the variable “acceptedbottles” is increased by one. On the other hand, every time that a bottle is refused, the value of the variable “refusedbottles” increases by one.

This was implemented by using two assign boxes.

The value of this variables is displayed in the HMI (Fig. 17).



Fig. 27-Quality control section of the HMI showing the results of the quality control process

### 3.8 Packing section

After defective bottles have been refused, the accepted bottles are packed into packs of six bottles.

This was implemented in Arena Software by using a batch box.

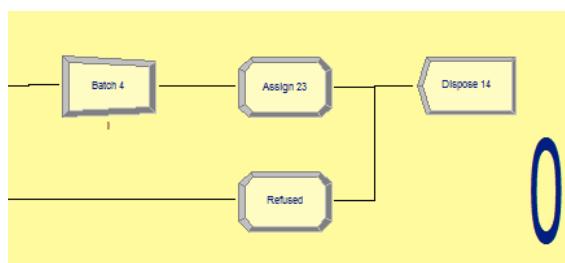


Fig. 28-Packing section in Arena

Once they have been batched, they pass through an assign box so that the image of the entity changes from a bottle to a box and entities are being removed from the system with a dispose box.

In the HMI, you can see how many packs have been produced. This was done by assigning the value of the variable “acceptedbottles” divided by six to the variable “totalpacks” and displaying it.

## **4. Conclusions**

After having done the simulation model, the control programme and the HMI, different configurations of the devices that transport bottles along the production line have been tested. It has been calculated the average time that takes to produce one bottle in a production batch of one hundred bottles depending on the speed of those devices.

Also, knowledge acquired during the realization of this thesis are explained in this chapter.

### **4.1 Influence of cart's and conveyor's speed**

The speed of the cart that takes batches of two packs of ten bottles each and the speed of conveyors that transport bottles along the production line has influence on the production speed.

The standard speed of the cart is 1 meter per second and the standard speed for the conveyors is 0,7 meters per second.

For this experiment, the simulation was run nine times with different speeds. One time at the standard speed, four times at higher speeds and four times at lower speeds.

The lower speeds were 0.2, 0.4, 0.6, and 0.8 times the standard speed.

The higher speeds were 1.2, 1.4, 1.6 and 1.8 times the standard speed.

With this, the speeds of the cart and conveyors were the following (Table 1):

<b><i>Cart speed (m/s)</i></b>	<b><i>Conveyors speed (m/s)</i></b>
0,2	0,14
0,4	0,28
0,6	0,42
0,8	0,56
1	0,7
1,2	0,84
1,4	0,98
1,6	1,12
1,8	1,26

Table 1-Speed of the cart and conveyors of the different tests

The results of the experiment are the following (Table 2):

<b><i>Multiplier</i></b>	<b><i>Total time (min)</i></b>	<b><i>Average time to produce one bottle (s)</i></b>
0,2	15,67	9,402
0,4	8,84	5,304
0,6	6,5	3,9
0,8	5,54	3,324
1	4,94	2,964
1,2	4,67	2,802
1,4	4,4	2,64
1,6	4,32	2,592
1,8	4,2	2,52

Table 2-Results of the different speed tests

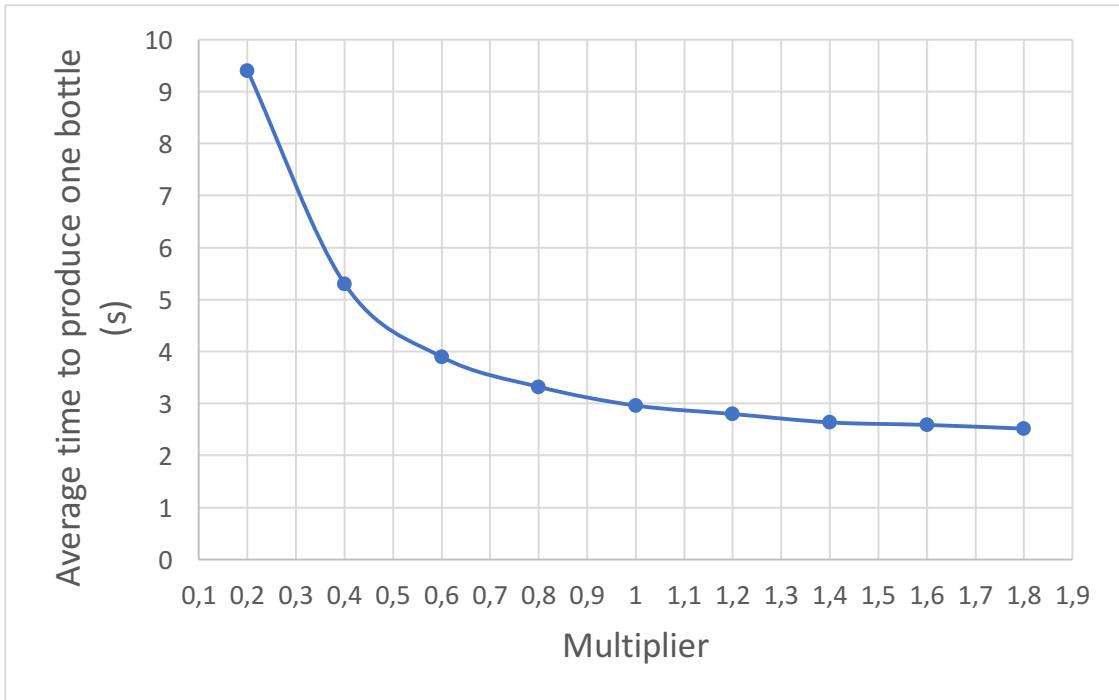


Fig. 29-Graph of the results of the different speed tests

As we can see in the graph above, decreasing the speeds has a huge influence on the mean time that takes to produce one bottle.

On the other hand, increasing the speed up to almost the double of the standard speed, obviously has influence on the average time, but not that much.

For this reason, it is considered that the standard speed is enough for this bottling plant.

In order to be able to know which will be the mean time that takes to produce one bottle in a one hundred bottle batch at any other speed, an equation was created from the experimental data.

This equation is a discontinuous function. For values of multiplier from 0.2 to 1.8 this equation is a polynomial function. For values out of this range, the equation is a potential function.

A discontinuous function was used because as it was expected, for values higher than 1.8 the polynomial function rises but for values in the interval from 0.2 to 1.8 it fits better than a potential function.

This was created using two trend lines in Excel.

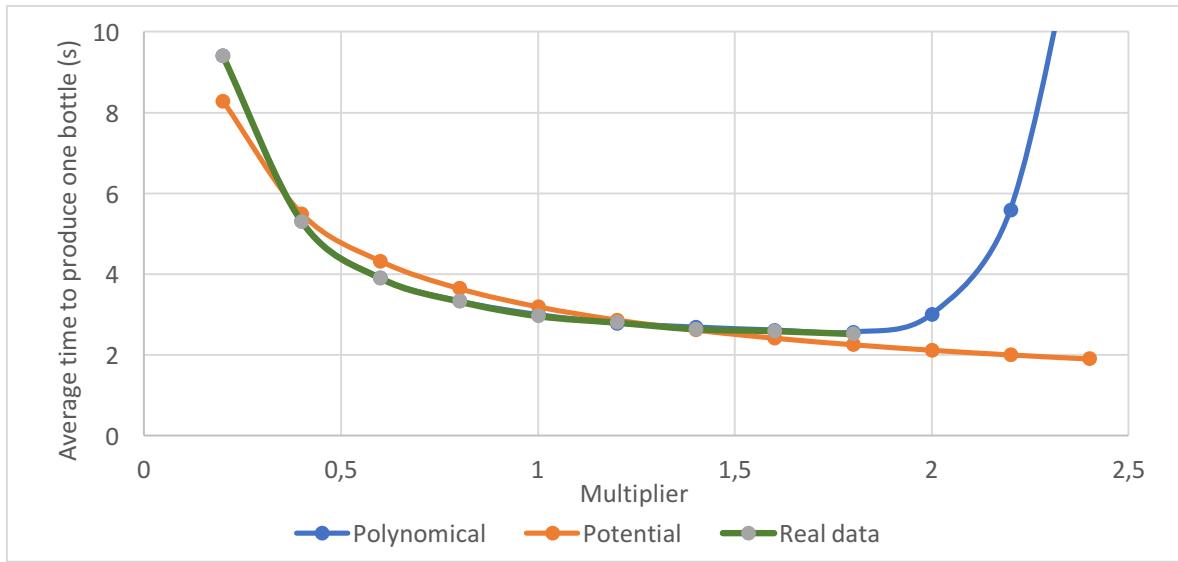


Fig. 30-Graph of the two trend lines and real data

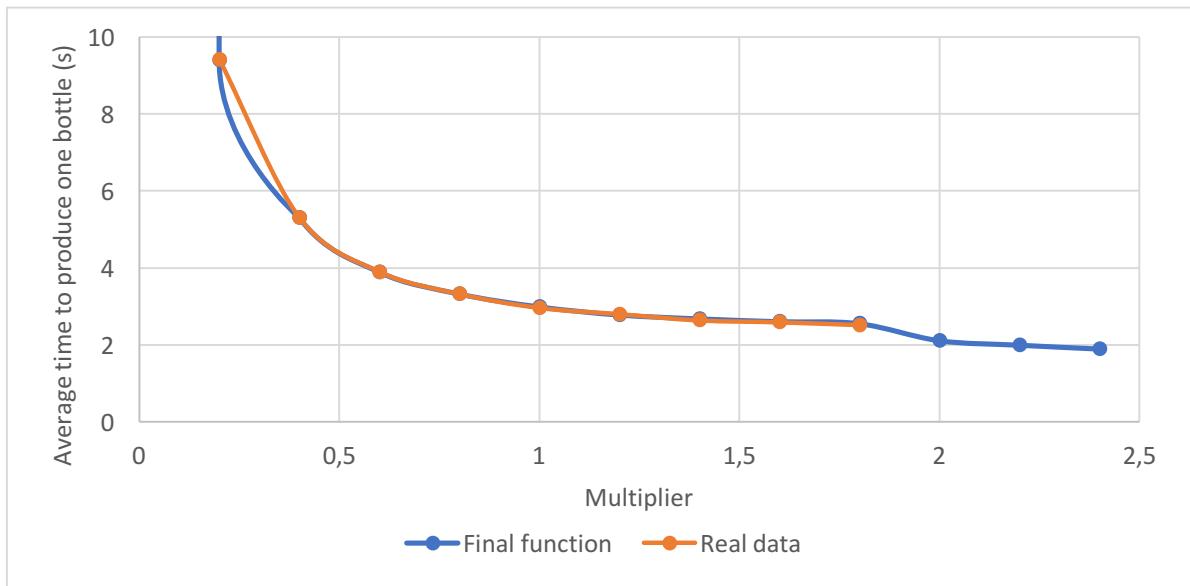


Fig. 31-Graph of the final function and real data

The Final function is:

$$f(x) = 3,1891x^{0,593} \quad \text{if } 0 \leq x < 0.2 \text{ or } x > 1.8$$

$$f(x) = 6,0503x^6 - 43,748x^5 + 129,01x^4 - 199,36x^3 + 172,28x^2 - 81,442x + 20,201 \quad \text{if } 0.2 \leq x \leq 1.8$$

## **4.2 Knowledge acquired**

It has been learnt which are the parts of a beer bottling plant and how they work.

Knowledge has been acquired in terms of using programs in the area of automatics. Specifically using Proficy Machine Edition.

More experience has been gained in the area of implementing algorithms in LD or using SFC.

Also, it has been learnt how to use programs that allow to configure a HMI

Using Arena Software has been useful to take the first steps in learning how simulation programs work and learn how to do simulations of discrete production processes. It has been useful in order to realize of its utility and potential in industry.

Once both the simulation and the control program were done, the correct operation of them has been checked using a touch panel to interact with the HMI.

## **5. List of figures**

Figure 1: 12" QuickPanel View & QuickPanel Control

Figure 2: Process state update section in Arena

Figure 3: HMI

Figure 4: Storage facility in Arena

Figure 5: Cleaning section in Arena

Figure 6: Ladder program of cleaners section

Figure 7 Cleaners section of the HMI

Figure 8: Cleaners section of the HMI

Figure 9: Pouring beer section in Arena

Figure 10: Beer tank section of the HMI

Figure 11: Labeling section in Arena

Figure 12: Pictures of the labelling animation

Figure 13: Pictures of the labelling section of the HMI

Figure 14: Pictures of the animation of the putting caps section in of the HMI

Figure 15: Putting caps section in Arena

Figure 16: Beer, labels and caps warehouse

Figure 17: Beer tank section of the HMI working in automatic mode

Figure 18: Beer tank section of the HMI working in manual mode

Figure 19: Ladder program of the filling tank process

Figure 20: Sequential function chart of the filling tank process

Figure 21: Beer tank section of the HMI working in manual mode

Figure 22: Ladder program of the filling tank process

Figure 23: Putting caps and labelling section of the HMI

Figure 24: Quality control and packing section

Figure 25: Quality control section working on “Refusing” mode

Figure 26: Configuration of the second decide box in the quality control section

Figure 27: Quality control section of the HMI showing the results of the quality control process

Figure 28: Packing section in Arena

Figure 29: Graph of the results of the different speed tests

Figure 30: Graph of the two trend lines and real data

Figure 31: Graph of the final function and real data

Table 1: Speed of the cart and conveyors of the different tests

Table 2: Results of the different speed tests

## **6. Bibliography and sources**

### **Simulation of production processes:**

- [1] Application of Modeling and Simulation Techniques for Technology Units in Industrial Control

[https://link.springer.com/chapter/10.1007/978-3-642-27552-4\\_67](https://link.springer.com/chapter/10.1007/978-3-642-27552-4_67)

### **Types of simulation software**

- [2] Softwares de simulación

<http://softwaresdesimulacion.blogspot.com/2014/02/softwares-de-simulacion.html>

### **GE's devices**

- [3] GE Automation Control Systems

<http://www.geautomation.com/products/control-systems>

### **Arena Software**

- [4] Clarcat

<https://www.clarcat.com/arena/>