Delft University of Technology

FACULTY MECHANICAL, MARITIME AND MATERIALS ENGINEERING
Department Marine and Transport Technology

Mekelweg 2
2628 CD Delft
the Netherlands
Phone +31 (0)15-2782889
Fax +31 (0)15-2781397
www.mtt.tudelft.nl

Specialization:	Transport Engineering and Logistics
Report number:	2017.TEL.8138
Title:	**Optimization of the capacity of a rose sorting system using discrete event simulation.**
Author:	W. Vreugdenhil

Title (in Dutch)	Optimalisatie van de capaciteit van een rozensorteersysteem door middel van discrete event simulatie.
Assignment:	Masters thesis
Confidential:	yes
Initiator (university):	Dr.ir. D.L. Schott
Initiator (company):	Ing. Kees Bukman
Supervisor (university):	Dr. ir Y. Pang
Supervisor (company):	Dr. ir. Wouter Bac
Date:	August 22, 2017

Summary

In rose cultivation greenhouses, roses are grown, sorted and packaged. The process of sorting and packaging is largely automated. This process is executed by a number of machines, that together form a sorting system. A sorting system consists of five standard machines: sorting machine, bunching stations, buffer belts, V-belts and sealers. Except for the sorting machine, the machines may be present multiple times within one sorting system depending on the number of quality classes, which have to be distinguished and other requirements regarding the sorting capacity and degree of automation.

However, due to the growth of the size of rose cultivation greenhouses, there is a demand for sorting systems, that have a larger capacity to be able to process all roses, that are harvested in one day may be sorted and packed within one working day (8 hours). This demand is also the case in the rose cultivation company “Porta Nova B.V.”. Porta Nova planned to build a new greenhouse, with a larger sorting system and better specifications. This research focuses on the casus of Porta Nova.

There are many ideas present for adjusting the rose sorting system machines, to enlarge the sorting capacity. These ideas come from rose sorting employees, greenhouse owners and 4More Technology employees. However, all these adjustments are reduced to five bottlenecks. Those five bottlenecks are 1) filling ratio of the sorting machine 2) fork flow of the sorting machine 3) speed of bunching station actions 4) belt system overflow and 5) number of attachment locations on the sorting machine. However, there is no method to predict the increase of capacity of the sorting system if one or multiple machine adjustments will be applied.

To be able to predict the change of sorting capacity for a number of machine adjustments, a discrete event simulation model is developed. This discrete event simulation model is constructed in Simulink & Matlab. For each of the five standard machines, a subsystems is developed. Using those five subsystems, each possible sorting system setup may be simulated. Subsequently the simulation model is verified matched and validated. To validated the model, the existing situation of rose cultivation company “Meewisse Roses B.V.” is mimicked. Results of a single day validation showed that the time to process the roses is simulated with a 97% accuracy.

When the model is validated, it is used for the rest of the research. For the casus of Porta Nova, a basic sorting system setup is defined and simulated. Subsequently five situations are defined. In each of those situations, one of the aforementioned bottlenecks is removed or reduced. However, entirely removing a bottleneck is not feasible in reality for all bottlenecks. Finally a seventh and last situation is defined where all feasible bottleneck improvements are combined.

Results of the simulations show that in the basic situation, it takes 13.3 hours to process all roses of one day. Four out of five situations where a bottleneck is reduced showed a reduction of processing time varying from 9% to 19%. The situation with all feasible bottleneck improvements showed a sorting time

reduction of 35 %. 8.6 hour remain to process all roses of one day. Although the time reduction is a large improvement, the desired 8 hours cannot be reached.

Further improvement of the sorting capacity is possible. It is recommended to do further research on the sorting capacity optimization.

List of symbols

Rf	Filling ratio [-]	(equation 1)
Rf_{actual}	Actual filling ratio in simulation [-]	(equation 3)
ff	Fork flow [Forks/hour]	(page 7)
Nf	Number of forks [forks]	(equation 6)
Nf_{full}	Number of full forks that passed the camera [forks]	(equation 1)
Nf_{total}	Total number of forks that passed the camera [forks]	(equation 1)
Nf_{empty}	Total number of empty forks that passed the camera [forks]	(equation 3)
Nr	Number of roses [roses]	(equation 5)
$Nr_{generated}$	Number of rose entities generated in simulation [rose entities]	(equation3)
Nr_{round}	Number of rose entities that made a round in the sorting machine [rose entities]	(equation 3)
$Nr_{present}$	Number of rose entities present [rose entities]	(equation 4)
$Nr_{remaining}$	Number of rose entities remaining [rose entities]	(equation 4)
t	Time [seconds]	(equation 6)
P	Period [seconds]	(equation 2)

Contents

1. Introduction

In rose cultivation greenhouses, roses are grown, sorted and packaged. The financial value of a rose is dependent on its appearance. Factors like stem length, bud size, colour and ripeness determine the quality of roses.

Rose growers in the Netherlands face a lot of competition from Rose growers that are settled in Africa(Henten, 2006; Verbeek, 2016) having favorable climate conditions and lower labor costs. Many Dutch rose growers went bankrupt in the past two decades. The Dutch rose growers that remain, are the rose growers with a large company and differentiate themselves by higher quality of large-flowered roses with a higher value per product (Hoog, 2001).

Since mid-90 there was hardly any innovation in the sorting process of roses. However, with the current developments of camera's and vision systems there was much room for improvement. One of the companies that is working on this innovation is 4More Technology B.V.. This company developed a new rose grading system, that is called IRISS (Intelligent Rozen Inspectie en Sorteer Systeem) ("IRISS," 2016). Compared with the old system, the quality class of each rose is determined more accurately, because IRISS is able to measure more properties than the old system and IRISS determines the properties more precisely. IRISS may be installed on the existing old rose sorting machines, that are produced by the company AWETA ("aweta," 2017). Besides the development of IRISS, 4More Technology ("4More Technology," 2017) also focuses on the development of the sorting systems.

The current rose sorting systems consists of a number of machines. In short:

- Sorting machine: Defines the quality class for each rose and sends each rose to the proper outlet.
- Bunching station: Bundles roses to bunches.
- Buffer belts: Buffers bunches of roses.
- V-belt: transport bunches of roses to the sealer.
- Sealer: Puts a plastic wrapping around each bunch of roses.
- Buffer belt after sealer: buffer packaged bunches of roses.

Not all machines are always necessary and in some companies not all machines are present. The composition of the sorting system is different for each greenhouse. The systems will vary in the length of the sorting machine, the number of bunching stations, belts and sealers and the presence or absence of the belts and sealers.

The growers may decide themselves on the quality classes by defining the number of quality classes and adjusting the limits for each property in each quality class. The number of quality classes is also dependent on the number of outlets (bunching stations) available in the sorting machine. In practice it appears that when IRISS is installed on the sorting machine, the growers may and want to define more quality classes than before. The number of outlets on the sorting machine is in many cases not sufficient for the ideal number of quality classes.

Because of the aforementioned reasons, in the present time there is a demand for rose sorting systems that have more outlets and a higher capacity than before. This demand is also the case in a Rose cultivation company called Porta Nova B.V. ("Porta Nova," 2017). Porta Nova is the biggest rose grower of the cultivar "Red Naomi" ("Red Naomi," 2017) (large flowered red roses) in the world and is famous for the high quality of the roses ("G-fresh," 2016). Porta Nova currently has two greenhouses that have both their own sorting machines. They planned to build a third greenhouse called Porta Nova III where also the roses of Porta Nova I & II will be are sorted and packaged. Therefore, in Porta Nova III a new sorting system will be composed, that will exist of the aforementioned standard machines.

For the new sorting system in the location Porta Nova III the following requirements are set:

- 36 outlets/quality classes
- 1 sorting system
- Process the roses produced in Porta Nova I and Porta Nova II (80.000 – 100.000 roses) in a regular working day (8 hours)

Many options may be thought to adapt the sorting system and gain a higher capacity. The restrictions from 4More Technology are:

- Existing machines has to be reused
- Small physical adaptions are allowed

1.1 Problem definition

In the existing sorting systems, the capacity that is desired has never been reached. Therefore adaptations to the standard machines have to be made. Many ideas are present on how to adapt the machines to achieve a higher capacity. These ideas have been put forward by the greenhouse owners and the employees of 4More Technology. For many ideas, with common sense it may be assumed that the capacity of the sorting system will increase. However, currently there is no way to quantify the capacity increase for each adjustment except for testing in reality.

1.2 Research goal

The goal for this research is to compose an advice for adjustments of the sorting system machines to achieve the aforementioned requirements. To be able to formulate such an advise at first an understanding of the sorting process has to be gathered. Thereafter a way to predict the effect of a certain adaptation to the sorting system machines has to be provided.

1.3 Research scope

It is important to define the boundaries of this research project. The boundaries of the sorting process determine the demarcation of the research. The sorting process is independent of other activities in the greenhouse. The sorting process starts with attaching roses to the sorting machine and ends with grabbing bunches of sorted and sealed roses from a belt to put them in a bucket.

1.4 Research questions

From the research goal a main research question is formulated:

Which adjustments of the sorting system machines are recommended to increase the capacity of an existing rose sorting system?

This main question corresponds to the following sub questions:

1. **How is the sorting process composed and which parameters influence the capacity?**
2. **What are the bottlenecks in performance of the current system?**
3. **How can the sorting system be simulated?**
4. **What is the effect of fixing each bottleneck?**
5. **Which adjustments are feasible in reality?**

1.5 Structure

The final goal for this research is to formulate an advice on how to enlarge the sorting capacity. To investigate the possibilities for improving the capacity, a simulation model is constructed. The steps that

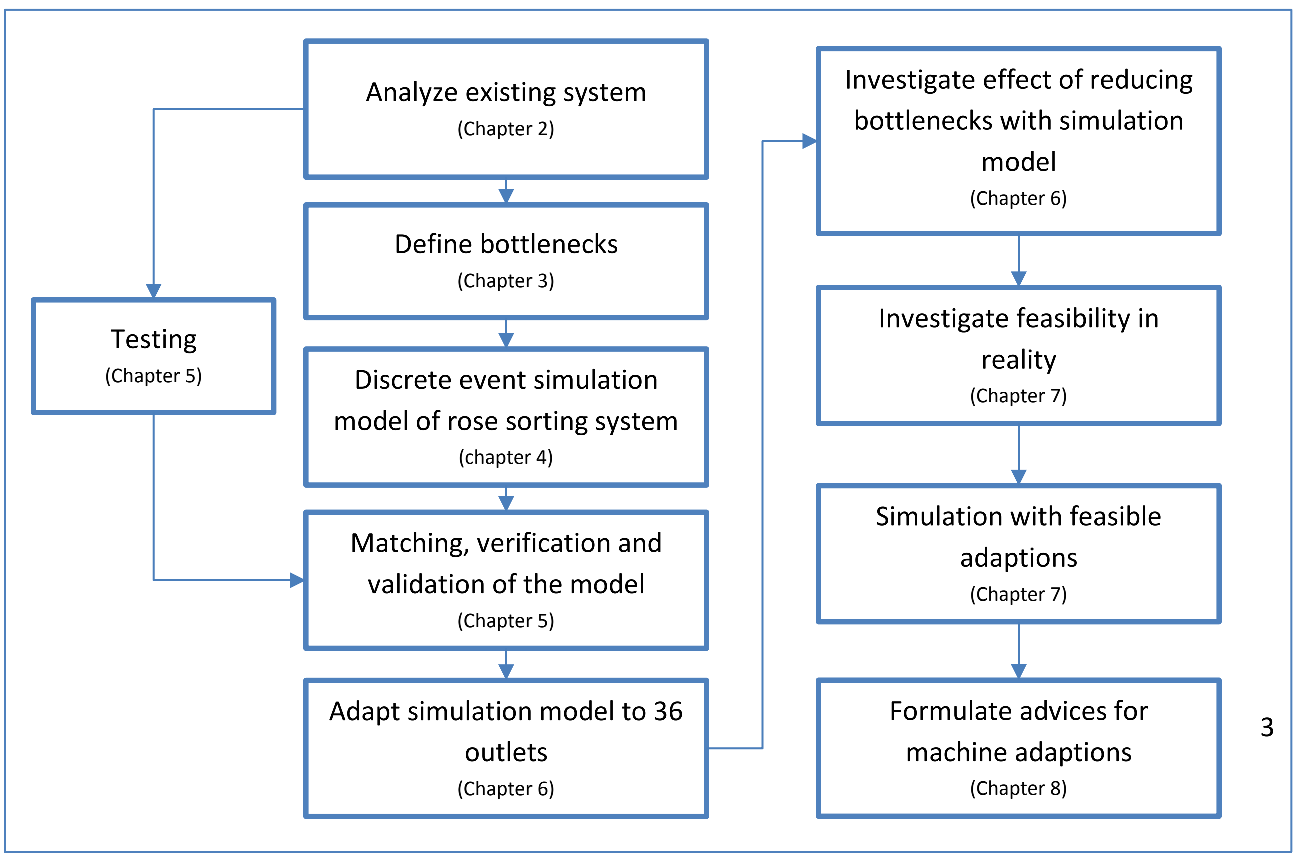


Figure 1-1 research structure

will be taken are depicted in Figure 1-1.

The research starts with the process description of the rose sorting system (Chapter 2). In the rose sorting system, 5 bottlenecks of are defined (Chapter 3). Using the process description and the bottlenecks, a discrete event simulation model is constructed and elaborated in chapter 4. This simulation model consists of subsystems that represent the standard machines in the rose sorting system. In chapter 5, the simulation model is verified, matched and validated using the information of a real sorting system. After the verification, the simulation model is adapted to a sorting system with 36 outlets, which is the desired situation of Porta Nova III(Chapter 6). In this chapter, also the effect of reducing the bottlenecks, that are discussed in chapter 3, is investigated using the simulation model. In reality the bottlenecks cannot entirely be removed, but in chapter 7 it is investigated what is feasible in reality at this moment. In chapter 7 also a last simulation is done, to check the effect of the adaptations that are feasible at this moment. Using the information achieved in chapter 6 and chapter 7, an advice is formulated for the adaptions to the sorting system machines (Chapter 8).

2 Process description

An example of a rose sorting system is shown in Figure 2-1. The system consists of the following 6 machines: `

- Sorting machine
- Multiple bunching stations
- Multiple buffer belts
- V-belt
- Sealer
- Buffer belt after sealer

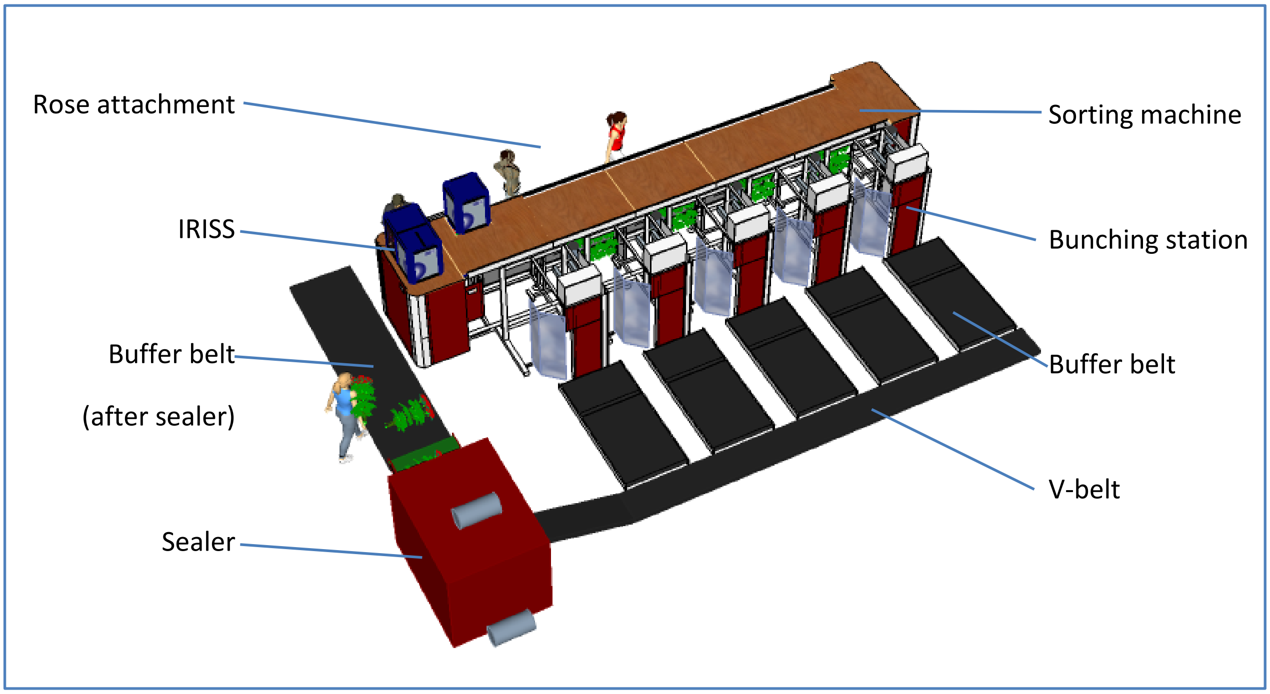


Figure 2-1 example sorting system

The composition of the sorting system is different in all greenhouses. The length of the sorting machine will vary, the number of bunching stations and buffer belts vary and there may be multiple V-belts and sealers, each serving a group of bunching stations and buffer belts. Besides varying the number of machines, some machines may even be left out. Only the Sorting machine and the bunching stations are necessary, the other machines, or a part of the other machines are left out in some greenhouses.

A schematic overview of all processes in a sorting system is shown in Figure 2-2. The sorting process starts with attaching the rose in the sorting machine and ends with grabbing packaged and sorted bunches of roses from the buffer belt, and putting them in a bucket. At first a brief introduction of the entire sorting system process is given, then a more elaborate process flow chart for each machine is

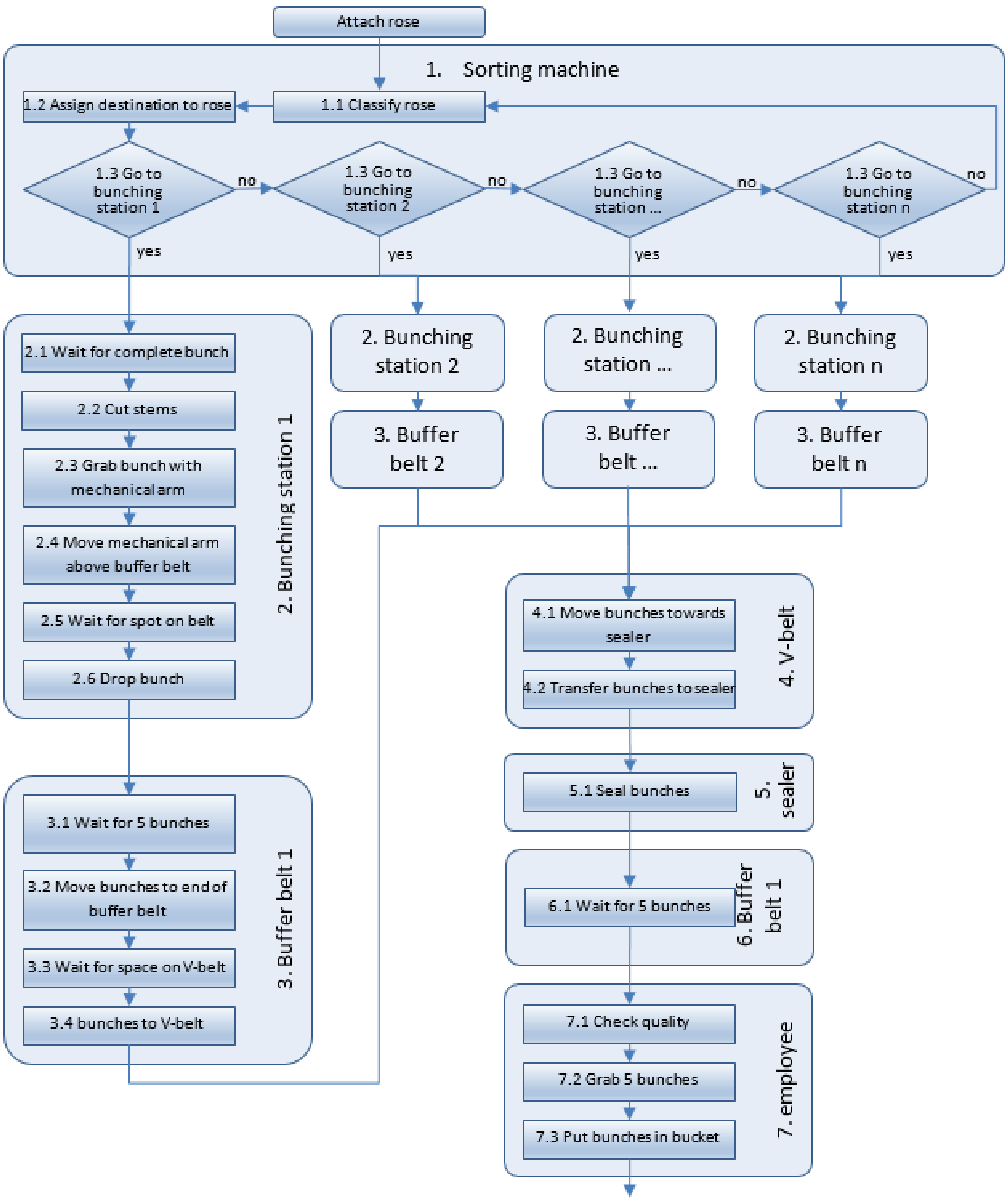
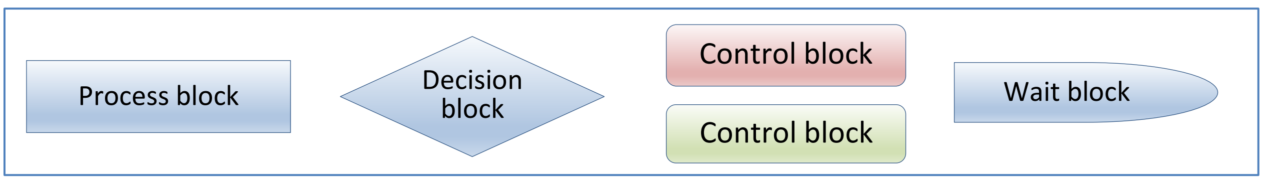


Figure 2-2 process flow chart of entire sorting machine provided and discussed.

The process starts with attaching roses in the sorting machine. This task is manually done by employees. The roses are classified in the sorting machine and a quality class is assigned to each rose. A destination (bunching station number) is assigned to the rose, depending on the quality class. The sorting machine sends the rose to the correct bunching station. The bunching station transforms separate roses into bunches of roses and transfers the bunches onto a buffer belt. On the buffer belt, the number of bunches (5 bunches) that is sold in a bucket is collected. If that number is reached the bunches are transferred to the V-belt, that transfers the bunches to the sealer. The sealer puts a plastic wrapping around the bunch. Afterwards the employees do a quality check and put the right number of bunches (5 bunches) in a bucket.

In the next sections a process flow chart of each standard machine is provided. Each sections starts with a general description of the tasks of the machine and ends with an elaborate process flow chart with a description per block. Four standard shapes are used in the flow charts (Figure 2-3).



A process block is defined as a process that takes a certain time. A decision block is a block where two routes may be taken. The control blocks control an according decision block, that may be located in the machine of the control block or in another machine. In a waiting block is waited until a process in

Figure 2-3 Process flow chart standard blocks

2.1 Sorting machine

Some pictures of the sorting machines are provided in Figure 2-4. In the sorting machines roses are attached to forks by employees. The forks are mounted to a chain that circulates around in the sorting machine. The forks move in and out from the sorting machine to be able to attach the roses and to transfer the roses to the bunching stations. In the sorting machine, mechanical switches are placed to be able to transfer a rose to a bunching station, or keep the rose in the sorting machine.

Figure 2-4 attaching roses to the sorting machine

Figure 2-5 transfer from sorting machine to bunching station

The number of roses that pass the camera is dependent on two variables, namely the fork flow (F_f in Forks/hour) and the filling ratio (R_f). The fork flow is adjustable through the settings of the sorting machine. The filling ratio is defined as the number of full forks (Nf_{full}) that pass the camera relative to the total number of forks (Nf_{total}). The filling ratio is calculated over a certain timespan.

$$R_f = \frac{Nf_{full}}{Nf_{total}} \qquad \text{(Equation 1)}$$

- R_f = Filling ratio [-]
- Nf_{full} = Number of full forks that passed the camera over a timespan [forks]
- Nf_{total} = Total number of forks that passed the camera over a timespan [forks]

The classification is done by IRISS. IRISS is described in more detail in Appendix B.

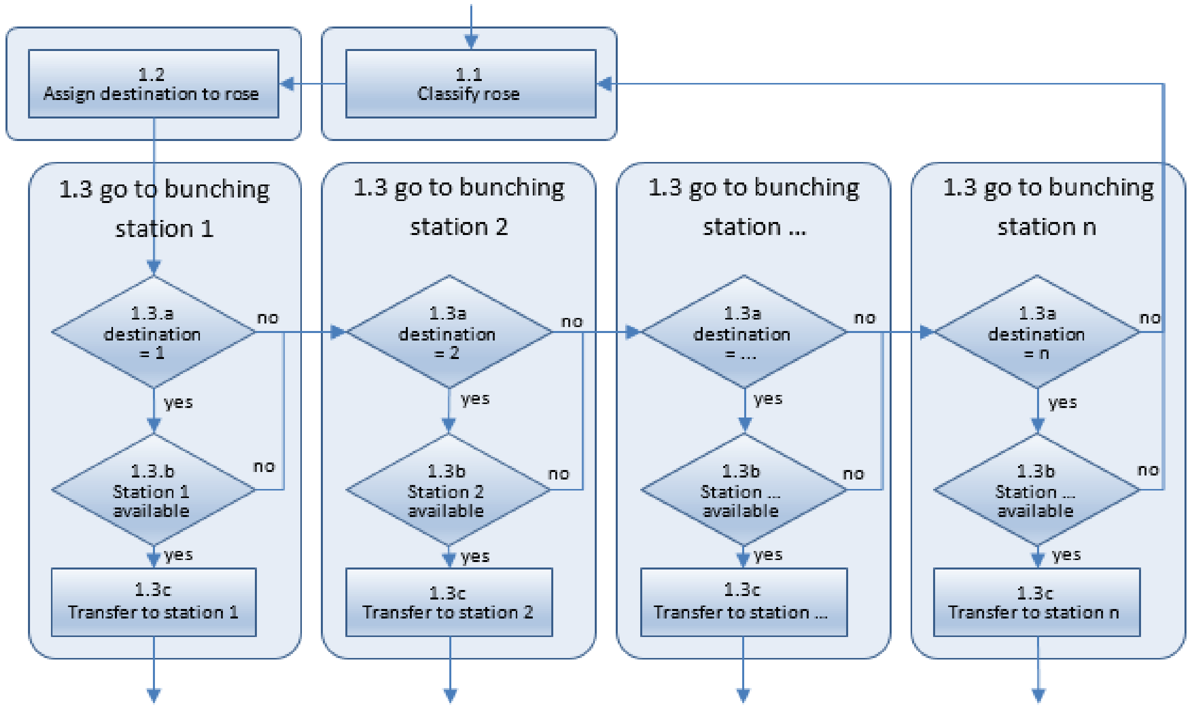


Figure 2-6 process flow chart of sorting machine

A process flow chart is provided in Figure 2-7. The process steps of the sorting machine are:

1.1 Within the Sorting machine multiple pictures are taken from each rose. With image processing the properties (Figure 2-7) of each rose are defined. Depending on the properties, a quality class is assigned to each rose.

1.2 Each quality class is assigned to a bunching station, that is called the destination.

1.3 For each rose it has to be decided at each bunching station of the sorting machine should move the fork out and transfer the rose to the bunching station.

1.3a At first the destination is compared to the number of the bunching station.

1.3b Then it is checked if the bunching station is available. The bunching station may be unavailable for multiple reasons. Those reasons is shown in the process flow chart of the bunching station through the green and red blocks.

1.3c The fork is moved out and the rose is transferred to the bunching station (Figure 2-5).

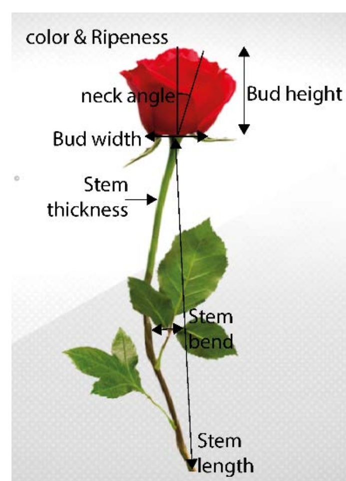


Figure 2-7 Rose properties

2.2 Bunching station

The bunching station consists of collector forks, a mechanism to attach a string at the stems of the roses, pushing rods and a mechanical arm . The collector forks are used to form bunches of roses. In case of a bunch of 10 roses 3 roses are collected on the first collector fork 4 roses are collected on the second collector fork and 3 roses are collected on the third collector fork. The distribution of 3, 4 and 3 roses on the collector forks, will form a nice round bunch when a string is attached to the stems. When the bunch is completed the string is attached. A bunch will stay in the bunching station until the number of roses is reached to transfer a bunch to the buffer belt. This transfer is done using the mechanical arm.

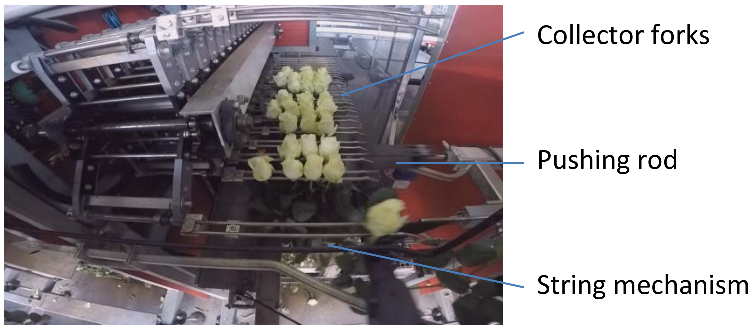


Figure 2-8 bunching station seen from the sorting machine

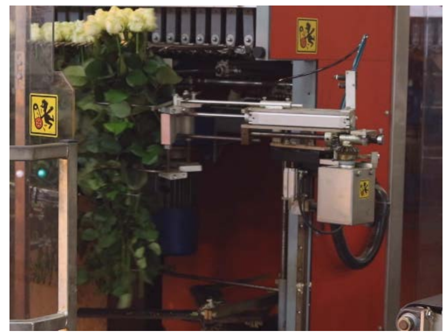


Figure 2-9 bunching station with mechanical arm grabbing a bunch

In Figure 2-10 the process flow chartof a bunching station is shown. In this process flow chartalso green and red blocks are present. These blocks influence a choice block somewhere else in the process flow chart.

2.1a In a bunching station there is one incomplete bunch (in Figure 2-8, the incomplete bunch has already 7 roses and the eight rose is arriving). Is the arriving rose the seventh rose?

2.1b Is the arriving rose the 3rd rose?

2.1c Is the arriving rose the 10th rose?

2.1d The collector fork is full and a new empty collector fork is positioned (fork change). The time it takes to change the fork may vary per bunching station. The time is adjustable through the bunching station settings (AWETA computer) and through the air valves on the mechanic component that are powered by compressed air.

2.1e Are there three bunches present in the bunching station? (In Figure 2-8 two full bunches are present).
Are there three bunches present?

2.1f Is a fork change allowed?

2.1g Wait until bunches < 3. (a bunch has to be transferred from the bunching station to the buffer belt).

2.1h A string is attached to the stems of the bunch. From now on the 10 roses will form a bunch. The time it takes to attach the string varies per bunching station. The time is adjustable through the bunching station settings and through the air valves on the mechanic component that are powered by compressed air.

2.2 The stems of the roses are cut to a preset length.

2.3a Is the mechanical arm available?

2.3b Wait until the mechanical arm is available. In some cases the mechanical arm is already occupied.

2.3c While the mechanical arm is moving out, no fork change can be executed.

2.3d When the mechanical arm is being used it is not available for the next bunch.

2.3e The bunch is grabbed with the mechanical arm

2.4a The mechanical arm is moved until the bunch is located above the buffer belt. The time it takes to move the arm out varies per bunching station and is adjustable through machine setting and air valves.

2.4b A fork change is allowed again.

2.5a Is the small buffer belt available?

2.5b Wait until the small buffer belt is available again.

2.6a The bunch is dropped from the mechanical arm on the small buffer belt.

2.6b The mechanical arm moves back. The time it takes to move the arm out varies per bunching station and is adjustable through machine setting and air valves.

2.6c The mechanical arm is available again.

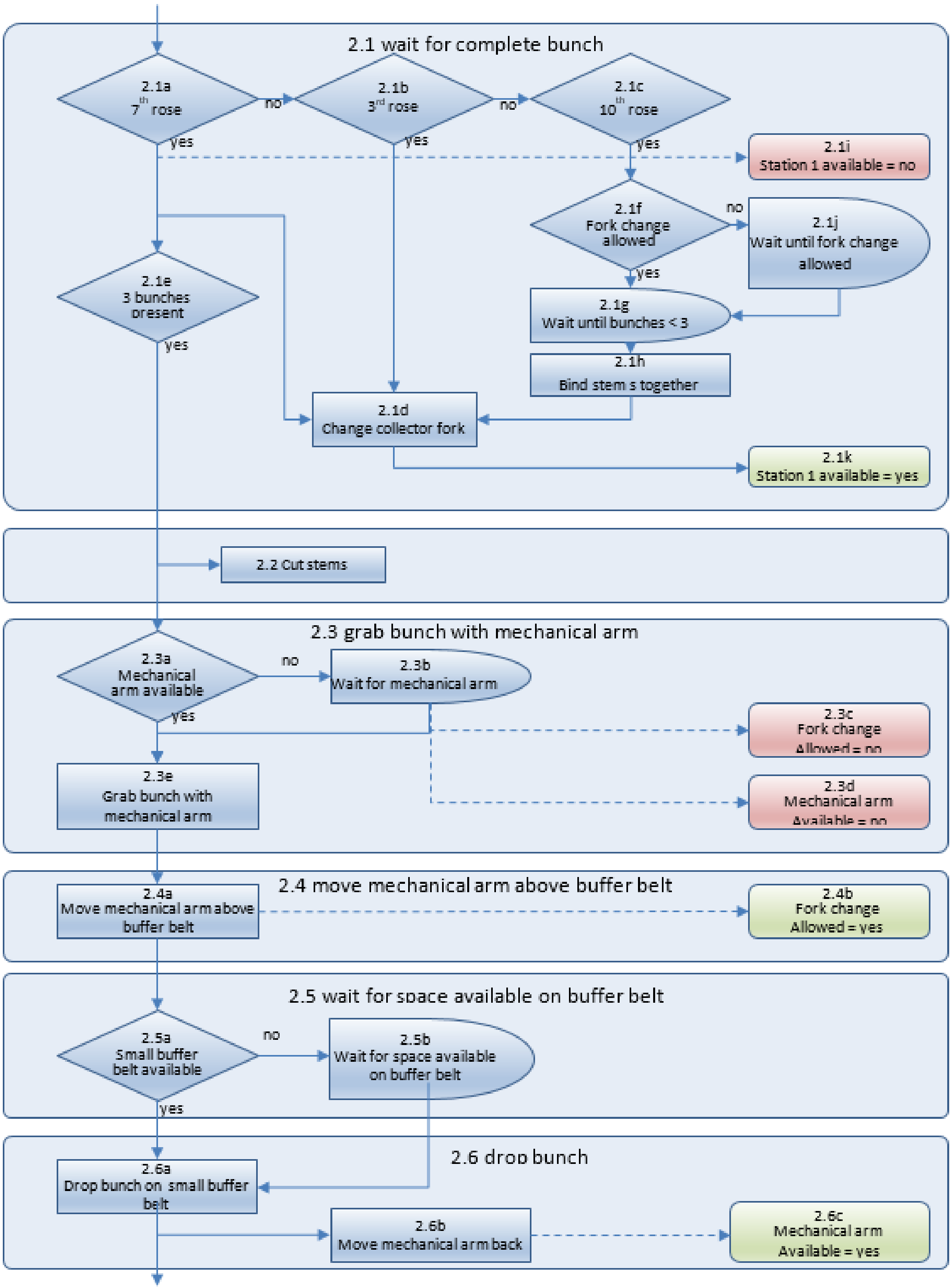


Figure 2-10 process flow chart of bunching station

2.3 Buffer belt

The bunches are packaged in a bucket per 5 bunches of the same quality. The buffer belts (Figure 2-11) are used to make it possible that the bunches arrive at the end of the system in groups of 5. The bunches are dropped on the beginning of the belt (right side in Figure 2-11). When there are 5 bunches on the buffer belt, the bunches are moved to the end of the buffer belt and transferred to the V-belt.

The buffer belt is split up into a small buffer belt at the beginning and a large belt on the end. This split up makes sure that when all bunches are aligned at the end of the large buffer belt and waiting to be transferred to the V-belt, a next bunch may already be dropped on the small buffer belt.

Figure 2-11 buffer belts

In some cases more belts are full and ready to transfer the bunches to the V-belt at the same time. In this case the priority rules define which belt is the first belt to transfer roses to the V-belt. Figure 2-12 is used to explain the priority rules. The priority rules are defined as follows:

1. When there is no buffer belt transferring bunches to the V-belt and one of the buffer belts is full, this buffer belt may start transferring bunches to the V-belt

2. When a buffer belt (belt A) is transferring bunches to the V-belt and another buffer belt (belt B), further away from the sealer, is full and is waiting until it is allowed to transfer bunches to the V-belt, buffer belt B starts transferring bunches as soon as the number of bunches on buffer belt A is equal to the number of empty locations on the V-belt between buffer belt A and buffer belt B. In

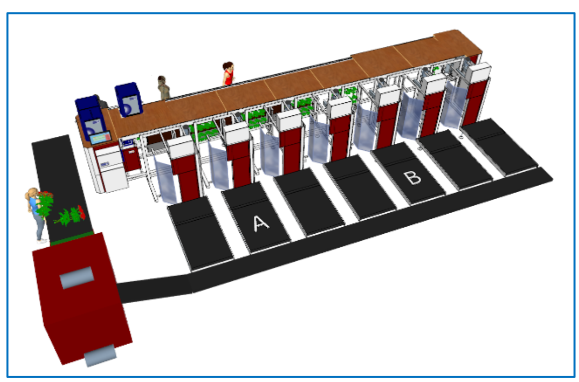


Figure 2-12 priority rules example

this way there will be no empty places on the V-belt

3. When a buffer belt (belt B) is transferring bunches to the V-belt, or bunches originating from buffer belt B are still on the V-belt and another buffer belt (Belt A), closer to the sealer is full and is waiting until it is allowed to transfer bunches to the V-belt, buffer belt A starts transferring bunches as soon as the last bunch of buffer belt B passed buffer belt A.

4. When a buffer belt (Belt A) is transferring bunches to the V-belt and more other buffer belts are full and waiting until they may transfer bunches to the V-belt, the first buffer belt in row, counted from the sealer, upward from belt A, has priority to the other buffer belts waiting. Depending on the situation rule two or three occurs.

In Figure 2-13 the process flow chart of the buffer belts is shown. The blocks are discussed below:

3.1a If the bunch that is dropped on the small buffer belt is the fifth bunch. The number of bunches for one bucket is reached.
3.1b Is the bunch that has arrived the first bunch?
3.1c Is the large buffer belt available?
3.1d If the large buffer belt is not available, the bunch stays on the small buffer belt. Therefore the small buffer belt is not available until the bunch is removed.
3.1e Wait until large buffer belt is available
3.1f The bunch is moved from the small buffer belt to the large buffer belt
3.1g the small buffer belt is empty again and is therefore available again.
3.2a All five bunches are moved until the first bunch is next to the V-belt. The time to move the bunches is dependent on the length of the buffer belt.
3.2b The large buffer belt is not available anymore. It is already occupied with five bunches
3.2c The small buffer belt is not available anymore until two bunches are removed from the large buffer belt.
3.3 Wait until V-belt is available. The availability is dependent on the afore mentioned priority rules.
3.4a One bunch is moved from the large buffer belt to the V-belt.
3.4b Wait until V-belt moved one location and an empty piece of the V-belt is aligned to the large buffer belt.
3.4c is the large buffer belt empty?
3.4d Is the number of bunches on the large buffer belt ≤3? In other words, are two bunches moved from the large buffer belt to the V-belt jet?
3.4e The small buffer belt is available again.
3.4f The large buffer belt is available again.

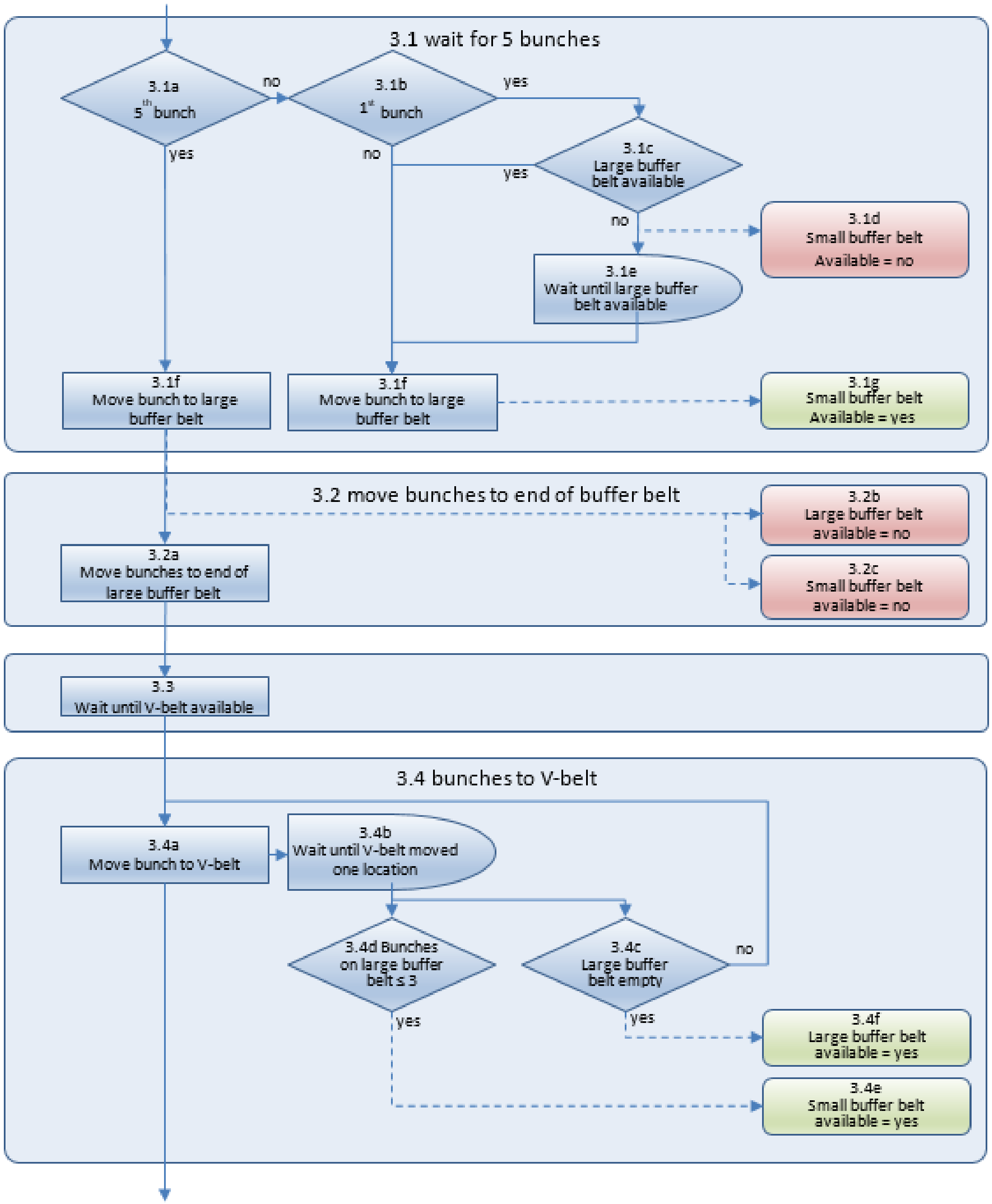


Figure 2-13 Process flow chart of buffer belt

2.4 V-belt

The V-belt transfers the bunches from the large buffer belts to the sealer. The V-belt is shown in Figure 2-11 on the left side. The space between two bunches on the V-belt is equal to the space between two buffer belts. The V-belt moves in steps of the same size. In Figure 2-14 the process flow chart is shown.

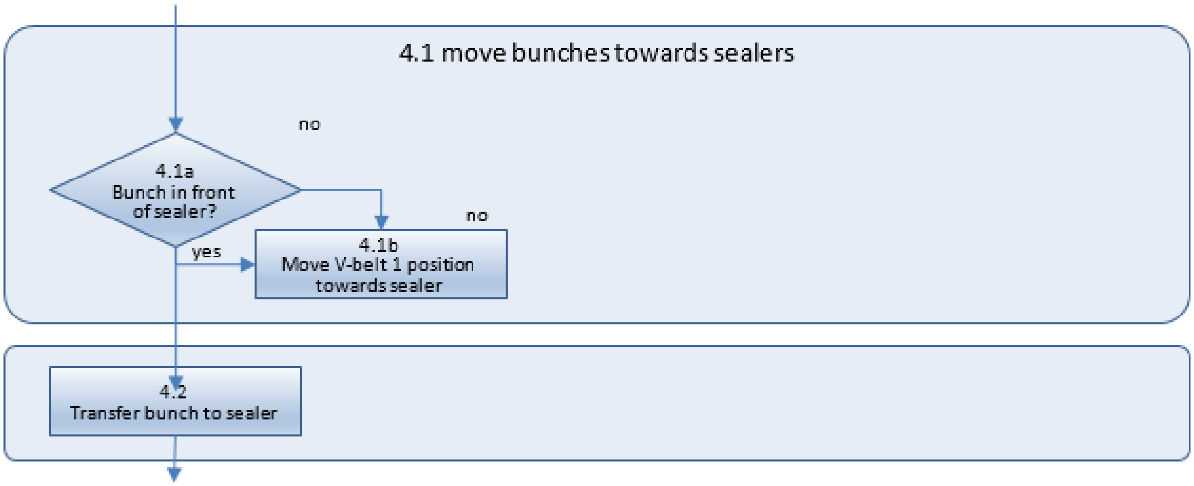


Figure 2-14 process flow chart of V-belt

The scheme is discussed below

4.1a Is there a bunch in front of the sealer?

4.1b The bunches on the V-belt are moved one position towards the sealer.

4.2 A bunch is transferred to the sealer

2.5 Sealer

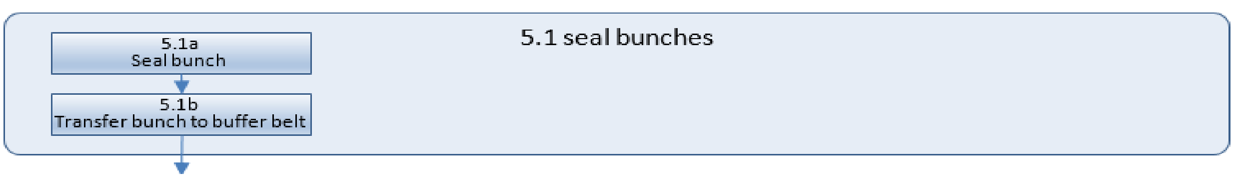


Figure 2-15 process flow chart of sealer

A picture of a sealer is shown in Figure 2-16. The bunches enter the sealer via the V-belt (at the right of the picture) and leave the sealer on a buffer belt (at the left of the picture). The sealers puts a plastic wrapping around a bunch of roses. In Figure 2-15 the process flow chartis shown. The process flow chartis discussed below:

5.1a A plastic wrapping is put around the bunch.

Figure 2-16 Sealer

5.1b The bunch is transferred from the sealer to the buffer belt.

Figure 2-17 process flow chart of buffer belt and employee

2.6 Buffer belt & Employee

in Figure 2-18 the buffer belt after the sealer and employees are shown. The employees do a last quality check and grabs five bunches of the same quality class from the belt and puts the bunches in a bucket. The process flow chartis shown in Figure 2-17.

6 Wait until five bunches are on the buffer belt

7.1 The employee checks the quality of each bunch
7.2 The employee grabs 5 bunches of the same quality class
7.3 The employee puts the 5 bunches in a bucket.

Figure 2-18 buffer belt after sealer & employees

3 Bottleneck identification

A number of bottlenecks is identified during the analyzation of all processes and machines of the rose sorting system. In the current situation, the theoretical maximum capacity is similar to the fork flow. In the optimal case every fork is used in every round and all roses are accepted by the bunching stations. However, in reality the sorting capacity is not equal to the fork flow. The capacity is limited for multiple reasons.

The identification of the bottlenecks is executed by analyzing all processes shown in the process flow charts in chapter 2 (Figure 2-6, Figure 2-10, Figure 2-13, Figure 2-14 and Figure 2-15). All blocks are displayed in the Table 3-1.

Block number	Block name	Block type	Influence capacity	Restriction	Bottleneck
0	Attach rose	process	yes	yes	filling ratio & number of attachment locations
1.1	Classify rose	process	yes	no	
1.2	Assign destination to rose	process	yes	yes	fork flow
1.3a	Destination = 1 to n	decision	no		
1.3b	Station 1 to n available	decision	no		
1.3c	Transfer to station 1 to n	process	yes	no	
2.1a	7th rose	decision	no		
2.1b	3rd rose	decision	no		
2.1c	10th rose	decision	no		
2.1d	Change collector fork	process	yes	yes	bunching station busy
2.1e	3 bunches present	decision	no		
2.1f	Fork change allowed	decision	no		
2.1g	Wait until bunches < 3	delay	no		
2.1h	Bind stems together	process	yes	yes	bunching station busy
2.1i	Station 1 Available = no	control	no		
2.1j	Wait until fork change allowed	delay	no		
2.1k	Station 1 Available = yes	control	no		
2.2	Cut stems	process	yes	no	
2.3a	Mechanical arm available	decision	no		
2.3b	Wait for mechanical arm	delay	no		
2.3c	Fork change Allowed = no	control	no		
2.3d	Mechanical arm Available = no	control	no		
2.3e	Grab bunch with mechanical arm	process	yes	yes	bunching station busy
2.4a	Move mechanical arm above buffer belt	process	yes	yes	bunching station busy
2.4b	Fork change Allowed = yes	control	no		

2.5a	Small buffer belt available	decision	no		
2.5b	Wait for space available on buffer belt	delay	no		
2.6a	Drop bunch on small buffer belt	process	yes	yes	bunching station busy
2.6b	Move mechanical arm back	process	yes	yes	bunching station busy
2.6c	Mechanical arm Available = yes	control	no		
3.1a	5th bunch	decision	no		
3.1b	1st bunch	decision	no		
3.1c	Large buffer belt available	decision	no		
3.1d	Small buffer belt Available = no	control	no		
3.1e	Wait until large buffer belt is available	delay	no		
3.1f	Move bunch to large buffer belt	process	yes	no	
3.1g	Small buffer belt Available = yes	control	no		
3.2a	Move bunches to large buffer belt	process	yes	yes	belt system flooding
3.2b	Large buffer belt Available = no	control	no		
3.2c	Small buffer belt Available = no	control	no		
3.3	Wait until V-belt available	delay	no		
3.4a	Move bunch to V-belt	process	yes	yes	belt system flooding
3.4b	Wait until V-belt moved one location	delay	no		
3.4c	Large buffer belt empty	decision	no		
3.4d	Bunches on large buffer belt <= 3	decision	no		
3.4e	Small buffer belt Available = yes	control	no		
3.4f	Large buffer belt Available = yes	control	no		
4.1a	Bunch in front of sealer	decision	no		
4.1b	Move V-belt 1 position towards sealer	process	yes	no	
4.2	Transfer bunch to sealer	process	yes	no	
5.1a	Seal bunch	process	yes	yes	belt system flooding
5.1b	Transfer bunch to buffer belt	process	yes	no	
6	Wait for 5 bunches	delay	no		
7.1	Check quality	process	yes	yes	belt system flooding
7.2	Grab 5 bunches	process	yes	yes	belt system flooding
7.3	Put bunches in bucket	process	yes	yes	belt system flooding

Table 3-1 bottleneck analysis

In the first column the number of the block in the process flow charts is shown. In the second column the name of the blocks is shown. In the third column the block type is shown. Each process block may influence the capacity (column 4). The decision blocks, delay blocks and control blocks cannot influence the capacity. However not all process block do influence the capacity. In some cases, multiple processes occur within the same time and only one process is limiting the capacity. For instance within the time of one fork (3600/Ff seconds) the quality class of one rose is assigned by IRISS, the destination of one rose is assigned and one or multiple roses are transferred from the sorting machine to a bunching station. In this case the process of assigning the destination to a rose is the process that cause a restriction for the fork flow (column 5). All processes that do cause a limitation for the sorting capacity are divided in five categories which are defined as the five bottlenecks. The bottlenecks are discussed below.

Filling ratio

The filling ratio R_f is defined in equation 1 (page 7). The camera measures whether a fork is empty or filled. The filling ratio is calculated over a certain period of time. A high filling ratio may be caused by two reasons:

- The employees attach many roses.
- The system performs badly and many roses are rejected at the bunching stations and will stay another round in the sorting machine.

All greenhouses have a filling ratio between 50% and 85%. The effectiveness off adding more employees is limited. In Figure 3-1 a situation with three attachment employees and with no retour roses is illustrated. For employee one all forks are empty and as soon as he has a rose ready to attach, he may place it on an empty fork. However at the location of employee 3, a large part of the forks is already filled. When employee 3 has a rose ready to attach to a fork he has to wait until an empty fork passes by. Therefore employee 3 has a lower attachment capacity than employee one. The higher the number of attachment employees, the lower the attachment capacity of the last employee will be.

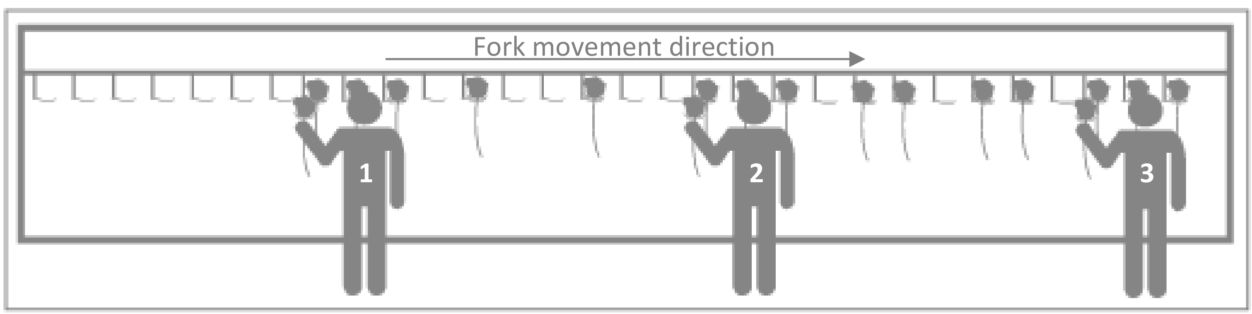


Figure 3-1 rose attachment employees

Fork flow

The speed of the sorting machine is adjustable and is defined as the fork flow (equation 1, page 7). The maximum speed in the old machine was about 9500 forks / hour. With the introduction of IRISS the machine speed may be higher due to faster image processing. However the old computer still executes the machine control and still limits the maximum speed. At this moment the maximum speed is 12000 forks/hour. In the future the old computer will be replaced and the maximum fork flow will be increased

The danger is that the speed of the forks will be too high whereby the attachment of the roses by the employees will cause damage to the roses.

Bunching station busy

There are some actions in a bunching station during which the bunching station cannot accept any roses. These are the following actions:

- Collector fork change
- Wrap a string around the stems of a bunch of roses.
- Move mechanical arm out. (During these actions no fork change can occur. Roses may be accepted until collector fork is full)
- Error state

The times desired for these actions may be changed through the machine settings and through compressed air valves.

Belt system flooding

The belt system is defined as the buffer belts, the V-belt and the buffer belt after the sealer together. There are multiple causes for that the belt system flooding. When a buffer belt is full, the accessory bunching station is able to accept roses until the bunching station is also full, afterwards it will reject all roses. This has an effect to the capacity of the entire machine. When roses are rejected by a bunching station they will remain in the sorting machine for another round. When a bunching station is rejecting roses for a long time, many forks of the sorting machine will be occupied with roses belonging to that bunching station. Three reasons for flooding are identified:

- Multiple buffer belts are full in a short time span. Buffer belts are waiting to have access to the V-belt.
- An error occurs. The belt system is equipped with many photocells to keep an eye on the bunches on the belts. When something uncontemplated happens the system will be in error state until it is reset by an employee.
- After the sealer there is a buffer belt that may also be flooded. When the employees are not fast enough with removing bunches from the buffer belt after the sealer to keep up with the sealer, this buffer belt will be filled up entirely. This is noticed by a photocell and the entire V-belt will stop moving until some bunches are removed.

Number of rose attachment locations

At first it seems that there is not a variable number of attachment locations. However adding more attachment locations would increase the capacity of the sorting system. The number of rose attachment locations is therefore also considered to be a bottleneck. At first it is explained what an extra rose attachment location is.

In the current situation there is always one attachment location with room for 2 to 5 people to attach roses. In this case all bunching stations are aligned next to the sorting machine. This situation is shown in Figure 2-12. Another option is to attach roses and then align half of the bunching stations to the sorting machine. Then add a second attachment location to fill up the empty forks of the sorting machine and then locate the other half of the bunching station. This setup is shown in Figure 3-2. With this setup some forks may be used two times within the same round. To realize this setup, software has

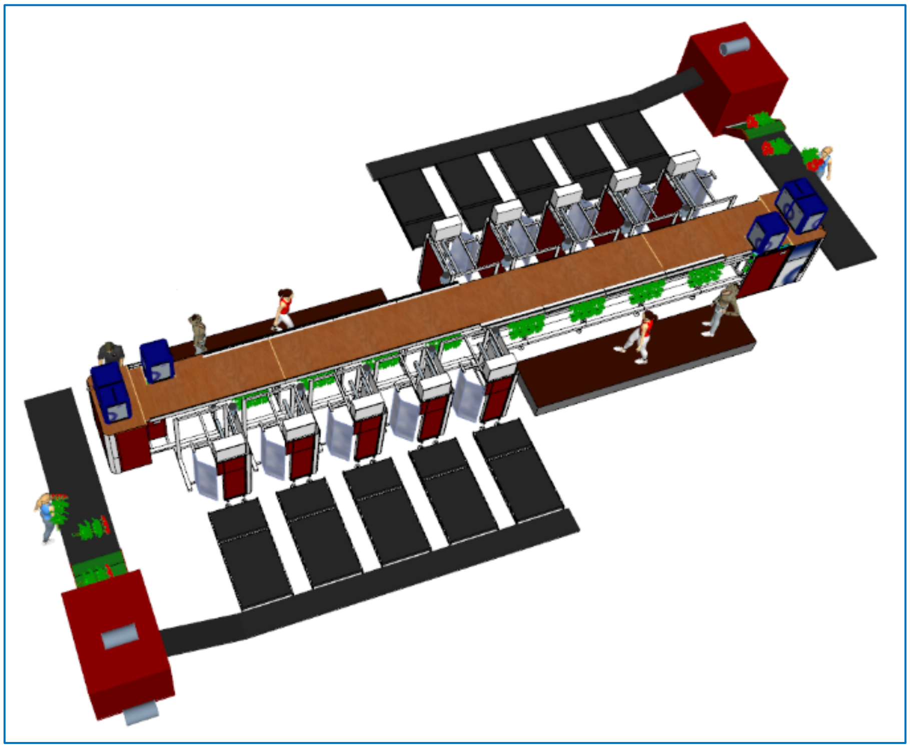


Figure 3-2 sorting system with two attachment locations

to be adapted and a second camera system has to be installed at the other end of the sorting machine.

4 Simulation model

In chapter 4 the setup of the simulation model is discussed. At first it is explained what a discrete event simulation is and which software is used (section 4.1). Second, an overview of all subsystems of the simulation model is provided (section 4.2). Then the inputs and outputs of the simulation model are discussed(section 4.3), followed by the description of the entities in the simulation model (section 4.4). Finally all subsystems (machines in reality) are discussed elaborate in section 4.5.

4.1 Discrete event simulation

In discrete event simulation (Cassandras & Lafortune, 2010) or a discrete event system, traffic units (entities) flow through the system(Schriber & Brunner, 2005). These entities respond to events. Events are occurrences that change the state of the model. Many events may occur in a model, it starts with the event of creating entities. The entities move between servers, where operations are executed and queues where the entities wait. Further there are control elements that control the logic and delays based on the systems state.

The Matlab environment using Simulink and SimEvents is used to construct the simulation model. This is a commonly used program for discrete event modelling (Gray, 2007; Van ’t Ooster, Bontsema, Van Henten, & Hemming, 2012). In Simulink a mathematical model may be constructed as a block diagram. The library SimEvents in simulink contains the blocks where entities flow through.

4.2 Subsystem overview

Similar to reality the simulation model is with standard elements that represent the standard machines in reality. In this way each composition of standard machines may be modelled. To illustrate the complete simulation model, the sorting system composition shown in Figure 2-1 is used. This sorting system constist of one sorting machine, five bunching stations, five buffer belts and one sealer. The overview of the subsystems is depicted in Figure 4-1. As you may see, each sealer, V-belt, buffer belt and bunching station consists of only one subsystem. However the sorting machine consists of multiple subsystems. In reality the sorting machine is modular and may exist in different lengths. To be able to mimick this modularity, the sorting machine is also modular in the simulation model. The composition of each subsystem is explained in section 4.5.

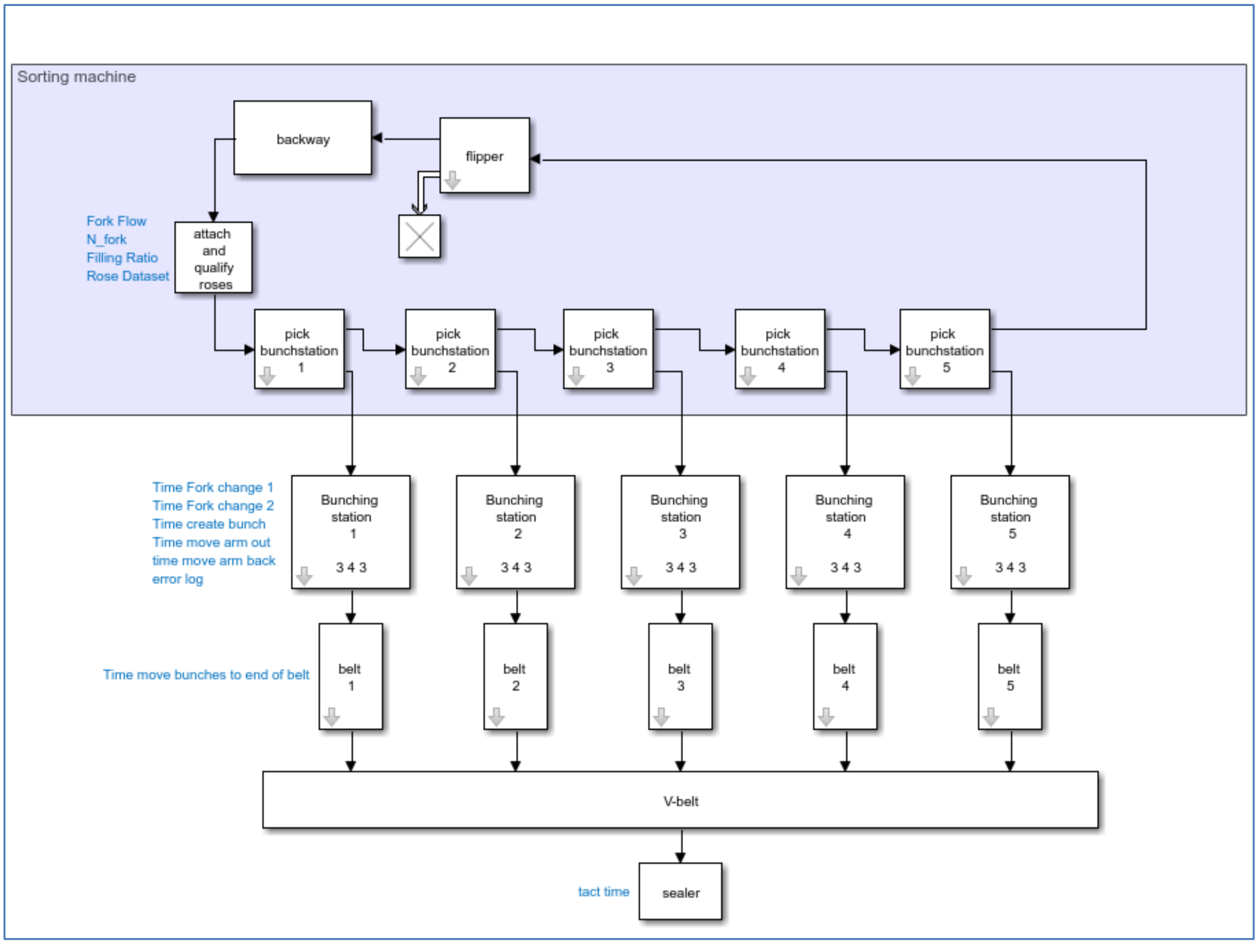


Figure 4-1 simulation model overview

4.3 Inputs and outputs

In Figure 4-1 the composition of the sorting system is shown. This composition may be different for each simulation depending on the sorting system that has to be simulated. A number of variables may be changed. These variables (inputs) are displayed in blue in Figure 4-1 next to the subsystems and listed below. In the next section these variables are discussed and how they may be changed.

Inputs:

Sorting machine

- Fork flow
- n_fork
- Filling Ratio (*Rf)*
- Rose Datasheet

Bunching station

- Time fork change 1
- Time fork change 2

- Time create bunch
- Time move arm out
- Time move arm back
- error log

buffer belt

- time move bunches to end of belt

V-belt

- no input variables

Sealer

- no input variables

outputs:

The output of the simulation consists of three components.

- The time to process all roses present in the rose dataset.
- The number of roses offered, accepted and rejected per bunching station.
- The number of roses rejected per reason per bunching station.

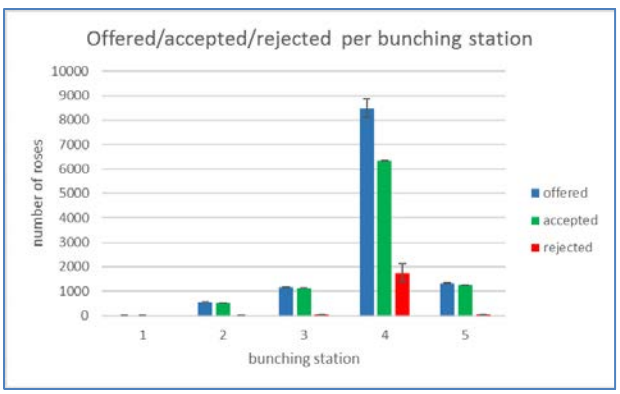


Figure 4-3a) number of roses offered, Accepted and Rejected per bunching station

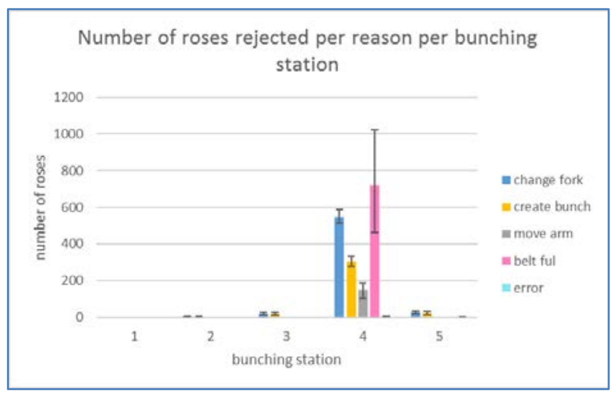


Figure 4-3b) number of Roses Rejected per bunching station per reason

Examples of the two last mentioned outputs are shown in in Figure 4-3a and Figure 4-3b

For the number of roses rejected per reason, five reasons are defined:

- Change fork: the collector fork is being changed
- Create bunch: a bunch is being created
- Move arm: the mechanical arm is moving outwards
- Belt full: the buffer belt accessory to the bunching station is full
- Error: the bunching station is in an error state

4.4 Entities

In the simulation model multiple entities exist. Namely roses, bunches of roses and forks.

Rose

The entity rose flows through the entire simulation model. The entity has two attributes

- Destination
- Pick_bunchstation

In the Destination attribute, the bunching station number where the rose is sent to, is saved. In many locations in the block system routing choices have to be made. For instance, the rose has to stay in the sorting machine or the rose has to go to the bunching station. This may be done through the attribute Pick_bunchstation. The attribute is edited at many points in the simulation model and can be set to 0 or 1.

Bunch

In the bunching stations ten roses are bundled to a bunch. In the simulation the bundling also occurs. A Bunch entity is a batch entity that consist of ten Rose entities.

Fork

At the start of the simulation the right number of Forks is generated. The Forks keep flowing in circles through the sorting machine. If a Rose is attached to a Fork, a Rose and a Fork form together a composite entity. If a Fork is empty, it still contains a Rose entity but the Destination attribute is set to 0.

Composite entity Fork and Rose

In the sorting machine, the roses are attached to forks. To make sure the entities travel together in the simulation, a composed entity is created. This entity contains one Fork entity and one Rose entity. If a Fork contains a Rose entity and the Destination attribute of the Rose entity is zero, it means that the Fork is empty and there is no Rose present.

4.5 Description per subsystem

In section 4.5 the simulation model is discussed elaborately per subsystem. Per subsystem an overview of the variables and how to adapt them is provided. Also an overview of the Simulink block scheme is provided. The blocks are grouped in the purple rectangles and explained in text. Each group of blocks represents one or multiple of the processes of the process flow chart in chapter 2. The accessory numbers of the process flow scheme blocks are displayed under the purple rectangles. A distinction has been made to the blocks where the entities flow through, from the SimEvents toolbox, and the other blocks. The other blocks mostly contain logic to control the SimEvent blocks. The SimEvent blocks are displayed in blue, the remaining blocks are displayed in white.

In Figure 4-1 it is shown that the subsystems Pick bunchstation, Bunchstation and Buffer belt have a number assigned. In the description and figures the subsystems with the number 4 is shown.

Sorting machine

As shown in Figure 4-1, The Sorting machine is modeled with four separate subsystems. In this way it is possible to extend the Sorting machine, to be able to make a setup with the desired number of outlets. The following subsystems are present in the sorting machine:

- Attach and qualify roses: in this subsystem the Forks and the Roses are generated.
- Pick bunchstation: for each Bunching station there is a Pick bunchstation subsystem. In the Pick bunchstation subsystem, it is determined if the Roses that pass through, has to be transferred to the accessory Bunching station or stay in the Sorting machine.
- Flipper: represents the flipper. In this subsystem, all Roses that are not classified in a certain category are flipped out.
- Backway: represents the part of the sorting machine where no machines are aligned. It contains no action, it is only there to be able to have the desired number of Forks.

These subsystems are discussed in separate sections.

Sorting machine - Attach and qualify roses

Variables

The variables for the Attach an qualify roses subsystem are:

- fork flow:

In a separate group in the subsystem Attach and qualify roses, a “pulse generator” is present. This “pulse generator” creates a pulse every time a rose passes by the camera. This signal is called Nextfork and is imported in multiple subsystems via a “from” block.

- number of forks:

The number of forks present on the sorting machine is dependent on the length of the sorting machine. The number of forks may be adapted via the Matlab user interface. The variable is imported in the Attach and qualify roses subsystem via the “constant” block in the group Create forks.

- rose dataset:

The rose dataset is set in the Matlab user interface. The rose dataset is a list that contains all roses that has to be processed in the simulation and the corresponding quality class and destination (bunching station the rose is sent to). This destination of the rose is set via the attribute Destination in the entity Rose. This is done in the “entity generator” block in the group Generate roses.

- filling ratio (*Rf*):

The filling ratio is determined in the Matlab user interface. The filling ratio is imported in the Attach and qualify roses subsystem via the “constant” block in the group Filling ratio.

Next fork

This group is separate from the rest of the blocks. In this group the signals Nextfork and forks_passed are created. Between each pulse one Rose and Fork leaves the Attach and qualify roses subsystem. The signal is created with a “pulse generator” block with a period (P) that is defined as follows:

(Equation 2)

$$P = \frac{3600}{ff}$$

- P = Period of the pulse generator [s]
- ff = Number of forks that pass the camera within an hour [forks]

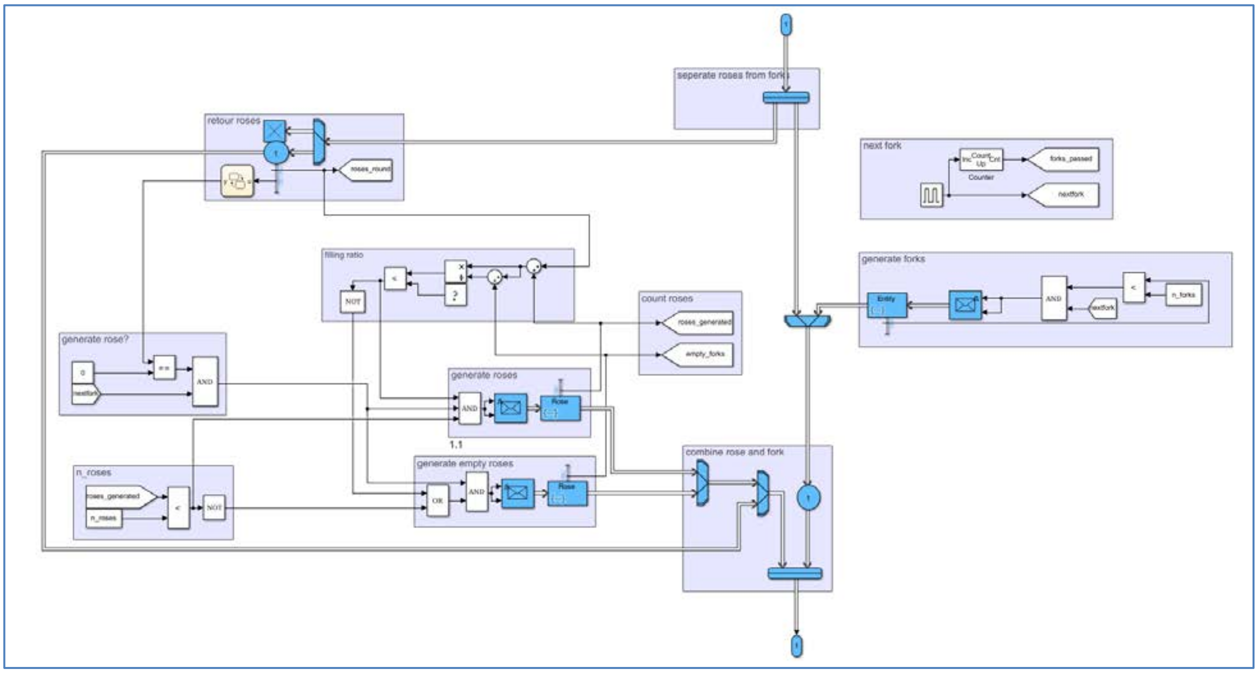


Figure 4-4 Simulink block scheme of the Attach and qualify roses subsytem

This signal is sent directly to the “goto” block, to be able to use it in any other location in the simulation model. A separate signal Forks_passed is created. This signal contains a number with the number of forks that left the Attach and qualify roses subsystem. To create this signal a “counter” is placed between the “pulse generator” and the “goto” blocks.

Separate roses from forks

Before and after the Attach and qualify roses subsystem, the Rose entity and Fork entity travel together as a composed entity. The first step in the Attach and qualify roses subsystem is to separate the entities to be able to treat them separately.

Retour roses

This group starts with an “entity output switch”. If the Destination attribute of the Rose entity is zero, it means there is actually no Rose. In this case, the upper output of the “entity output switch” is used. This leads to an “entity terminator” and the entity is destroyed. If the Destination of the Rose is not zero, it means the rose could not be accepted by a Bunching station in the last round and the Rose has to make another round in the Sorting machine. In this case the lower outlet of the “entity output switch” is used.

The entity goes to an “entity server” where it stays for 0.1 seconds. From this server two signals are sent. One signal contains a 0 if there is no entity present and a 1 if there is an entity present. The other signal contains a number that counts the number of entities departed from the entity server. Via the “stateflow” block the signal for the presence of an entity is prolonged until the next Rose enters the “entity output switch”.

Generate rose?

The output signal of this block group contains pulses. On each pulse, a Rose is generated in one of the two “entity generators”.
The signal Nextfork is used via a “from” block. However, not on every pulse a Rose is generated. If there is a retour Rose, no Rose has to be generated. This is reached via the “constant” (0) block, the “compare” (==) block and the “AND port”.

N_roses

If all Roses that are present in the dataset are generated, the simulation has to continue until all Roses are processed. From that same moment, no Roses will be generated anymore. In other words only empty Rose entities has to be generated from that moment.
The number of Roses generated is compared to the number of Roses in the dataset (N_roses). If the number of Roses generated is lower than or equal to the number of Roses in the dataset, the signal to the group Generate roses is 1 and the signal sent to the group Generate empty roses is 0. If the number of Roses generated is higher than the number of Roses in the dataset, the signals are the other way around.

Filling ratio

In this group, the actual filling ratio in the simulation is at each time step compared to the desired filling ratio. The actual filling ratio is the sum of the Roses_generated signal and the Roses_round signal divided by the total number of Forks that passed.

$$Rf_{actual} = \frac{Nr_{generated} + Nr_{round}}{Nr_{generated} + Nr_{round} + Nf_{empty}} \quad \text{(Equation 3)}$$

- Rf_{actual} = Actual filling ratio of the simulation
- $Nr_{generated}$ = Roses generated: current number of generated roses
- Nr_{round} = Current number of roses that made a second round in the sorting machine
- Nf_{empty} = Current number of empty forks that made a round in the sorting machine

This number is compared to the desired filling ratio. If the actual filling ratio is higher than the desired filling ratio, the signal to the group Generate roses is 1 and the signal sent to the group Generate empty roses is 0. Otherwise, it is the other way around.

Generate roses

Three signals enter the "AND port", which means that three conditions have to be met to generate a Rose:

1) There is no retour Rose.
2) The actual filling ratio < the desired filling ratio.
3) The number of Roses generated < the total number of Roses in the rose dataset.

Via the "send message" block the pulse signal is converted to a discrete event, so that it may be used by the "entity generator". From the "entity generator" a signal is sent with the number of entities that left the "entity generator". The attributes of the entity are set in this "entity generator". The Pick bunchstation attribute is set to 0 and the Destination attribute is the first unused Destination from the rose dataset.

Generate empty roses

To generate an empty Rose entity, two conditions has to be met:

1) There is no retour Rose.
2) Actual filling ratio > desired filling ratio OR number of Roses generated ≥ total number of Roses in the rose dataset.

The attributes of the Rose entities are set as follows. Destination = 0 and Pick bunchstation = 0.

Count roses

This group contains two "goto" blocks:

- The number of Roses generated.
- The number of empty Roses generated.

In this way these signals may also be used in other parts of the simulation model.

Generate forks

At the start of the simulation no entities flow into the subsystem Attach and qualify roses and the Fork entities firstly has to be generated. On each pulse of the Next fork signal, a Fork entity is generated. This is executed until the number of Forks generated is ≥ n_forks (number of Forks in the machine). From that moment on the Forks that go round will arrive from the inlet of the subsystem Attach and qualify roses.

Combine Rose and Fork

In the two "entity input switches" the flows of empty Roses, normal Roses and retour Roses are combined to one flow. For the "composed entity creator", both entities (Fork and Rose) has to arrive at the exact same time. Because the Roses stay in an "entity server" for a little time earlier in this subsystem, the Fork must also have the ability to stay in an 'entity server" for a while.

Sorting machine - Pick bunchstation

Variables

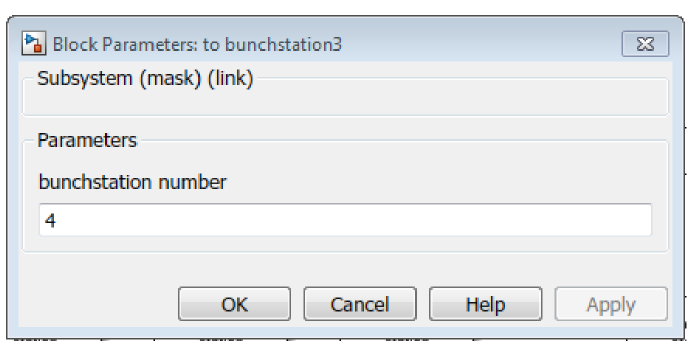


Figure 4-5 dialog window of Pick bunchstation subsystem

There are no variables in the Pick bunchstation subsystem. However as you may see in Figure 4-1, each Pick bunchstation subsystem has a number, that corresponds with the number of the accessory Bunching station and Buffer belt. This number has to be set through a dialog window (Figure 4-5).

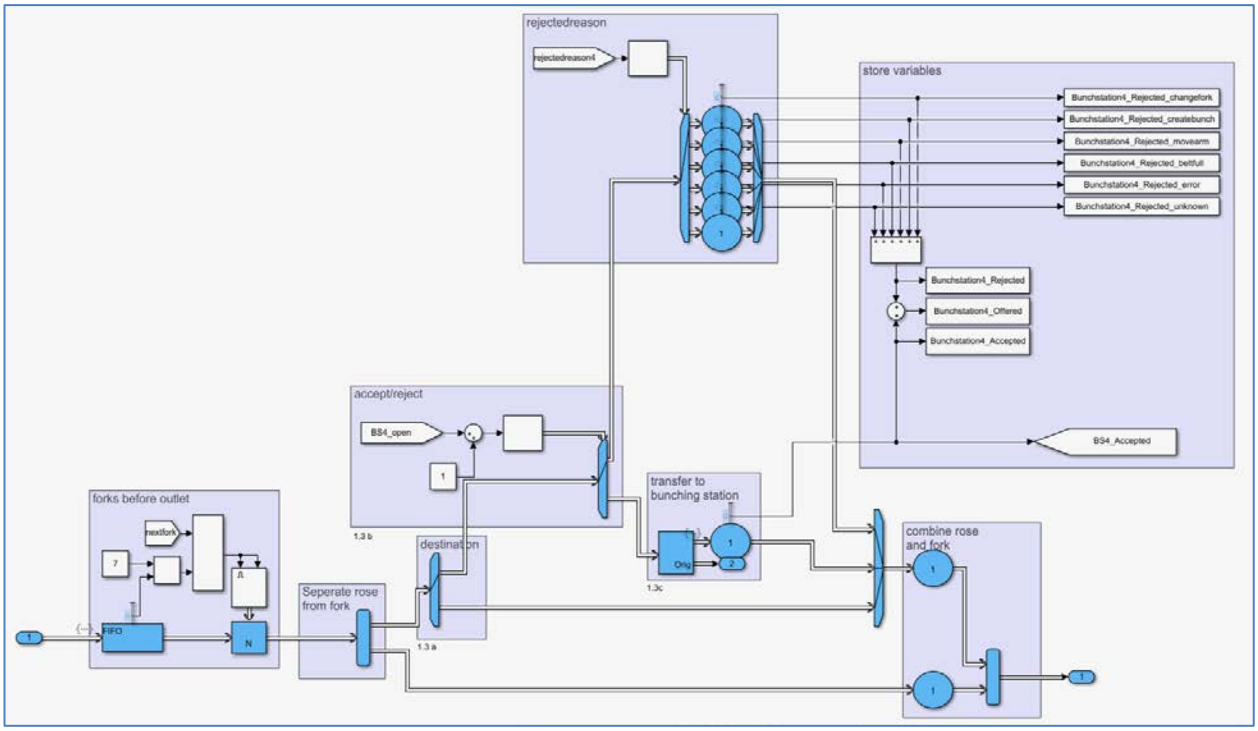


Figure 4-6 Simulink block scheme of Pick bunchstation subsystem

Forks before outlet

Between two Bunching stations there is a certain distance, that contain a certain number of Forks. These Forks are in the "queue" of this group. On each Next fork pulse one Fork may flow through the "entity gate", except in the beginning. At first 7 Forks has to be present before the first may flow through the "entity gate".

Separate Rose from Fork

In this block the composed entity is split up in the Rose and the Fork.

Destination

If the Destination attribute of the Rose is equal to the number of this Pick bunchstation subsystem, the Rose entity takes the upper outlet of the "entity output switch", otherwise the Rose takes the lower outlet of the "entity output switch".

Accept/Reject

A signal is sent from the accessory Bunching station subsystem (in this case Bunching station 4). If the Bunching station is available, the signal is 1. If the Bunching station is not available, the signal is 2. 1 is added to the signal via the “constant” and the “sum” blocks, because the “entity output switch” needs the signal 1 or 2 to switch between outlet one and two. Via the “send message” block the signal is converted to a signal that may be used by the “entity output switch”. If the accessory Bunching station is available, the Rose goes to the lower outlet of the “entity output switch” and if the accessory Bunching station is not available, the Rose goes to the upper outlet of the “entity output switch”.

Rejected reason

If the accessory Bunching station is not available a signal is received from the Bunching station subsystem that can vary from 1 to 5 and stands for the reason of rejection. Dependent on this signal the Rose flows through one of the “entity servers”. From these “entity servers” a signal is sent to the group Store variables that contains the number of entities departed from the “entity server”.

Store variables

The number of Roses rejected per reason, and the number of Roses offered, accepted and rejected is sent to the Matlab workspace to be able to see the results of the simulation. The number of Roses accepted by the Bunching station is also sent to a “goto” block to be able to use this signal in another part of the simulation model.

Transfer roses to bunching station

If a Rose is accepted by the Bunching station it will flow through this group. The “entity replicator” replicates the Rose entity. One of the outlets of the “entity replicator” goes to outlet 2 of the Pick bunchstation subsystem. This outlet is connected to the accessory Bunching station subsystem. The other outlet of the “entity replicator” is connected to an “entity server” that sets the two attributes of the Rose entity to 0. This represents an empty Fork.

Combine Rose and Fork

In this group there are two “entity servers” to make sure the Rose entity and Fork entity can leave at the exact same time to the “composed entity creator”. From the “composed entity creator” the Rose and Fork travel together.

Sorting machine - Flipper

Variables

There are no variables in the Flipper subsystem

Block scheme

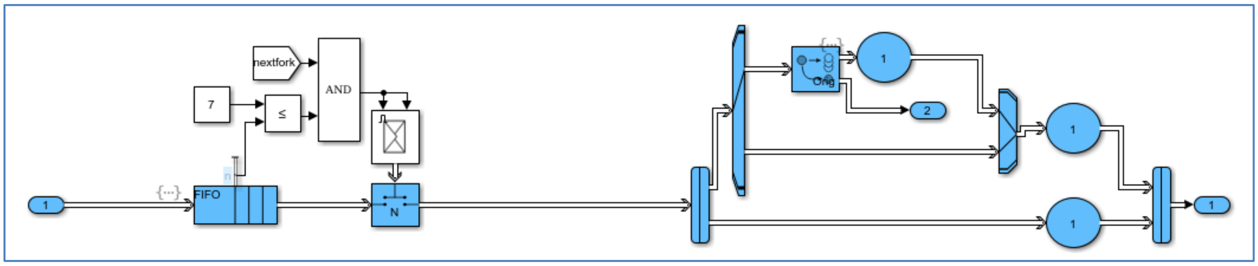


Figure 4-7 Simulink block scheme of subsystem Flipper

In Figure 4-7, the blocks are not divided into groups. This subsystems is the equal to a part of the To bunching station subsystem. It is equal, except that there is no Bunching station connected to the outlet. All roses where the Destination attribute contains another number than one of the numbers of the Bunching stations are flipped out over here. These Rose entities are terminated.

Sorting machine - Backway

Variables

There are no variables in the Backway subsystem.

Block scheme

The Backway subsystem contains an “entity queue” that contains all Forks between the last Bunching station and the Attach and qualify roses subsystem. On each pulse of the Next fork signal a Fork may go through the “entity gate” except for the first round. From the moment that all Forks are generated, (Fork passed < n_forks) the Forks start flowing through the “entity gate”. Forks passed is the number of pulses generated by the Nextfork signal.

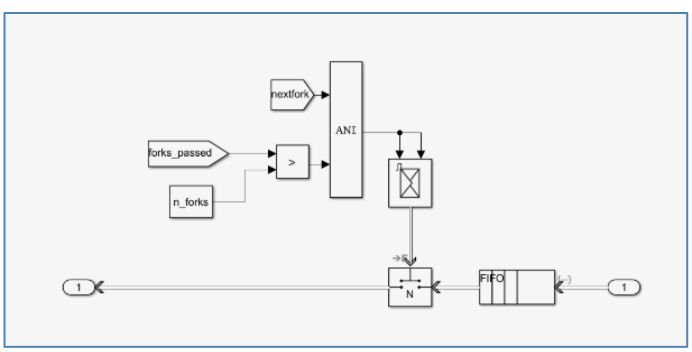


Figure 4-8 Simulink block scheme of Backway subsystem

Bunching station

Variables

The variables are:

- fork change 1 (seconds)
- fork change 2 (seconds)
- create bunch (seconds)
- move arm out (seconds)
- move arm back (seconds)
- error log

When clicking on the subsystem Bunching station, a dialog window as displayed in Figure 4-1 appears, through which the variables may be edited. Apart from the error log, all variables may be edited through this dialog window. For each Bunching station these variables may be different. There is also a check box for an error log. If an error log is used, this check box is checked. The error log has to be loaded via the Matlab user interface.

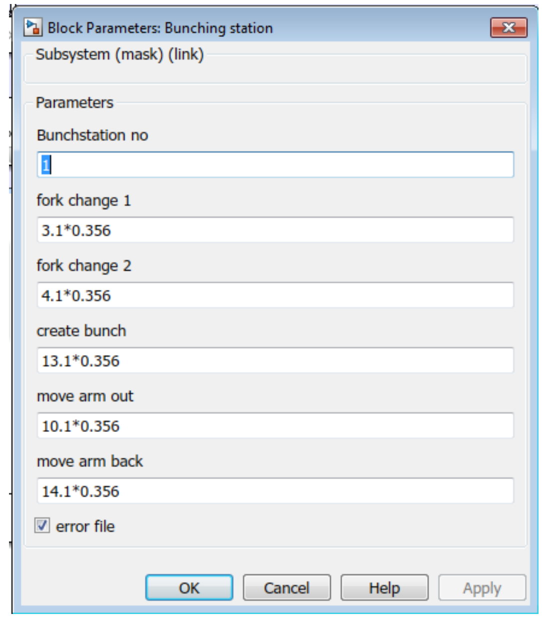


Figure 4-9 dialog window of Bunching station subsystem

Create bunch

The scheme start with a “batch entity creator”. The “batch entity creator” waits untill 10 Roses have arrived. Then the Bunch entity is created that consists of 10 Rose entities. Information about the number of Roses that are present in the incomplete Bunch is used further on. Unfortunately only information about the number of entities that remain for the next batch entity is available. With the use of a “constant” block and a “sum” block the number of entities present is calculated.

$$Nr_{present} = 10 - Nr_{remaining} \qquad \text{(Equation 4)}$$

- $Nr_{present}$ = Number of rose entities within the batch entity creator
- $Nr_{remaining}$ = Number of rose entities remaining to create a batch entity

Bunch queue

A certain number of Bunches may be hold in the Bunching station. In practice this number is four. In the simulation three Bunches may be present in the Bunch “entity queue” and one in the “batch entity creator”. Information about the number of entities in the “entity queue” is sent to the group Mechanical arm control and the group Bunching station available?.

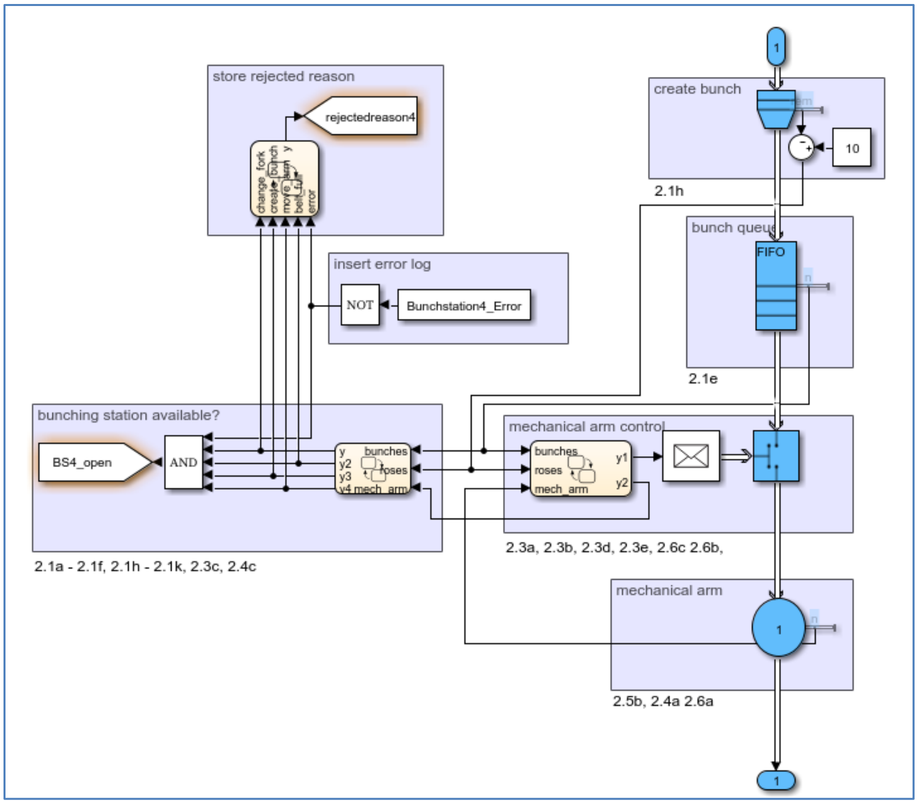


Figure 4-10 Simulink bock scheme of Bunching station subsystem

Mechanical arm control

The entrance of the Mechanical arm is controlled via an “entity gate”. This gate may be closed or opened. The “send message” block converts the signal from the “stateflow" block to a message that is usable for the “entity gate”. In the “StateFlow” block a decision is made about the closing and the opening of the “entity gate”. Information about the number of Roses in the “batch entity creator”, the number of Bunches in the Bunch “entity queue” and the occupation of the Mechanical arm is used to make the decision.

Mechanical arm

The Mechanical arm is represented as an “entity server”. If an entity enters this block it stays there for a certain period. This period is the variable time move arm out that is filled in via the mask of the Bunching station.

Bunching station available

In these blocks it is determined if the Bunching station is available. Information about the number of Roses, number of Bunches, occupation of the Mechanical arm and if the Bunching station is in error state is necessary to determine if the bunching station is available. Whether the Bunching station is available or not available is decided in the “StateFlow” block. The bunching station not being available, can have multiple causes:

- Fork change (a collector fork change is executed).
- Create bunch (bunch creation is executed).
- Move arm (mechanical arm is moving out).
- Buffer belt full (the acessory buffer belt is full and the bunch in the mechanical arm cannot be dropped on the buffer belt).
- Error (the bunching station is in error state).

For each cause there is an output of the “StateFlow” block. The information is sent to the accessory subsystem via a “goto” block.

Insert error log

In the verification, a day is mimicked. The errors in the bunching station are logged in reality and implemented in the simulation via these blocks. When an error occurs, the Bunching station is not available.

Store rejected reason

When the Bunching station is not available, the reason is known. The “StateFlow” block converts the reason to number 1 to 5. This number is sent to the “goto” block that sends the information to the corresponding Pick bunch station subsystem. Later on, this information is used to generate a bar graph displayed in Figure 4-3.

Buffer belt

Variables

The only variable for the Buffer belt is the time to move the Bunches to the end of the Buffer belt. For this time two different values are used, one that stand for a long Buffer belt and one that stands for a short Buffer belt. A long or a short Buffer belt is chosen by clicking on the Buffer belt subsystem. A dialog window appears (Figure 5-1). Also the number of the Buffer belt has to be chosen. In this case Buffer belt 4 is

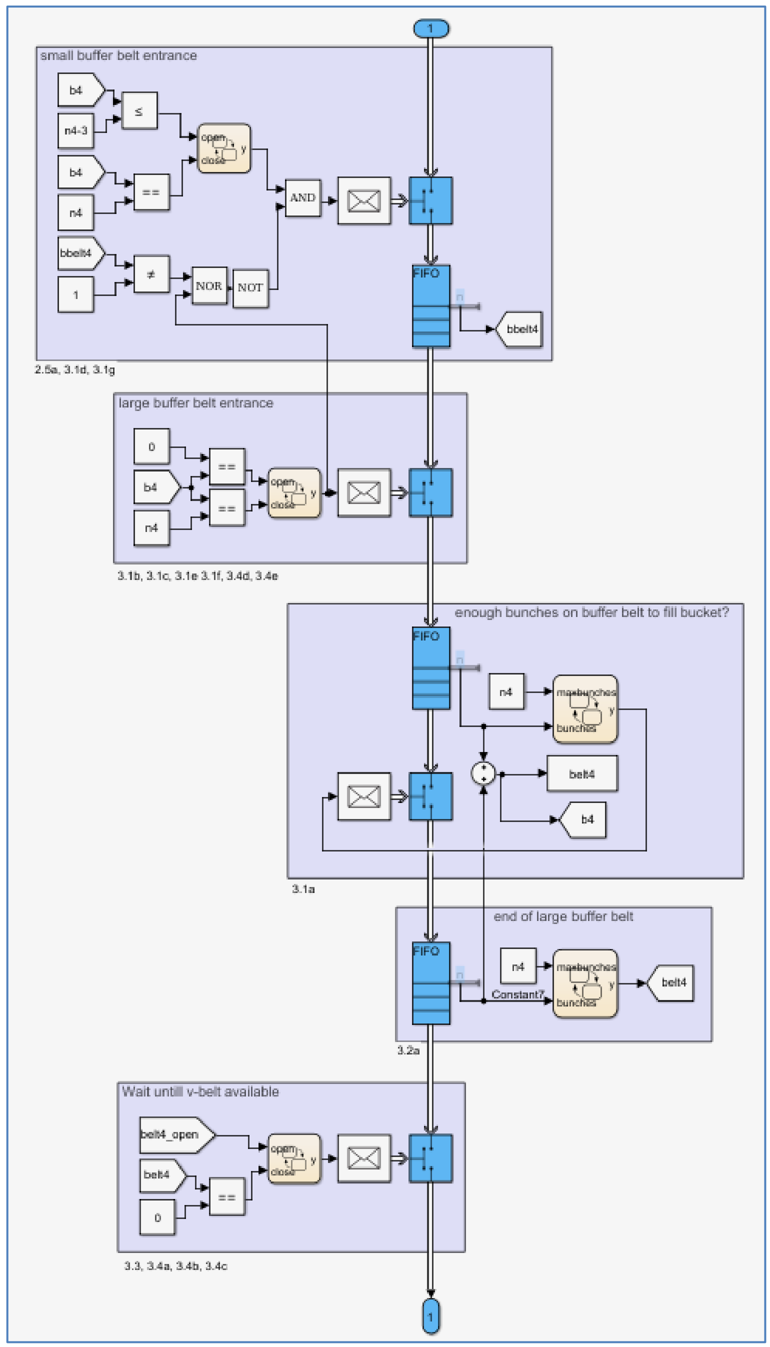


Figure 4-12 Simulink bock scheme of Buffer belt subsystem

chosen, that is connected to Bunching station 4.

Block scheme

The Simulink block scheme is shown in Figure 4-12. In this block scheme multiple signals are sent to "goto" blocks and used again via a "from" block. These signals are:

- B4: number of Bunches on large Buffer belt.
- Bbelt4: number of Bunches on the small Buffer belt.

Also a constant is imported:

- N4: total number of Bunches that may be collected on the Buffer belt.

Small buffer belt entrance

The small Buffer belt is represented by an "entity queue". From the "entity queue" a signal with the number of entities in the queue is sent to a "goto" block so that it may be used anywhere else in the simulation model. The entrance of the small Buffer belt is controlled by an "entity gate". The blocks connected to the upper entrance of the "AND port" block, represents the opening and the closing of the Small buffer belt entrance as a result of a full large Buffer belt. When the large Buffer belt is full ($b4 = n4$) the small Buffer belt entrance is closed. When two Bunches are transferred from the large Buffer belt to the V-belt ($b4 \leq n4 - 3$), the entrance of the small Buffer belt is opened again. When this last mentioned condition is met and one Bunch is dropped on the small Buffer belt, the entrance of the small Buffer belt is closed again. This is represented by the lower input of the "AND port" block.

Large Buffer belt entrance

The entrance of the large Buffer belt is controlled by an "entity gate". The gate is closed when the large Buffer belt is full ($b4 = n4$). The "entity gate" is opened when the large Buffer belt is empty (again) ($b4 = 0$).

Enough bunches on the Buffer belt to full bucket?

The large Buffer belt is represented by two "entity queues". The "entity queue" in this group, stands for the part of the large Buffer belt where the Bunches are transferred to from the small Buffer belt and the "entity queue" in the group End of large Buffer belt. When the Bunches are in this "entity queue", the Bunches may be transferred to the V-belt. The Bunches are transported to the end of the large Buffer belt when the number of Bunches to full a bucket (in this case n4) is reached.

The transport from the first "entity queue" to the last "entity queue" is controlled via an "entity gate". When the correct number of Bunches is reached ($b4 = n4$), and the time to move the Bunches to the end of the large Buffer belt has elapsed, the "entity gate" is opened. This is arranged in the "Stateflow" block. The total number of Bunches on the large Buffer belt (in first "entity queue" and last "entity queue") is added together and sent to the Matlab workspace as belt4.

End of large Buffer belt

When the number of Bunches on the End of the large Buffer belt is n4, belt4 is set to n4.

Wait until V-belt available

The transfer from the Buffer belt subsystem to the V-belt subsystem is arranged in this group with an “entity gate”.

The priority rules of the large Buffer belts to the V-belt is arranged in another subsystem. When this large buffer belt may transfer Bunches to the V-belt, the signal belt4_open is switched from 0 to 1 for a very short time and the “entity gate” will switch to open. The “entity gate” stays open until the End of the large Buffer belt is empty (b4 = 0).

V-belt

Variables

There are no variables in the V-belt subsystem.

Simulink block scheme

The V-belt is represented by “entity servers”. The V-belt moves in steps as large as the distance between two buffer belts, so next to the five large buffer belts, 5 bunches may be located on the V-belt. Each location of the V-belt is represented by an “entity server”. The Bunches stay as long in the “entity server” as the tact-time of the Sealer(5.7 seconds). Each large Buffer belt that is aligned to the V-belt is an entrance for the V-belt. The number of Bunches on each position of the V-belt is sent to the corresponding “goto” block.

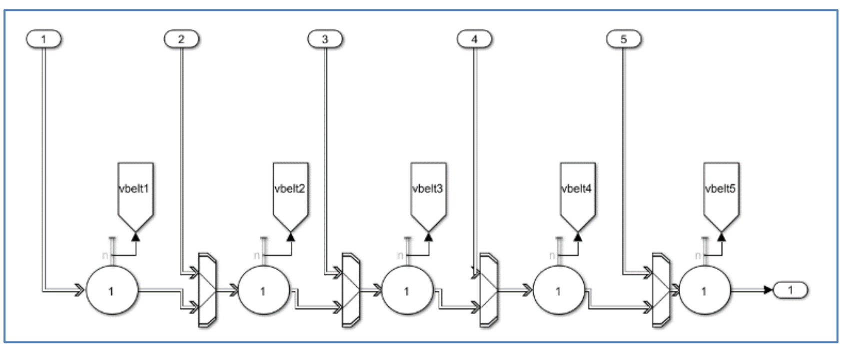


Figure 4-13 Simulink bock scheme of V-belt subsystem

Sealer

variables

There are no variables in the Sealer subsystem.

Simulink block scheme

In the Sealer, there is no action in the process. The only action done in the simulation model is

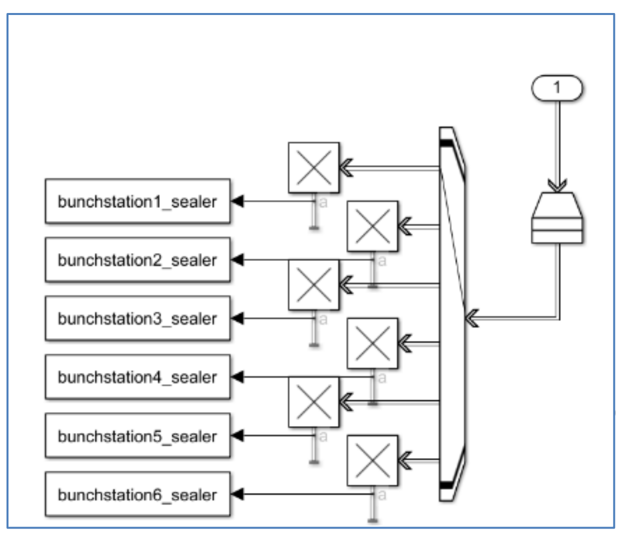


Figure 4-14 Simulink block scheme of the Sealer subsystem

counting the number of Roses arrived in the Sealer.

The scheme starts with a “batch entity splitter”. Here the Bunch entities are split up in single Roses again. In the “entity output switch” the roses are sent to one of the outlets, depending on the Destination attribute. The Destination attribute corresponds with the Bunching station the Rose has gone through. Then the Rose entities are terminated in one of the “entity terminators”. The number of Roses arrived in each “entity terminator” is sent to the Matlab workspace.

5 Verification, matching and validation and sensitivity analysis

Chapter 5 describes the verification, matching and validation of the discrete event simulation model. Verification, matching and validation is done to make sure the simulation model mimics the reality and the output variables are correct and accurate. The steps of verification, matching and validation are based on the structure of Grimmelius (2005). In the original structure Grimmelius starts from the principle that each subsystem of the model is firstly verified, matched and validated (Hilberink, 2005). In this research instead of dealing with the subsystems separately the complete model will be matched and validated. The verification is executed in two steps.

At first, in section 5.1, the existing system is described. Then the verification(section 5.2), matching(section 5.3) and validation(section 5.4) is discussed. Finally a sensitivity analysis is executed (section 5.5).

5.1 Setup of the sorting system that is used to validate the simulation model.

The sorting system of the rose grower named "Meewisse Roses B.V." is used to execute the validation. The setup of the sorting system is shown in Figure 5-1. The sorting system consist of one sorting machine, 20 bunching stations, 12 buffer belts, two V-belts and two sealers. One sealer serves bunching station 3 to 7 and the other serves bunching station 8 to 14. Bunching station 1 and 2 and 15 to 20 are not connected to a buffer belt, V-belt and sealer. The system of Meewisse is also equipped with TRSwatch (TRS is the sorting machine) in which all errors of the bunching stations are logged. This is

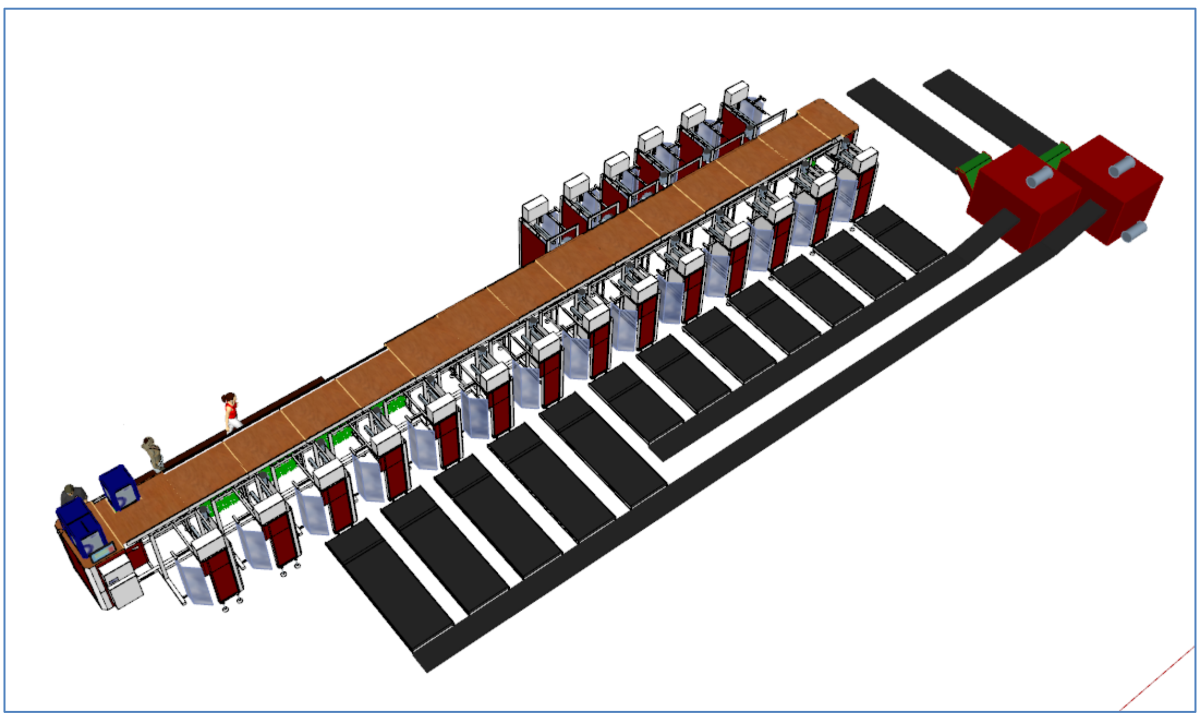


Figure 5-1 Sorting system of Meewisse B.V.

used in the simulation.

The sorting settings are very straight forward. Each quality class has only one corresponding bunching station. If a rose is rejected by the corresponding bunching station, the rejected rose will circulate in the sorting machine until the bunching station is able to accept the rose.

5.2 Verification of simulation model

According to Grimmelius (2005), verification is the initial evaluation of the models, mainly based on theoretical process knowledge. Verification should show the ability of the models to describe the physical processes. Theoretically expected trends should be visible, but as yet without demands on numerical precision. The simulation model is verified in two steps.

1) **Single bunching station:** At first the simplest situation is to verify the simulation. The simplest situation consists of a sorting machine with one, single bunching station.
2) **V-belt priority rules:** In section 2.3 the priority rules of the transfer of bunches of roses from the buffer belts to the V-belt is provided. In this step a situation with multiple bunching stations and multiple buffer belts with one V-belt is simulated. The transfer of bunches from the buffer belts to the V-belt in the simulation is compared with the priority rules.

A rose sorting system consists of five standard machines (chapter 2). With the aforementioned situations at least one of each standard machines is used. Also the logic between multiple similar machines is tested in the second situation. With the verification of these two situations the entire sorting system is considered to be verified.

Single bunching station

To be able to test a bunching station in reality, a testing situation is realized. In this situation the settings of the sorting machine are changed in a way that all roses are assigned to one single bunching station. The only restriction for accepting roses the availability of the bunching station. In this way the accepting and rejecting sequence of a bunching station may be measured. The testing situation is filmed and analyzed through a slow-motion playback. A repeating pattern for each bunch is determined. A bunch of roses exist of 10 roses, which are spread over three collector forks. The accepting-rejecting sequences is shown in Table 5-1 and Table 5-2.

The accepting-rejecting sequence of the first three bunches (Table 5-1) is different from the following bunches. After these first three bunches the bunching station is entirely filled. From the moment that the bunching station is full, the extra process of transferring a bunch from the bunching station to the buffer belt with the mechanical arm is executed in each sequence.

Accept	Reject
3	
	3
4	
	4
3	
	13

Table 5-1 Accepting rejecting sequence of single bunching station in reality.

First three bunches.

Accept	Reject
3	
	3
4	
	4
3	
	18

Table 5-2 Accepting rejecting sequence of single bunching station in reality.

Following bunches.

Different bunching stations could have a different accepting rejecting sequence. In the simulation the parameters of the Bunching station are set in a way that it should mimic reality. The results of the simulation is showed by graphs. The first bunch of both reality and simulation result are shown in Figure 5-2. It shows that the accepting and rejecting sequence matches reality. First three roses are accepted (green line), then three roses ere rejected (yellow line) etc. The number of roses offered is displayed in blue and is the sum of the accepted and rejected roses.

The bunches that follow after this first bunch also show the correct results. The same test is done for other accepting rejecting sequences. In all cases the simulation shows the correct result.

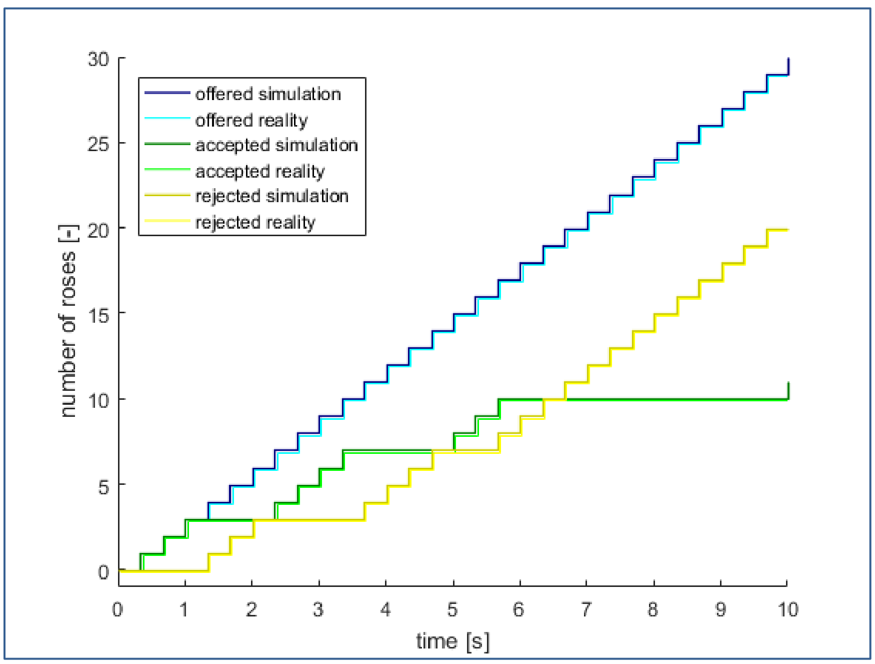


Figure 5-2 accepting rejecting sequence of single bunching station in simulation. first bunch

Priority rules belt system.

To verify the priority rules (discussed in page 12) of the belt system, a part of a system is simulated. This part contains a Sorting machine with 7 Bunching stations, 7 Buffer belts, a V-belt and a Sealer(Figure 2-12). In Figure 5-3 and Figure 5-4 is zoomed in on timespans where multiple Buffer belts are full, to analyze if the priority rules work as desired.

In Figure 5-3 a) the rules one, two and three are shown. At first the situation of rule one occurs. Buffer belt 4 is full and ready to be unloaded and immediately starts to transfer Bunches to the V-belt. Afterwards, Buffer belt 7 gets full and rule two occurs. The number of empty locations between Buffer belt 7 and Buffer belt 4 is 2, so Buffer belt 7 waits until there are only two Bunches left on Buffer belt 4 and then its starts to transfer Bunches to the V-belt. Than Buffer belt 2 gets full, and rule three occurs. There are still bunches on the V-belt, so Buffer belt 2 has to wait until the last Bunch of Buffer belt 7 is passed, then it starts transferring Bunches to the V-belt.

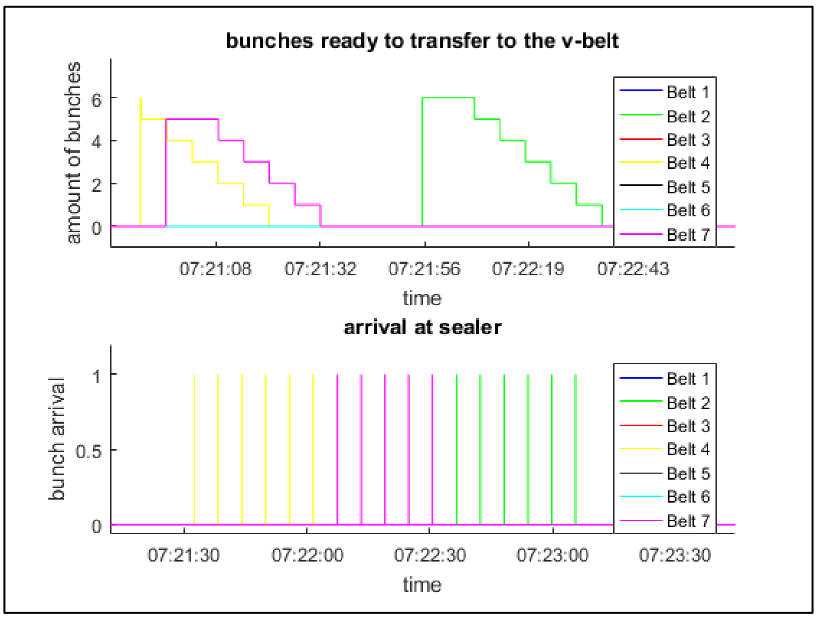


Figure 5-3

a) number of Bunches waiting to be transferred to the V-belt

b) arrival of Bunches at the Sealer

In Figure 5-3 b), the arrival of Bunches at the Sealer is shown. It is shown that the same time between each Bunch is present. This shows that there are no empty places on the V-belt.

In Figure 5-4, all priority rules, including rule four, are shown. At first Buffer belt 7 is full and immediately starts transferring Bunches to the V-belt. Then Buffer belt 5 gets full and has to wait until the last Bunch of Buffer belt 7 has passed. During this waiting period Buffer belt 2 gets full and receives priority over Buffer belt 5.

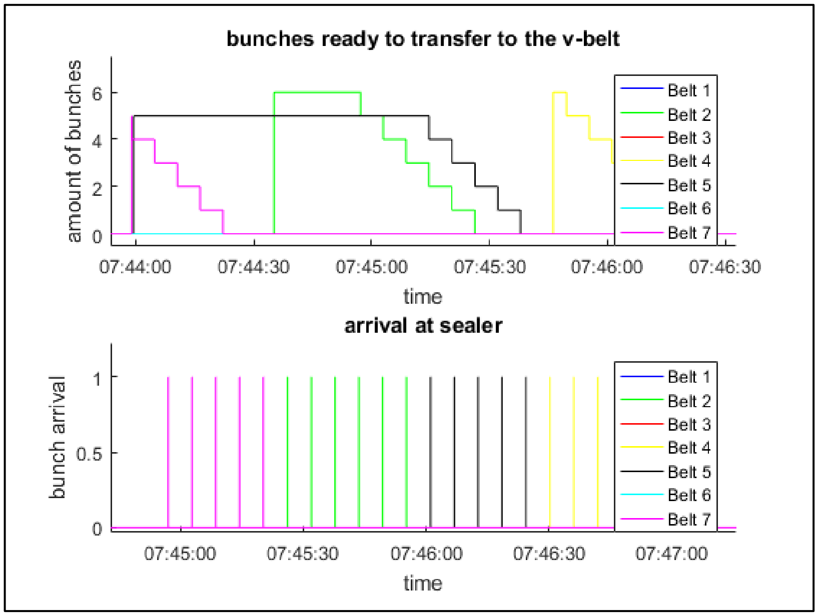


Figure 5-4

a) number of Bunches waiting to be transferred to the V-belt

b) arrival of Bunches at the Sealer

5.3 Matching of simulation model

Matching is understood to be the adjustment of parameters in the models, such that the simulated outputs approximate the measured outputs as accurately as possible over the entire operational range (Grimmelius, 2005). However since the exact reaction of the output on the adjustments of parameters is not known, because one of the goals of the research is to gather information about reaction of the sorting system in a more quantitative way. Therefore matching is executed in a more general way.

With common sense the reaction of the output parameters on adjusting the input parameters is predicted. The expectation is shown in Table 5-3 With arrows. Thereafter the change and result of each parameter is discussed per section.

Input		Output	
parameter	**Adjustment**	**Processing time**	**Number of roses rejected**
Fork flow (*Ff*)	↑	↓	↑
N_forks	↑	-	-
Filling ratio (*Rf*)	↑	↓	↑
Rose dataset	Equally divided	↓	↓
Bunching station action times	↑	↑	↑

Time to move bunches to end of belt	↑	-	-

Table 5-3 predicted reaction for matching of the simulation model

basic situation

To be able to change the input parameters, at first a starting situation has to be introduced. The sorting system setup shown in Figure 2-12 is used for matching. The input parameters of the standard situation are as follows:

Fork Flow: the fork flow is set to 8000 forks/hour
n_fork: the number of forks per section of the sorting machine is 15. There are 13 forks at the end of the sorting machine. The sorting machine displayed in Figure 6-1 consist of 7 sections and 2 ends. This results in 7 * 15 + 2 * 13 = 131 forks
Rose dataset: The roses in the dataset are divided over the bunching stations as follows:

Destination	Number of roses
1	2000
2	3000
3	1000
4	2000
5	3000
6	1000
7	2000

Table 5-4 matching rose dataset

Filling ratio: 30%.
Time fork change 1: 2 seconds.
Time fork change 2: 2 seconds.
Time create bunch: 4 seconds.
Time move arm out: 4 seconds.
Time move arm back: 2 seconds
Error log: no error log
Time move bunches to end of buffer belt: short belts, so 13 seconds.

Fork flow

In this simulation, the fork flow is increased from 8000 forks/hour to 9000 forks/hour. The results and expected change direction is shown in Table 5-5. The results matches the expectation.

	Processing time	Number of roses rejected
expected	↓	↑
From	6.3	1063

To	6.0	2236

Table 5-5 matching fork flow result

N_forks

In this simulation the number of forks is changed from 131 forks to 160 forks. The results and expected

	Processing time	Number of roses rejected
expected	-	-
From	6.3	1063
To	6.3	1007

Table 5-6 matching N_forks result

change direction is shown in Table 5-6.The results matches the expectation.

Filling ratio

In this simulation the filling ratio is changed from 30% to 40%. The results and expected change direction

	Processing time	Number of roses rejected
expected	↓	↑
From	6.3	1063
To	5.9	4987

Table 5-7 matching filling ratio result

is shown in Table 5-6.The results matches the expectation.

Rose dataset

In this simulation the rose dataset is changed in a way that the roses are equally divided over the bunching stations. The new distribution is shown in Table 5-8,

Destination	Number of roses
1	2000
2	2000
3	2000
4	2000
5	2000
6	2000
7	2000

Table 5-8 changed matching rose dataset

	Processing time	Number of roses rejected
expected	↓	↓
From	6.3	1063
To	6.2	956

Table 5-9 matching rose dataset result

The results and expected change direction is shown in Table 5-9.The results matches the expectation.

bunching station action times

In this simulation, the bunching station action times are changed as follows:

Time fork change 1: from 2 seconds to 3 seconds.

Time fork change 2: from 2 seconds to 3 seconds.

Time create bunch: from 4 seconds to 6 seconds.

Time move arm out: from 4 seconds 6 seconds.

Time move arm back: from 2 seconds to 3 seconds

	Processing time	Number of roses rejected
expected	↑	↑
From	6.3	1063
To	6.5	1539

Table 5-10 matching bunching station result

The results and expected change direction is shown in Table 5-10.The results matches the expectation.

Time to move bunches to end of the belt

In this simulation, the belts are changed from short belts to long belts. The time to move bunches to end of the belt changes from 13 seconds to 21 seconds. The results and expected change direction is shown in Table 5-10. The processing time matches the expectation. The number of roses decrease a negligible

	Processing time	Number of roses rejected
expected	-	-
From	6.3	1063
To	6.3	1016

Table 5-11 matching time to move bunches to end of belt result

amount, which matches the expectation.

5.4 Validation of simulation model

In the validation part the model is validated against measurements. Validation is a complex process that involves the evaluation of the numerical accurateness and stability, both based on measurements (Grimmelius, 2005). The simulation model is validated in three steps. The steps are shortly described below and elaborate described per section.

1) **Entire system of Meewisse Roses b.v.:** In this step the simulation model of the entire sorting system of Meewisse Roses b.v. (section 5.1) is tested. In this step the roses cannot go round in the sorting machine. With this restriction, a deviation of a bunching station does not affect the rest of the system. In this way it is possible to locate the cause of a rejection. As rose dataset for this simulation, the exact IRISS dataset of a random day is used, including the empty forks and roses that pass the camera for a 2nd time. In this way the number of roses offered at each bunching station is forced to be similar to reality. The number of roses accepted and rejected may vary from reality and may be compared to reality. The output of this simulation sufficiently matches reality, except for the deviation in the bunching stations that are intensively used.
2) **Employee restrictions:** The simulation model is adapted. Assumptions are made to solve the above mentioned deviation. The deviation in the intensively used bunching stations is caused in the case of overflow at the buffer belt after the sealer. This overflow is not measured in reality and is caused by the number of employees and the working speed of the employees. An assumption of the employee restriction has been made to be able to add this human factors to the simulation model.
3) **Final model:** In the final simulation model the rose dataset is changed. From the IRISS dataset all empty forks and double roses are deleted. Also a variable filling ratio is added and the possibility for roses to circulate in the sorting machine is activated. In this simulation the time to sort all roses may vary and is also compared to reality.

Entire system of Meewisse Roses b.v.

In this step, the first validation of the entire sorting system of Meewisse Roses b.v. is executed. As rose dataset input for the simulation, the sheet with roses from reality is used. This spreadsheet(Figure 5-5) is created by IRISS. Each fork that pass the cameras has a rule in this sheet. In the columns all data of the roses is stored. A number of rows of such a sheet is shown in Figure 5-5.

ID is the identity of the measurement, ForkNo is the number of the fork, TimeStamp is the time when the fork passed the camera. The properties Color, Klasse, budWidth, budHeight, Maturity, NeckAngle, StemLength, Stemwidth, Stembend, Stemreliability and ManualCode are used to define the sorting. The StationNo is dependent on the sorting. The vision algorithms are dependent on the set cultivar. Flipper stands for the roses that could not be accepted. At the end of all bunching stations a photo sensor is located that measures if the rose is still in the fork. If flipper = 1 the rose is still in the fork and thus not accepted by any bunching station. These roses will probably make a second round. If the rose is accepted in the second round it will appear for a second time in the Spreadsheet with flipper = 0. A rose may be present multiple times in the spreadsheet. The last five columns are not used jet.

ID	ForkNo	TimeStamp	Color	Klasse	budWidth	budHeight	Maturity	NeckAngle	StemLength	StemWidth	StemBend50	StemBend60	StemBend70	StemBend80	StemBend90	StemReliability	StationNo	StationNo2	SendLength	Sorting	Flipper	ManualCode	Cultivar	BudTipDefect	StemBend30	AttachPlace	AttachWorkerI	MatID
625	158	09/08/2016 06:04:22.280	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Leeg	0	0	Red Naomi	0	NULL	-1	-1	-1
626	159	09/08/2016 06:04:22.640	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Leeg	0	0	Red Naomi	0	NULL	-1	-1	-1
627	160	09/08/2016 06:04:22.997	65	24	39	53	33	11	575	60	25	33	0	0	0	5	2	0	570	A1 60 34	1	0	Red Naomi	0	NULL	-1	-1	-1
628	161	09/08/2016 06:04:23.357	67	24	38	47	31	27	663	71	12	20	23	0	0	1	2	0	670	A1 70 33	0	0	Red Naomi	0	NULL	-1	-1	-1
629	162	09/08/2016 06:04:23.700	51	24	40	49	32	14	594	63	20	30	0	0	0	1	2	0	570	A1 60 34	0	0	Red Naomi	0	NULL	-1	-1	-1
630	163	09/08/2016 06:04:24.077	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Leeg	0	0	Red Naomi	0	NULL	-1	-1	-1
631	164	09/08/2016 06:04:24.417	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Leeg	0	0	Red Naomi	0	NULL	-1	-1	-1
632	165	09/08/2016 06:04:24.777	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Leeg	0	0	Red Naomi	0	NULL	-1	-1	-1
633	166	09/08/2016 06:04:25.150	47	80	39	56	30	6	743	77	9	23	37	37	0	5	9	0	765	A1 80 33	0	0	Red Naomi	0	NULL	-1	-1	-1
634	167	09/08/2016 06:04:25.493	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Leeg	0	0	Red Naomi	0	NULL	-1	-1	-1
635	168	09/08/2016 06:04:25.853	69	24	40	56	38	4	625	58	20	22	0	0	0	1	2	0	570	A1 60 34	0	0	Red Naomi	0	NULL	-1	-1	-1
636	169	09/08/2016 06:04:26.213	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Leeg	0	0	Red Naomi	0	NULL	-1	-1	-1
637	170	09/08/2016 06:04:26.570	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Leeg	0	0	Red Naomi	0	NULL	-1	-1	-1
638	171	09/08/2016 06:04:26.913	69	48	49	48	45	4	543	54	23	24	0	0	0	1	5	0	570	Rijp 60	0	0	Red Naomi	0	NULL	-1	-1	-1
639	172	09/08/2016 06:04:27.287	66	48	44	54	45	11	723	63	7	12	19	0	0	1	5	0	670	Rijp 70	0	0	Red Naomi	0	NULL	-1	-1	-1
640	173	09/08/2016 06:04:27.633	64	88	44	51	43	22	553	54	9	10	0	0	0	2	10	20	570	Cod A2 60 44	0	1	Red Naomi	0	NULL	-1	-1	-1
641	174	09/08/2016 06:04:27.990	49	80	38	50	28	2	746	70	3	8	17	17	0	2	9	0	765	A1 80 33	0	0	Red Naomi	0	NULL	-1	-1	-1
642	175	09/08/2016 06:04:28.350	56	88	43	50	38	11	589	63	8	11	0	0	0	2	10	18	570	Cod A2 60 33	0	1	Red Naomi	0	NULL	-1	-1	-1
643	176	09/08/2016 06:04:28.693	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Leeg	0	0	Red Naomi	0	NULL	-1	-1	-1
644	177	09/08/2016 06:04:29.067	52	80	41	51	30	11	751	78	11	13	21	21	0	1	9	0	765	A1 80 33	0	0	Red Naomi	0	NULL	-1	-1	-1

Figure 5-5 Rose sheet example

In the simulation only the StationNo is relevant.

In this simulation this sheet is used as a direct input for the discrete event simulation. In this simulation the Roses cannot make a 2nd round through the Sorting machine. This means the number of Roses offered per Bunching station will be equal to reality. The number of Roses accepted and rejected at each Bunching station may vary from reality. With this restriction, a deviation in a bunching station will not influence the rest of the sorting system and causes of deviations can be located. So the order of Roses that are offered are exactly equal to reality. However, the most important output, which is the time to sort all roses will not vary because of the restrictions.

The safest way to compare the simulation to reality is to take a look at the number of roses offered, accepted and rejected over time. This is done per bunching station. An example of such a graph is shown

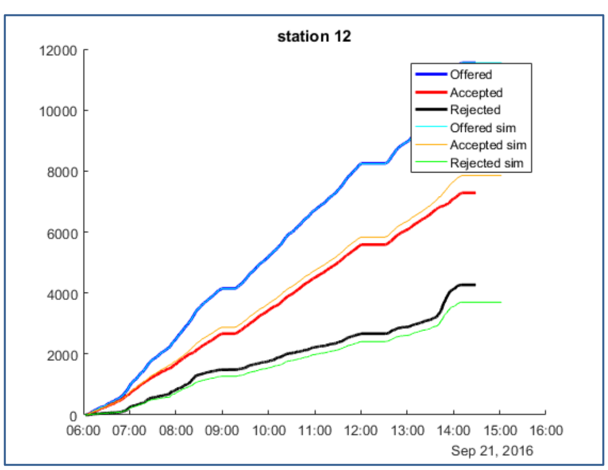


Figure 5-7a) offered accepted rejected, simulation vs. reality

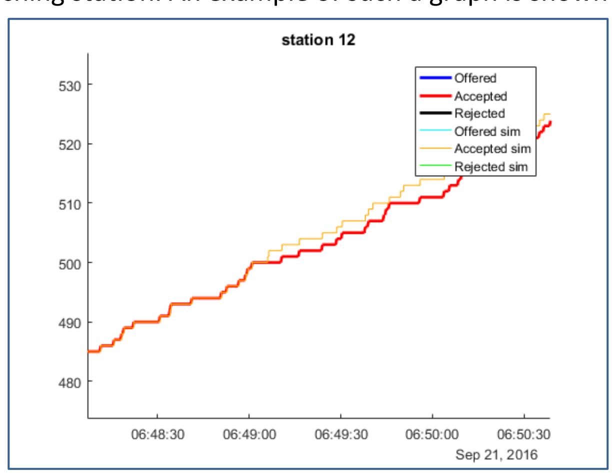


Figure 5-7b) offered accepted rejected, simulation vs. reality. Zoomed in on deflection point.

in Figure 5-7a and Figure 5-7b. It may be seen that the simulation matches reality very good in the beginning of the day, but differs from reality after a while.

Unfortunately this happens and cannot be avoided. Multiple reasons for these deviations are found:

- A rose falls out of the sorting machine after it is rejected by the bunching station. In this way it doesn't pass the return sensor so it cannot be seen in the datasheet that the rose is rejected.
- People walk by the bunching station and hold the mechanical arm for a few seconds to pass the bunching station.
- The buffer belt after the sealer is flooded, caused by people. This is not logged and therefore won't be present in the simulation.

As soon as there is a deviation from reality, the simulation will not match reality from that point on. This is shown in Figure 5-7. However, the simulation is still able to simulate the overall process. So the percentages of offering, accepting and rejecting roses may still match reality.

The first two reasons won't have a large effect on the number of roses accepted or rejected on a large scale. Therefore it is still possible to match the number of roses accepted and rejected with reality apart from minor deviations. The overview of the comparison between the simulation and reality is shown in Table 5-12. For each bunching station the number of roses offered, accepted and rejected is displayed. In the simulation the number of roses offered will be exactly equal to reality. The accuracy of the simulation can be read from the rule '*Difference Accepted [%]*'. It shows that most bunching stations match reality quite well, but some have a larger deviation.

	Station1	Station2	Station3	Station4	Station5	Station6	Station7	Station8	Station9	Station10	Station11	Station12	Station13	Station14	Station15	Station16	Station17	Station18	Station19	Station20
Real Offered	28	559	1173	8080	1338	12526	634	1248	2188	3806	11718	11548	655	1097	1020	771	186	437	979	302
Real Accepted	28	545	1125	6331	1268	7292	608	1194	2007	3200	7908	7283	625	1040	972	730	184	424	918	289
Real Rejected	0	14	48	1749	70	5234	26	54	180	606	3810	4265	30	57	48	41	2	13	61	13
Real accepted [%]	100.0	97.5	95.9	78.4	94.8	58.2	95.9	95.7	91.7	84.1	67.5	63.1	95.4	94.8	95.3	94.7	98.9	97.0	93.8	95.7
Simulation Offered	28	559	1173	8080	1338	12526	634	1248	2188	3806	11718	11548	655	1097	1020	771	186	437	979	302
Simulation Accepted	28	547	1122	7011	1288	8563	608	1193	2004	3303	8306	8003	625	1041	982	741	185	425	929	291
Simulation Rejected	0	12	51	1069	50	3963	26	55	184	503	3412	3545	30	56	38	30	1	12	50	11
simulation accepted [%]	100.0	97.9	95.7	86.8	96.3	68.4	95.9	95.6	91.6	86.8	70.9	69.3	95.4	94.9	96.3	96.1	99.5	97.3	94.9	96.4
Difference Offered	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Difference Accepted	0	2	-3	680	20	1271	0	-1	-3	103	398	720	0	1	10	11	1	1	11	2
Difference Rejected	0	-2	3	-680	-20	-1271	0	1	4	-103	-398	-720	0	-1	-10	-11	-1	-1	-11	-2
Difference Offered [%]	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Difference Accepted [%]	0.0	0.4	-0.3	10.7	1.6	17.4	0.0	-0.1	-0.2	3.2	5.0	9.9	0.0	0.1	1.0	1.5	0.5	0.2	1.2	0.7
Difference Rejected [%]	0.0	-14.3	6.3	-38.9	-28.6	-24.3	0.0	1.9	2.2	-17.0	-10.5	-16.9	0.0	-1.8	-20.8	-26.8	-50.0	-7.7	-18.0	-15.4

Table 5-12 Offered, Accepted, Rejected reality vs. simulation

In particular bunching station 4, 6, 10, 11 and 12 seem to deviate from reality. It is notable that the bunching stations where a deviation between reality and simulation occur seems to be the most intensively used stations. This is depicted in Figure 5-8. In this graph each bunching station is represented by a dot.

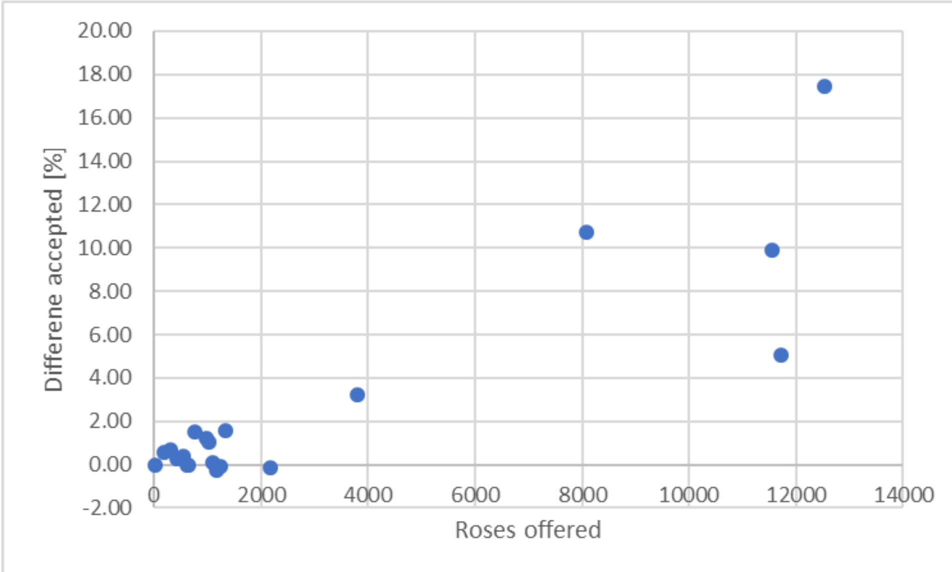


Figure 5-8 Difference accepted - number of roses offered. Each data point represents a bunching station

Add employee restriction to simulation model

Probably the deviation is caused by the third abovementioned reason. When the buffer belt after the sealer is flooded, roses may be accepted by each bunching station until the corresponding buffer belt of the bunching station is full. In most cases the most intensively used bunching stations and buffer belts will be full at first.

Unfortunately flooding of the buffer belt after the sealer is not logged in reality and cannot be included directly in the simulation. However, the flooding of the buffer belt may be simulated. Two parameters are needed to do that. These parameters are the number of bunches that may be located on the buffer belt and the time it takes to remove the bunches from the buffer belt.

25 bunches may be located on the buffer belt after the sealer and 5 bunches are removed every 50 seconds. Nevertheless, when these values are used in the simulation, it has no effect because the belt is never flooded in the simulation. A reason for flooding of the buffer belt may be the variable working speed of the employees. It is obvious that employees work harder when the belt is almost flooded and work slower when there are only a few bunches on the belt. However this behavior is very difficult to quantify and probably even differs per person. Besides, there is a big difference in company policies, at some greenhouses the buffer belts are always almost full and at other greenhouses the buffer belts are always almost empty. More ways to simulate this kind of behavior may be thought of.

The approach chosen for this simulation is to lower the capacity of the belt and increase the time to remove 5 bunches from the belt. Multiple simulations are run and the presumption that only the intensively used bunching station will be affected is confirmed. The result is shown in Table 5-13 and is accomplished using the following buffer belt variables:

Sealer 1	
Capacity (bunches)	Time to remove five bunches (s)
10	78

Sealer 2	
Capacity (bunches)	Time to remove five bunches (s)
10	54

	Station 1	Station 2	Station 3	Station 4	Station 5	Station 6	Station 7	Station 8	Station 9	Station 10	Station 11	Station 12	Station 13	Station 14	Station 15	Station 16	Station 17	Station 18	Station 19	Station 20
Real Offered	28	559	1173	8080	1338	12526	634	1248	2188	3806	11718	11548	655	1097	1020	771	186	437	979	302
Real Accepted	28	545	1125	6331	1268	7292	608	1194	2007	3200	7908	7283	625	1040	972	730	184	424	918	289
Real Rejected	0	14	48	1749	70	5234	26	54	180	606	3810	4265	30	57	48	41	2	13	61	13
Real accepted [%]	100	97.5	95.91	78.35	94.77	58.21	95.9	95.67	91.73	84.08	67.49	63.07	95.42	94.8	95.29	94.68	98.92	97.03	93.77	95.7
Simulation Offered	28	559	1173	8080	1338	12526	634	1248	2188	3806	11718	11548	655	1097	1020	771	186	437	979	302
Simulation Accepted	28	547	1122	6291	1288	7263	608	1193	2004	3239	7696	7143	625	1041	982	741	185	425	929	291
Simulation Rejected	0	12	51	1789	50	5263	26	55	184	567	4022	4405	30	56	38	30	1	12	50	11
simulation accepted [%]	100	97.85	95.65	77.86	96.26	57.98	95.9	95.59	91.59	85.1	65.68	61.85	95.42	94.9	96.27	96.11	99.46	97.25	94.89	96.36
Difference Offered	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Difference Accepted	0	2	-3	-40	20	-29	0	-1	-3	39	-212	-140	0	1	10	11	1	1	11	2
Difference Rejected	0	-2	3	40	-20	29	0	1	4	-39	212	140	0	-1	-10	-11	-1	-1	-11	-2
Difference Offered [%]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Difference Accepted [%]	0	0.37	-0.27	-0.63	1.58	-0.4	0	-0.08	-0.15	1.22	-2.68	-1.92	0	0.1	1.03	1.51	0.54	0.24	1.2	0.69
Difference Rejected [%]		-14.3	6.25	2.29	-28.6	0.55	0	1.85	2.22	-6.44	5.56	3.28	0	-1.75	-20.8	-26.8	-50	-7.69	-18	-15.4

Table 5-13 Offered, Accepted, Rejected reality vs. simulation, with buffer belt after sealer limitations

The result shown above is acceptable to go to the next validation step.

final simulation model

This model is used in the rest of the research. In this simulation the spreadsheet with the roses is reduced by deleting all empty forks and the double roses. In this way a sheet with one rule for each rose remains. In this simulation the ability for the roses that are not accepted to make a 2nd round is activated. To match reality, a similar filling ratio and filling sequence as in reality should be realized. A vector with filling ratio, for each row of the original sheet is calculated.

In this simulation the number of roses accepted per bunching station will be identical to reality, however the number of roses rejected and offered may differ from reality. The number of roses rejected will in this case also affect the total time to process all roses. The more roses are rejected, the more time it takes to process all roses. The difference offered and difference rejected will now form the parameters to compare the simulation to reality. Besides the number of roses offered accepted and rejected, now the most important parameter may be compared to reality, which is the time to process all roses of a day.

To calculate the filling ratio the entire sheet, with empty forks and double roses, equation 1 (page 7) is used as reference. The equation is translated from a calculation from reality to a calculation on the datasheet:

$$R_f(i) = \frac{N_{roses}(i)}{i} \qquad \text{(Equation 5)}$$

- *i* = Row number (number of forks passed)
- *Rf (i)* = Filling ratio at row number *i*
- *Nr(i)* = Number of roses until row number *i*

In the simulation the filling ratio is also calculated. If the filling ratio in the simulation after fork *i* is lower than reality, roses are generated. If the filling ratio in the simulation is lower than in reality no roses are generated, so empty forks will go through the sorting machine.

In Figure 5-9, a comparison between reality and the simulation for one bunching station is shown. It is shown that the simulation doesn't follow reality as well as the previous simulation. This has multiple reasons:

- In reality you may see clear breaks that are not simulated
- In this simulation a deviation has effect on the sequence of the roses and empty forks. In other words a deviation in one bunching station affects all other bunching stations.

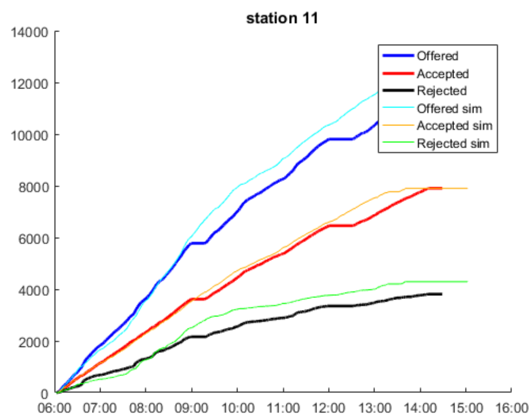


Figure 5-9 comparison reality-simulation, bunching station 11

Although the lines cannot be compared as easy as in the previous validation step, the total number of roses offered, accepted and rejected per bunching station should still match reality. An overview is shown in Table 5-14.

To compare the simulation to reality the difference in number of roses offered is used. This difference varies over the bunching station from 0% to 10.7%. The average difference is 2.45%.

	Station 1	Station 2	Station 3	Station 4	Station 5	Station 6	Station 7	Station 8	Station 9	Station 10	Station 11	Station 12	Station 13	Station 14	Station 15	Station 16	Station 17	Station 18	Station 19	Station 20
Real Offered	28	559	1173	8080	1338	12526	634	1248	2188	3806	11718	11548	655	1097	1020	771	186	437	979	302
Real Accepted	28	545	1125	6331	1268	7292	608	1194	2007	3200	7908	7283	625	1040	972	730	184	424	918	289
Real Rejected	0	14	48	1749	70	5234	26	54	180	606	3810	4265	30	57	48	41	2	13	61	13
Real accepted [%]	100.0	97.5	95.9	78.4	94.8	58.2	95.9	95.7	91.7	84.1	67.5	63.1	95.4	94.8	95.3	94.7	98.9	97.0	93.8	95.7
Simulation Offered	28	559	1162	8944	1293	13337	618	1213	2177	3704	12195	11487	633	1126	1008	765	186	430	943	298
Simulation Accepted	28	545	1127	6339	1271	7292	614	1197	2013	3196	7902	7250	625	1040	973	734	185	426	919	291
Simulation Rejected	0	14	35	2605	22	6045	4	16	164	508	4293	4237	8	86	35	31	1	4	24	7
simulation accepted [%]	100.0	97.5	97.0	70.9	98.3	54.7	99.4	98.7	92.5	86.3	64.8	63.1	98.7	92.4	96.5	95.9	99.5	99.1	97.5	97.7
Difference Offered	0	0	-11	864	-45	811	-16	-35	-11	-102	477	-61	-22	29	-12	-6	0	-7	-36	-4
Difference Accepted	0	0	2	8	3	0	6	3	6	-4	-6	-33	0	0	1	4	1	2	1	2
Difference Rejected	0	0	-13	856	-48	811	-22	-38	-16	-98	483	-28	-22	29	-13	-10	-1	-9	-37	-6
Difference Offered [%]	0.0	0.0	-0.9	10.7	-3.4	6.5	-2.5	-2.8	-0.5	-2.7	4.1	-0.5	-3.4	2.6	-1.2	-0.8	0.0	-1.6	-3.7	-1.3
Difference Accepted [%]	0.0	0.0	0.2	0.1	0.2	0.0	1.0	0.3	0.3	-0.1	-0.1	-0.5	0.0	0.0	0.1	0.6	0.5	0.5	0.1	0.7
Difference Rejected [%]		0.0	-27.1	48.9	-68.6	15.5	-84.6	-70.4	-8.9	-16.2	12.7	-0.7	-73.3	50.9	-27.1	-24.4	-50.0	-69.2	-60.7	-46.2

Table 5-14 comparison reality-simulation overview

The more important and overall parameter is the time to process all roses of the day. In reality this time is seen as the time from the first rose in the sheet until the last rose in the sheet minus the two breaks, that is 7.6 hours. In the simulation it takes 7.8 hours. The difference is 2.9% or 13 minutes.

5.5 Sensitivity analysis

The objective for the sensitivity analysis is to investigate the effect of the uncertainty of the input parameter on the key performance indicators (Macdonald & Strachan, 2001; Van, Bontsema, & Henten, 2013), which are the processing time, the number of roses offered, accepted and rejected and the number of roses rejected per reason. In this research the uncertain input parameter is the generation order of the rose entities. To determine the sensitivity of the simulation, the simulation on the same situation is executed 10 times. The only difference is in the generation order of the rose entities. For each simulation the same set of roses is used, but the order is randomized.

The main result, which is the processing time of all the roses varies from 7.28 hours to 7.35 hours, with a mean of 7.32 hours. The maximum difference is 0.07 hours (4.2 minutes), which corresponds to 1% of the total time.

The variation of the number of roses offered, accepted and rejected is shown in Figure 5-10.

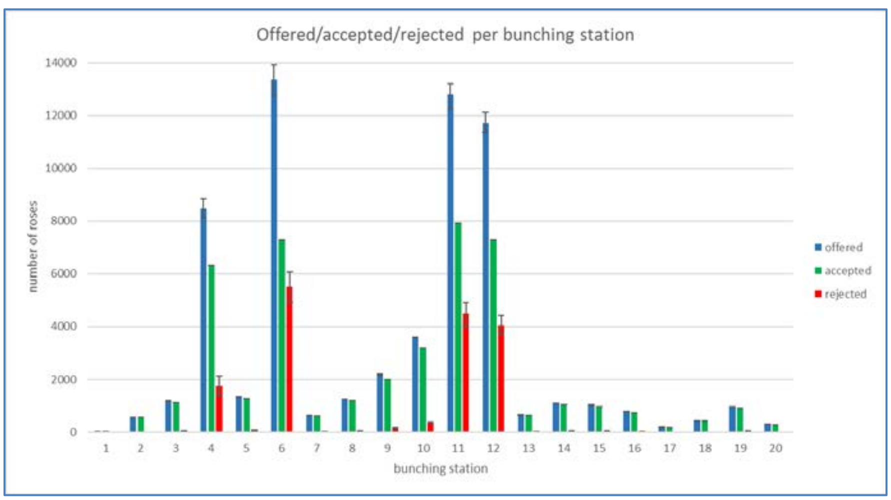


Figure 5-10 number of roses offered accepted and rejected per bunching station with error bars

The bar graph shows the bunching stations on the horizontal axis and the number of roses on the vertical axis. For each bunching station the number of roses offered (blue), the number of roses accepted (green) and the number of roses rejected (red) is displayed. The differences between the multiple simulations are shown with error bars. The number of roses offered is the sum of the number of roses accepted and rejected. It can be seen that the variation of the number of roses offered is fully caused by the variation in the number of roses rejected. The number of roses accepted is equal in each simulation, because that corresponds to the number of roses present in the rose dataset. The variation of the number of roses offered is at maximum 5% at bunching station 6. On average the variation of offered roses at a bunching station is 1% of the total number of roses offered at that bunching station.

The rejected roses are divided in five categories of rejection and shown in Figure 5-11.

- Change fork (dark blue): roses rejected during the fork change within the bunching station
- Create bunch (orange): roses rejected caused by bunch creation (pushing roses together and wrap a string around the stems)
- Move arm (grey): roses rejected caused by moving the arm of the bunching station
- Belt full (pink): roses rejected caused by belt the buffer belt that is waiting until it may transfer bunches to the V-belt
- Error (light blue): roses rejected because the bunching station is in an error state.

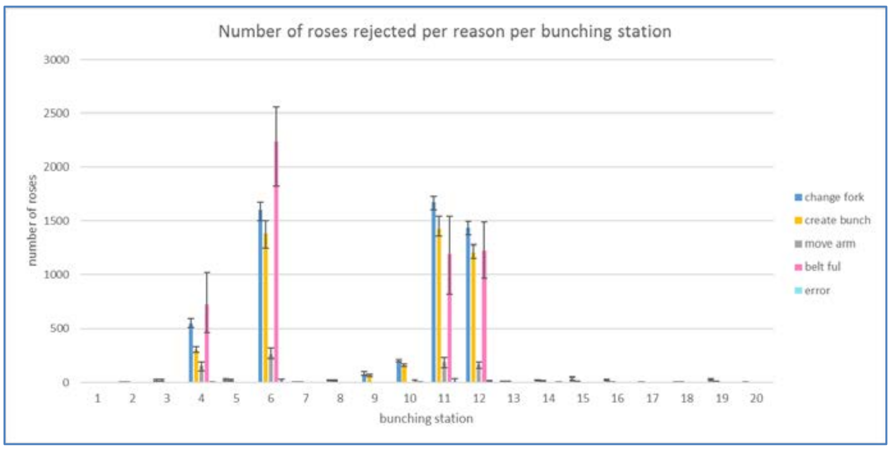


Figure 5-11 number of roses rejected per reason per bunching station with error bars

The sum of these bars per bunching station is similar the total number of roses rejected displayed in Figure 5-10. Notable in this graph is that the biggest part of the difference between the simulation (displayed by error bars) is caused by the rejection reason belt full. This is caused by the coincidence of the event and the large effects of the incident. “Belt full” will mainly occur when multiple buffer belts are full at the same time. If a buffer belt is full, the bunching station will start rejecting roses when it is also entirely full. This may last for a few minutes. The distribution of the variation over the different causes is as follows. Change fork 16%, create bunch 17%, move arm 8%, belt full 58% , error 2%.

The most important output parameter is the processing time, that showed a deviation of 1% of the total time. Compared to the changes in processing time, which is a result of chapters 6 and 7 this is a very small deviation. Therefore the result of the sensitivity analysis is satisfactory.

6 Effect of reducing each bottleneck in setup of Porta Nova III

In this chapter the effect of the maximum theoretical reduction of each bottleneck(defined in chapter 3) is investigated. At first the basic situation for Porta Nova III is defined. The simulation model and input, used for verification matching and validation, has to be adapted to match the setup of Porta Nova III (section 6.1).

To investigate the effect of reducing the bottlenecks, the simulation model and input used for the basic situation have to be modified (Section 6.2). These modifications are discussed per bottleneck.

In section 6.3 the result of the aforementioned situations is discussed. The result is expressed in total time to process the roses and in number of roses offered accepted and rejected (per reason).

All experiments are listed in Table 6-1.

No.	Situation	Section
1	Basic situation PNIII	6.1
2	100 % filling ratio	6.2
3	Increase fork flow	6.2
4	Remove bunching station bottleneck	6.2
5	Remove buffer belts	6.2
6	Two attachment locations	6.2
7	All feasible improvements	7.1

Table 6-1 list of experiments

The last situation, where all feasible improvements are combined is discussed in chapter 7. Each situation is simulated 10 times with a random order of the roses.

6.1 Standard setup of Porta Nova III

As discussed in the introduction, Porta Nova desires a system with 36 bunching stations. For this setup the sorting machine has to be extended. Also it has to be decided how the buffer belts V-belts and sealers are composed.

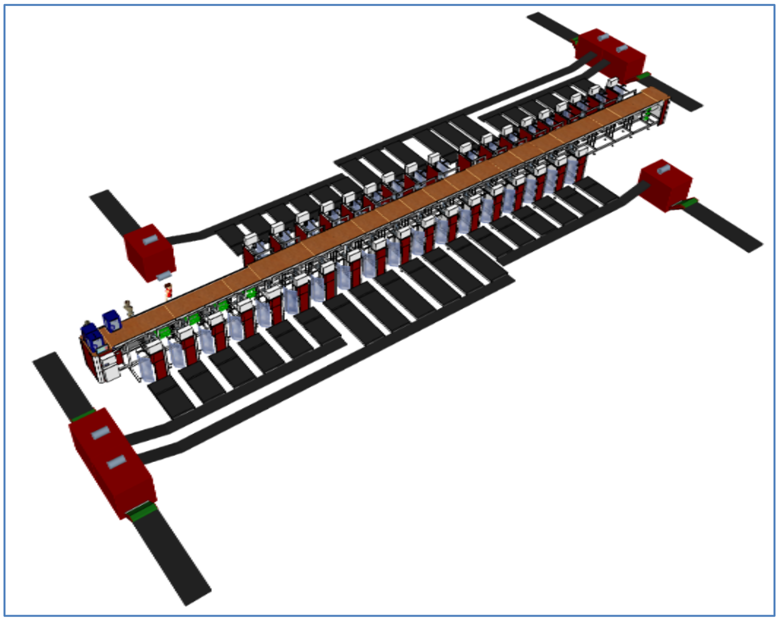


Figure 6-1 Basic setup of Porta Nova 3

Normally in large rose cultivation greenhouses, there is a maximum 7 buffer belts connected to one V-belt and sealer. Because it is desired to process more roses in this greenhouse Porta Nova decided to connect only 6 buffer belts to one V-belt. This results in 6 sets of 6 bunching stations and buffer belts. Each set is connected to a V-belt and sealer. This setup is shown in Figure 6-1. 18 bunching station are connected to each side of the sorting machine.

Model adaptations

To adapt the simulation model to this setup the composition of subsystems has to be changed. Compared to the situation shown in Figure 4-1, one more Pick bunchstation, Bunching station and Buffer belt subsystem has to be created and connected to the V-belt. Then the whole set of the aforementioned subsystems has to be copied 5 more times.

Model input

Fork Flow: the fork flow is set to 9500 forks/hour (conform current sorting systems of Porta Nova)

n_fork: the number of forks per section of the sorting machine is 15. There are 13 forks at the end of the sorting machine. The sorting machine displayed in Figure 6-1 consist of 26 sections and 2 ends and results in

26 * 15 + 2 * 13 = 416 forks

Rose dataset: For the rose dataset the datasets of Porta Nova II and Porta Nova I are used. The day where the number of roses processed is the closest to 90.000 in December 2016 is used. In this day 90.192 roses are processed. However this dataset cannot be used directly, because the number of bunching stations is different in the new situation. Porta Nova does already have 36 sortings in the current situation, but sends multiple sortings to one bunching station right now. In the new situation, there will be a bunching station for each sorting. The StationNo is changed to one station for each sorting. The new distribution of the roses over the bunching stations is shown in Table 6-2.

Destination	Number of roses	Destination	Number of roses	Destination	Number of roses
1	786	13	2358	25	175
2	173	14	167	26	6971
3	1050	15	4078	27	1869
4	3420	16	7402	28	77
5	4928	17	4143	29	1566
6	4208	18	1070	30	3010
7	210	19	11096	31	3146
8	3471	20	422	32	696
9	180	21	150	33	2013
10	1103	22	6720	34	1332
11	86	23	1350	35	1805
12	7184	24	1557	36	220

Table 6-2 Rose destination distribution

Filling ratio: The filling ratio is calculated from the used rose dataset. The overall filling ration is 84%.

Time fork change 1: The time for the first fork change may be different for each bunching station. However, this time is set to the most common situation of Porta Nova II (8 out of 14 bunching stations) for all bunching stations. This is 4 forks that corresponds to 1.52 seconds. This is calculated with the following formula.

$$t = Nf * \frac{3600}{ff} \qquad \text{(Equation 6)}$$

- t = time
- Nf = number of forks
- ff = fork flow (forks/hour)

Time fork change 2: For the same reason as “Time fork change 1”, this setting is for all bunching stations set to 1.89 seconds (equation 6).

Time create bunch: For the same reason as “Time fork change 1”, this setting is for all bunching stations set to 4.93 seconds (equation 6).

Time move arm out: For the same reason as “Time fork change 1”, this setting is for all bunching stations set to 3.79 seconds (equation 6).

Time move arm back: For the same reason as “Time fork change 1”, this setting is for all bunching stations set to 1.89 seconds (equation 6).

Error log: The error log has a negligible contribution to the number of roses rejected and is therefore left out.

Time move bunches to end of buffer belt: For the long buffer belts (7-12 and 25-30) this time is set to 21 seconds and for the short buffer belts (1-7, 13-24 and 31-36) this time is set to 13 seconds.

6.2 Simulation model per bottleneck

In this section it is discussed how a bottleneck may be reduced. Afterwards it is discussed how to achieve this situation in a simulation. This is discussed per bottleneck. For each situation the input variables will be similar to the basic situation except for the variables that are discussed.

filling ratio

The filling ratio is determined from the rose dataset and is on average 80.0%. To reduce this bottleneck, the filling ratio is set to 100%. In this situation, the filling ratio is constant over the entire simulation.

fork flow

The fork flow limits the number of roses processed per hour and is therefore a bottleneck. However the fork flow is not a bottleneck that may be removed, theoretical the fork flow may be infinitely high. During the research it was discovered that the maximum fork flow is right now 12.000 forks/hour. This number is used for this simulation.

bunching station busy

When the bunching station is busy roses are rejected and make another round in the sorting machine. In the simulation this bottleneck may be removed entirely. This is achieved by setting the time variables for the actions in the bunching stations all to zero. This concerns the parameters “Time fork change 1”, “Time fork change 2”, “Time create bunch”, “Time move arm out”, “Time move arm back”.

belt system flooding

To remove the belt system flooding, the entire belt system (page 21) is removed. This is also the case in reality for some greenhouses. In that case the bunches are dropped in a net and the bunching station will always immediately drop the bunch when the arm is moved out. To mimic this situation in the simulation model, the buffer belt, V-belt and Sealer subsystems are removed. The “entity servers” and “entity terminators” that are in the Sealer (page 16) are connected directly to the outlet of the Bunching stations.

number of attachment locations

Theoretical there may be an attachment location between every bunching station. However this high number of attachment locations would result in very radical changes in the size of the sorting machine and the logistics of supply of roses around the sorting system. The camera system is located at the end of the machine and can be only mounted on the end of the sorting machine right now. Therefore, the number of attachment locations is set to 2, placed at both ends. In the simulation model, this situation is created by copying the Attach and qualify roses subsystem, Flipper subsystem and Backway subsystem and paste the subsystems between the Pick bunchstation subsystem 18 and Pick bunchstation subsytem 19. One rose dataset is used and the “entity generators” in both Attach and qualify rose subsytems will pick the first unused rule from that rose dataset. For the “entity generators”, that creates Fork entities it means that the generators will both create half of the total number of Forks. Also the constant in the Backway subsystem has to be changed from n_forks to n_forks / 2

6.3 Results

Each of the aforementioned situations is simulated 10 times. All graphs shown contain the average of 10 simulations. The minima and maxima are displayed with error bars. Detailed result per situation are shown in Appendix C. An overview, that consists of three graphs is shown in this section:

- Time to process all Roses per situation (Figure 6-2).
- Number of Roses offered, accepted and rejected per situation (Figure 6-3).
- Number of Roses per reason of rejection, per situation (Figure 6-4).

Processing time

The time to process all Roses is depicted in Figure 6-2. In the basic situation, the processing time is 13.3 hours. Most time may be saved by adding a second attachment location. With this adaption the time to

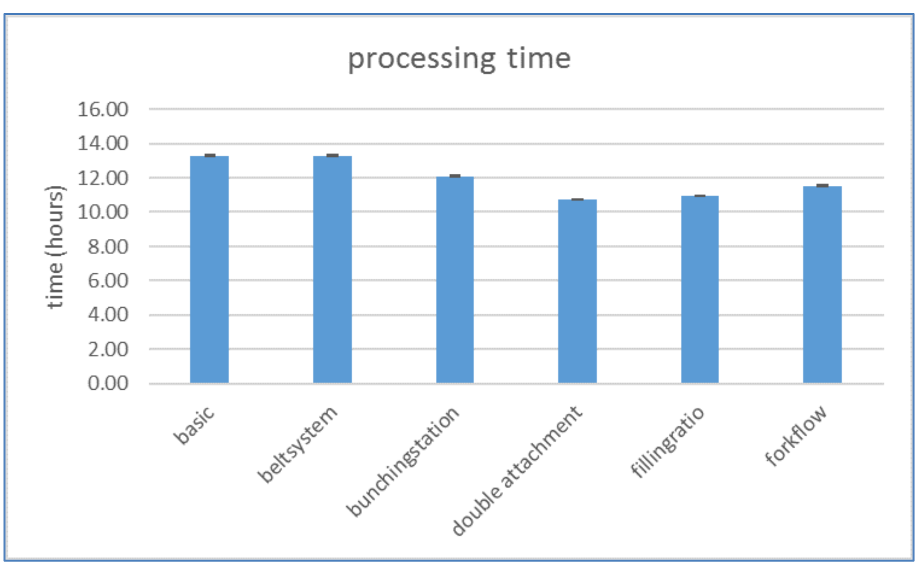


Figure 6-2 time to process all roses per situation process all roses is 10.7 hours. This is a time saving of 20%. The time to process all roses without the belt system is equal to the basic situation. The rest, bunching station busy (12.2 hours, time saving of 9%), filling ratio (10.9 hours, time saving of 18%) and fork flow (11.5 hour, time saving of 13%) are in between the basis situation and an extra attachment location.

Number of roses offered, accepted, rejected

The number of roses offered accepted and rejected per situation is shown in Figure 6-3.

Removing the belt system should result in a reduction of the number of rejected roses. However in Figure 6-3 It can be seen that the number of rejected roses of the situation without the belt system is equal to the number of rejected roses in the basic situation. In the situations with an extra attachment location or a higher filling ratio, the number of roses rejected increased. With a higher fork flow, the number of roses rejected stays equal.

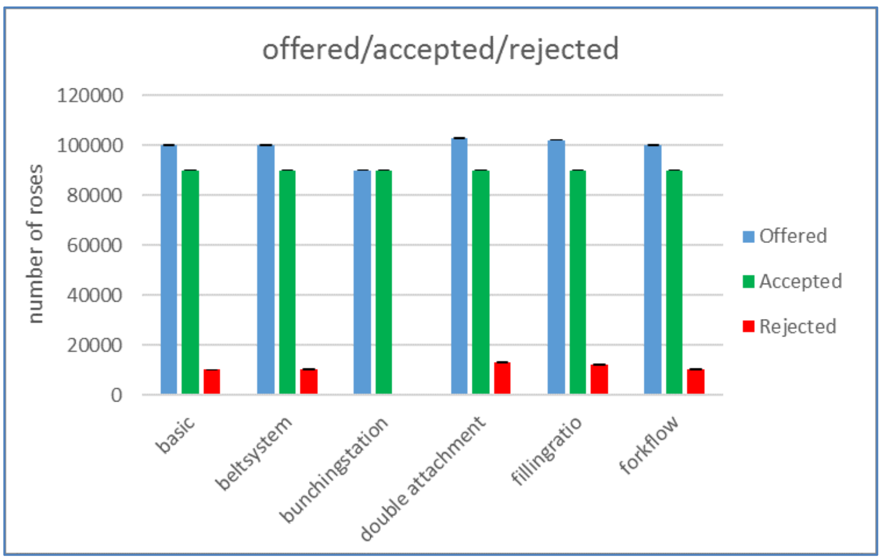


Figure 6-3 number of roses offered accepted and rejected per situation

number of roses rejected per reason.

In Figure 6-4 the number of roses per reason of rejection is shown. This shows that there were almost no roses rejected caused by the belt system in the basic situation, which explains that the number of rejected roses does not decrease when the belt system is removed. With two attachment location the number of roses rejected caused by the belt system is just above 0.

The number of roses rejected caused by move arm is also almost 0 in the basis situation. In the other situations it hardly differs.

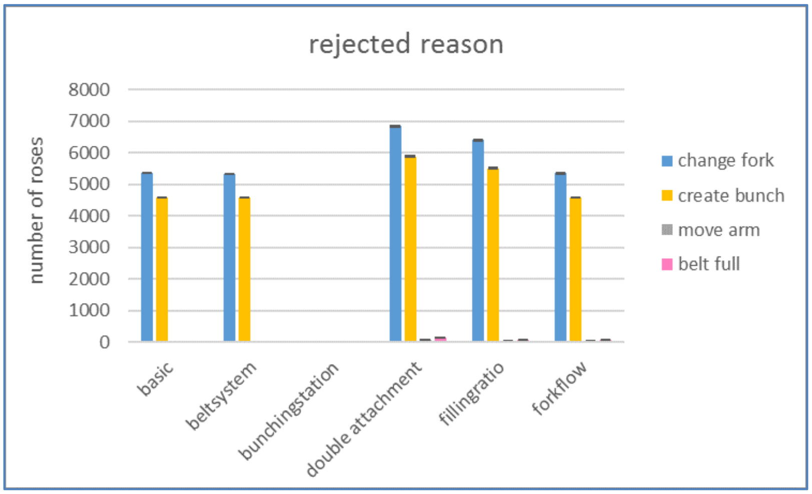


Figure 6-4 number of roses rejected per reason per situation.

6.4 Discussion

It is notably that no roses are rejected in the basic situation caused by the belt system. The number of roses rejected caused by the belt systems is mainly dependent on the number of bunches, that have to be processed by each V-belt and sealer. The fact that no roses are rejected caused by belt system flooding, shows that the number of bunches, that have to be processed by each V-belt and sealer is low enough when 6 V-belts and 6 sealers are used to ensure no roses will be rejected caused by belt system flooding. However when a second attachment location is created, there are roses rejected because of belt system flooding. Probably the number of rejected roses caused by belt system flooding will get larger when multiple bottleneck reductions are combined, which is done in chapter 7.

Besides it can be seen that the number of roses rejected remain equal when the fork flow increases, but increase when the filling ratio is increased or when there are two attachment locations. When the fork flow increases the time to process the roses decreases because the time between the presence of each fork is shorter. However, the number of roses rejected during a fork change or bunch creation stays similar. If a second attachment location is created or the filling ratio is improved, the time to process the roses will be shorter because a larger percentage of the forks will contain a rose.

6.5 Conclusion

With these simulations multiple conclusions are drawn

- In the basic situation, it takes 13.3 hours to process all roses.
- In the basic situation, hardly any roses will be rejected because of the belt system.
- In the basic situation, hardly any roses will be rejected because of the arm movement.
- Introducing an extra attachment location will have the largest time saving.
- A higher filling ratio, a higher fork flow or decreasing the time for the bunching station actions will also save time.

7 Effects of feasible adjustments in setup of Porta Nova III

Some of the bottleneck improvement mentioned in chapter 6 are feasible in reality right now, others are not completely feasible in reality. In this chapter it is discussed whether the bottleneck improvements are feasible right now and how it is determined. This is followed by the result of a simulation with all feasible bottleneck improvements.

7.1 Feasible adjustments

Filling ratio

In the previous chapter, the filling ratio was set to 100%. In reality a filling ratio of 100% is only possible when, the attachment of roses is automated. However at this moment there are no machines are available to automate the attachment. This does not mean that the filling ratio cannot be higher than in the basis situation. To determine the maximum feasible filling ratio the data of multiple greenhouses is studied. The maximum filling ratio is determined and considered as the maximum feasible filling ratio. This results in an overall filling ratio change from 84% to 87%.

Fork flow

The maximum fork flow at the moment is 12.000 forks/hour.

Bunching station busy

The number of roses rejected due to bunching station actions is set to 0 in the previous chapter. This is not feasible in reality, however the time for the actions in the bunching station may be decreased compared to the basic situation. The settings of the bunching station and the pressure for each pneumatic activator may be adapted. This is tested in reality and described in appendix D. The lowest achievable times are as follows:

- Change fork: 1 fork or 0.3 seconds (equation 6)
- Create bunch: 5 forks or 1.5 seconds (equation 6)
- Move arm out: 10 forks or 3.0 seconds (equation 6)
- Move arm in: 14 forks or 4.2 seconds (equation 6)

Belt system flooding

In the previous chapter the belt system flooding bottleneck is removed by entirely removing the buffer belts, V-belt and sealer. This is also feasible in reality. In some greenhouses it even is the current situation.

Number of attachment locations

Introducing another attachment location is feasible on short term. The hardware already exist, it only has to be placed in another assembly. Also the IRISS software has to be updated.

7.2 Result

The result of the simulation with all above discussed adjustments is depicted in the figures below. A detailed result is shown in Appendix C.

Processing time

In Figure 7-1 the time to process all roses in the dataset is shown. The basic situation, the situations discussed in chapter 6 and the situation with the feasible improvements are shown. The processing time is decreased from 13.3 hours in the basic situation to 8.6 hours in the situation with feasible improvements. The processing time is reduced with 35%. The overall capacity of the sorting system is now 10465 roses/hour.

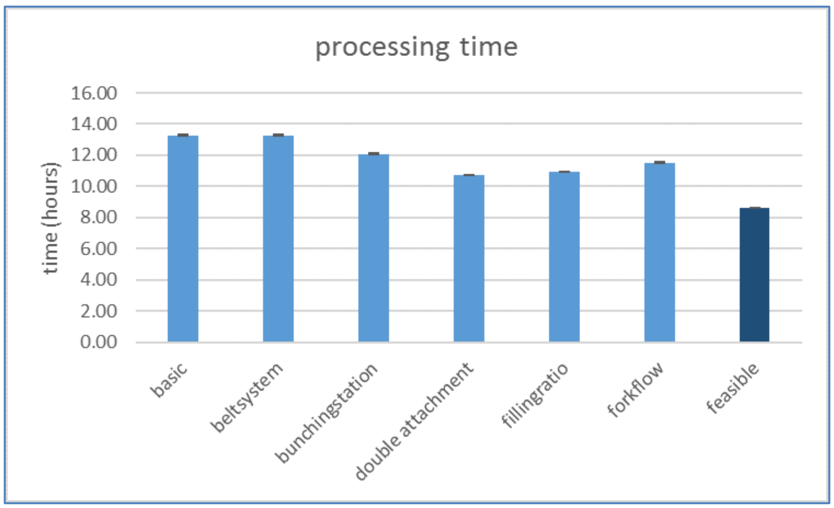


Figure 7-1 time to process all roses per situation

Number of roses offered, accepted, rejected

In Figure 7-2 the number of roses offered, accepted and rejected is shown. The basic situation, the situations discussed in chapter 6 and the situation with the feasible improvements are shown. The number of roses rejected is decrease from 9965 roses to 6457 roses (6457 roses rejected takes 0.6 hours to process). This is a reduction of 35%. The number of roses accepted obviously stay equal. The number of roses offered is the sum of both the rejected and accepted roses.

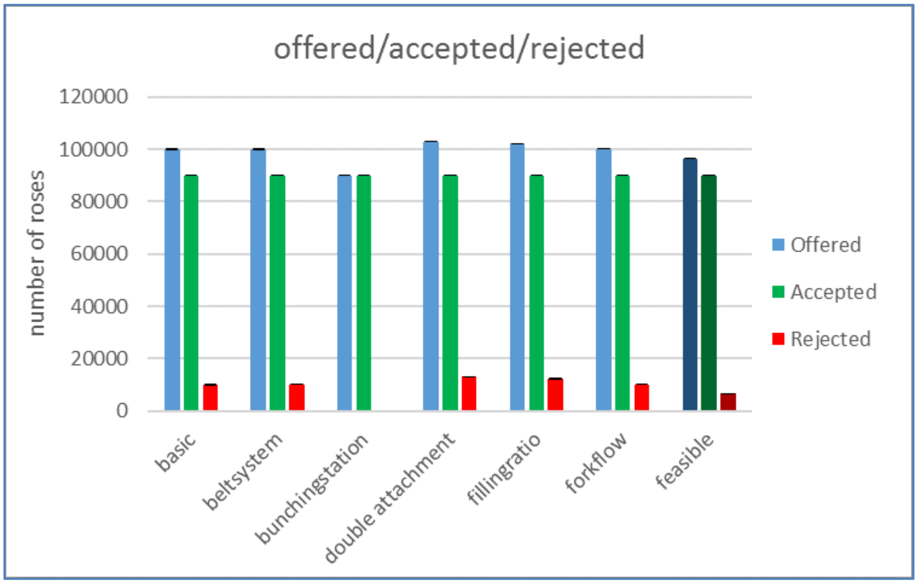


Figure 7-2 number of roses offered accepted and rejected per situation

Number of roses rejected per reason

In Figure 7-3 the number of roses rejected per reason is shown. The number of roses rejected caused by move arm and belt full are similar to the other situations. The number of roses rejected cause by fork change and create bunch however decreased significantly compared to the basic situation.

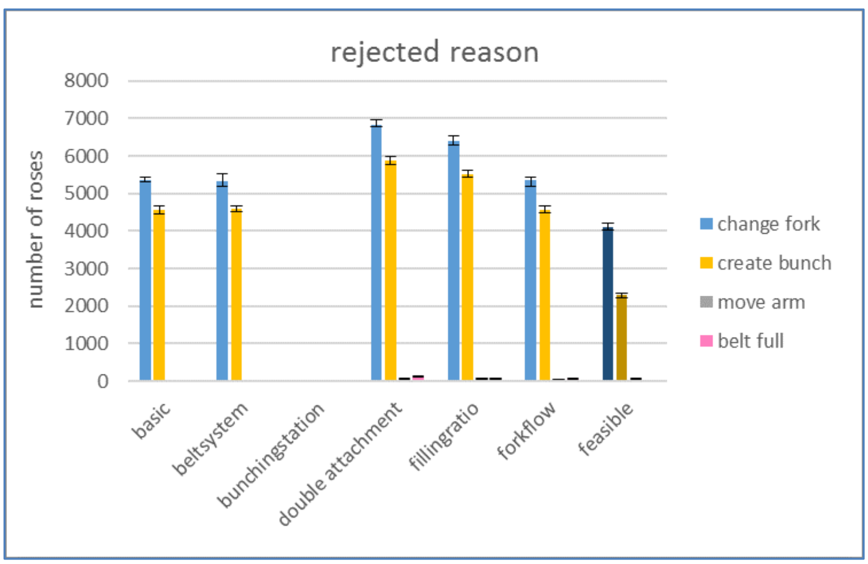


Figure 7-3 number of roses rejected per reason per situation

7.3 Discussion

All improvements are implemented together in this last simulation. The result is satisfying, however it is not exactly clear which improvement cares for which part of the time reduction.

The number of roses rejected should increase because of the double attachment and the filling ratio, but decreases because of the improved bunching station. Apparently the decrease is more than the increase.

In this last simulation all improvements are applied. Among which is the removing of the belt system. This is a quite radical change to the system, however it is not clear how much this adjustment contributes to the time savings.

With the improvements “double attaching”, “filling ratio” and “forkflow” there are a few roses rejected because of the belt system flooding. Therefore it is expected that if those three improvements are applied, the number of roses rejected caused by belt system flooding would be significantly higher. However this expectation is just a hypothesis and is not proven.

7.4 Conclusion

With these simulations multiple conclusions are drawn.

- With the feasible improvements, stated in section 7.1, the processing time is reduced from 13.3 hours to 8.6 hours.
- The desired 8 hours is not achieved.
- The maximum extra time, that may be saved with improving the bunching station is 0.6 hours (no retour roses).
- There is still improvement possible through improving the fork flow and/or the filling ratio.

8 Conclusion

In the case of Porta Nova III four out of five bottleneck improvements cause a decrease in processing time. The improvements are; adding an extra attachment location, increase fork flow, increase filling ratio and speed up bunching station actions. Using these improvements, the time to process 90192 roses is decrease from 13.3 hours to 8.6 hours. Despite the processing time is decreased significantly, the desired 8 hours is not reached. However further improvement may be possible (section 8.1).

As a result of the research multiple conclusions are drawn. These conclusions are stated below. Further there are some recommendations for further research. This is discussed in section 8.1.

The goal for the research was to have an understanding of the rose sorting process, to develop a method to predict the sorting capacity using different machine adaptations and to provide an advice for machine adaptations to reach the desired sorting capacity.

The sorting process is elaborated per standard machine in process flow charts. The dependencies within the machines are identified and displayed. Using this information, the 5 bottlenecks are identified. A simulation model is developed using discrete event simulation. This discrete event simulation model accurately mimics the rose sorting process. Supplementary, the simulation model is easily adjustable in different combinations of machine and input variables. The discrete event simulations provides more insight in the dependencies of the rose sorting process. Using the simulation model, it is possible to quantify the number of roses rejected per cause.

The simulation model is developed for the case of Porta Nova III. However it the model is developed in such a way that it is easily adaptable and may be used for the rose sorting system setup of all rose cultivation companies. The simulation model provides insight in the effect of reducing bottlenecks. The effect is expressed in processing time and the number of roses offered, accepted and rejected per reason.

8.1 Recommendations

The simulation model is validated with data of one day at one location. It is recommended to validate the model on multiple days and multiple locations.

For further decrease of the processing time there are multiple options left: increase the filling ratio, increase the fork flow, and increase the bunching station accepting ratio.
Probably increasing the filling ratio to 100% will cause the largest time saving. With the current attachment method a filling ratio of 100% is not feasible, however the effect of an automated system for attaching roses to the sorting machine may be investigated.
Another possibility is to further increase the fork flow. However, the current machine control computer cannot handle a higher fork flow. When this limitation is fixed, the fork flow could probably be increased further and the next fork flow limitation may be investigated. This means probably a mechanical limit or an ergonomic limit for the rose attachment employees. There is also a possibility that the roses will get

damaged during attaching and machine handling when the fork flow is further increased. Some of the mentioned problems may however be solved by an automated attaching system.
The bunching station acceptance ratio may be increased a little more, however this adjustment will probably result in relatively large mechanical changes and only 0.6 hours per day may be saved extra.

The last option is to add even more attachment locations. At this moment this is not feasible because of the qualifying system, that is mounted at the end of a sorting machine. However the options to modify the qualifying system and the extra time that may be saved, may be investigated.

Except for investigating the bottlenecks as a whole, only a part of the bottleneck may be improved. For instance within the belt system, only the priority rules for the buffer belts may be changed to enlarge the sorting capacity. Another example is to develop a smart technology that predicts the number of roses for the coming period and predicts the time of each buffer belt or bunching station that is full. If it is possible to empty the buffers before they are entirely full, it is probably possible to prevent that multiple buffer belts are full at the same time.

An interesting phenomenon that showed up is the difference in mindset of employees of different companies. This is visible at the employees that are working on the buffer belt after the sealer. In some companies the buffer belt is mostly nearly overflown and in other companies the buffer belt is mostly nearly empty. This mindset may influence the capacity of the entire rose sorting system. The productivity of the attachment employees also influence the capacity of the entire rose sorting system. It is interesting to investigate the cause of this effect and the increase of the sorting capacity that may be achieved.

9 References

4More Technology. (2017). Retrieved January 26, 2017, from www.4mt.nl

aweta. (2017). Retrieved January 26, 2017, from www.aweta.nl

Cassandras, C. G., & Lafortune, S. (2010). *Introduction to Discrete Event Systems* (2nd ed.). New York: springer.

G-fresh. (2016). Retrieved December 15, 2016, from http://www.g-fresh.nl/nl/nieuws/20161117-grower-of-the-week-porta-nova-with-red-naomi

Gray, B. M. A. (2007). Discrete event simulation: A review of simevents. *Computing in Science and Engineering*, *9*(6), 62–66.

Grimmelius, H. T. (2005). *Condition monitoring for marine refrigeration plants*. Delft University of technology.

Henten, E. J. Van. (2006). Greenhouse Mechanization : State of the Art and Future Perspective. *Acta Hortic*, *710*, 55–70.

Hilberink, A. (2005). *Generating knowledge with a software model "A knowledge-based expert system for condition monitoring of an hydraulic brake system."* Delft university of technology.

Hoog, J. de. (2001). *Handbook for modern greenhouse rose cultivation*. Aalsmeer: Applied Plant Research.

IRISS. (2016). Retrieved February 15, 2017, from https://www.4mt.nl/iriss

Macdonald, I., & Strachan, P. (2001). Practical application of uncertainty analysis. *Energy and Buildings*, *33*, 219–227.

Porta Nova. (2017). Retrieved January 26, 2017, from http://www.portanova.nl/

Red Naomi. (2017). Retrieved January 26, 2017, from http://www.rednaomi.com/nl/home-nl/

Schriber, T. J., & Brunner, D. T. (2005). Inside Discrete event Simulation Software: How It Works and Why It Matters. In *Simulation Conference, 2009 Proceedings of the Winter* (pp. 158–168).

Van, A., Bontsema, J., & Henten, E. J. Van. (2013). Sensitivity analysis of a stochastic discrete event simulation model of harvest operations in a static rose cultivation system. *Biosystems Engineering*, *116*(4), 457–469.

Van ’t Ooster, A., Bontsema, J., Van Henten, E. J., & Hemming, S. (2012). A discrete event simulation model on crop handling processes in a mobile rose cultivation system. *Biosystems Engineering*, *112*(2), 108–120.

Verbeek, J. (2016). Hoe de rozenkassen langzaam verdwijnen uit het Nederlandse landschap. Retrieved from https://fd.nl/economie-politiek/1169913/hoe-de-rozenkassen-langzaam-verdwijnen-uit-het-nederlandse-landschap

Appendix A Paper

The paper is attached on the next pages.

Optimization of a Rose Sorting system using discrete event simulation

*Wouter Vreugdenhil, Yusong Pang, Dingena Schott, Wouter Bac**

Delft University of Technology (Section of Transport Engineering and Logistics, Delft, The Netherlands)

**TechNature B.V. Jan Dorrekenskade-Oost 1A, 2741 HT, Waddinxveen*

Abstract

In Rose cultivation companies in the Netherlands, there is a demand for a higher sorting capacity on the existing sorting systems. The objective for this research is to advice which part of the sorting system needs to be adjusted to gain a higher sorting capacity. For the current sorting systems, five bottlenecks are defined, the bottlenecks limit the sorting capacity. To be able to forecast the effect of machine adjustments, a discrete event simulation model has been constructed, using Simulink and Matlab. This simulation model is verified, matched and validated using data of on an existing rose sorting system. Results of a single day validation showed that the time to process the roses can be simulated with a 97% accuracy. Subsequently, five different simulations are executed. In each simulation, one of the five bottlenecks is removed or reduced. With the results of these simulations the capacity limitation due to each bottleneck is quantified. However entirely removing a bottleneck is not feasible in reality for all bottlenecks. A last situation is simulated where all feasible bottleneck reductions are combined. This showed that the time to sort all roses is reduced by 35%.

1. Introduction

In rose cultivation greenhouses, roses are grown, sorted and packaged. The financial value of a rose is dependent on its appearance. Factors like stem length, bud size, color and ripeness determine the quality of roses. Rose growers in the Netherlands face a lot of competition from rose growers that are settled in Africa (Verbeek, 2016). Many Dutch rose growers went bankrupt in the past two decades. The Dutch rose growers that remain, are the rose growers with a large company and differentiate themselves by higher quality of large-flowered roses.

Since mid-90 there was hardly any innovation in the sorting process of roses. In the last two years, the company 4More Technology developed a new rose grading system called IRISS (Intelligent Rozen Inspecteer en Sorteer Systeem) ("IRISS," 2016). This system may be installed on an existing rose sorting machine. Using IRISS the quality class of the rose may be determined much more precisely. Therefore more quality classes may be distinguished, however the number of quality classes is also dependent on the number of outlets on the sorting machine.

Due to the aforementioned reasons, there is a demand for larger sorting machines (more outlets) with a higher sorting capacity (roses/hour).

There are many ideas for adjusting the rose sorting system machines, to enlarge the sorting capacity. These ideas come from rose sorting employees, greenhouse owners and 4More Technology employees. However there is no method to predict the increase of capacity applying possible machine adjustments.

This research focuses on the rose cultivation company PORTA NOVA ("Porta Nova," 2017), which is the biggest rose grower of the cultivar "Red Naomi" and is famous for the high quality of the roses("G-fresh," 2016). PORTA NOVA is

rose production of one day (90.000 roses) in one working day (8 hours).

2. Material & method

At first, the rose sorting process and the bottlenecks are described. Subsequently, a simulation model is developed using discrete event simulatio. This simulation model is verified with an existing rose sorting system. Finally the simulation model is used to simulate the basic situation of PORTA NOVA 3 and to investigate the influence of each bottleneck.

Rose sorting system

An example of a rose sorting system is shown in Figure 1.

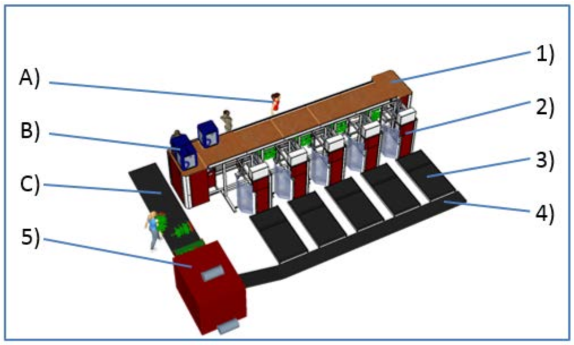


Figure 1 example of rose sorting system

A rose sorting system consists of five standard machines. 1) Sorting machine: the sorting machine contains a long chain, with forks attached to it. This chain rotates through the sorting machine. At location A) employees attach roses to the forks of the sorting machine. At location B) the roses are qualified by IRISS. Depending of the quality class of the rose, the rose is handed over to a bunching station. 2) Bunching station: multiple bunching station are part of a sorting system. A bunching station is linked to a quality class and will receive only roses of that quality class. The bunching station transforms the separate roses into bunches of roses. 10 roses are tied together and the stems are all cut to the same length. When the bunching station is full, it will transfer a bunch from the bunching station to the accessory buffer belt. 3) Buffer belt: roses are sold per 5 bunches in a bucket. To make sure 5 bunches of 1 quality class arrive directly after each other at location C), 5 bunches are buffered on the buffer belt. When five bunches are on the buffer belt and space is available on the V-belt, the buffer belt will start transferring bunches to the V-belt. 4) V-belt: the V-belt transfers bunches of roses from the buffer belts to the Sealer. 5) Sealer: the sealer packages each bunch in a plastic wrapping. A last buffer belt is attached to the sealer where the packaged bunches are buffered. Finally an employee executes a last quality check of the bunches and puts the bunches in a bucket.

The number of bunching stations and buffer belts may vary per sorting system. The more bunching stations are present, the more quality classes may be distinguished. The buffer belts, V-belt and sealer may even be left out if desired. Then the transport and packaging of the roses is executed by hand.

Bottleneck identification

Many adjustments of the sorting system to enlarge the capacity may be thought of. The identification of the bottlenecks is executed by analyzing all processes. However the adjustments to the sorting capacity may be deduced to five bottlenecks.

The theoretical maximum sorting capacity in the current situation is equal to the fork flow (forks/hour) of the sorting machine. This fork flow itself is the first bottleneck. The following three limitations may cause that the sorting capacity is not equal to the fork flow. The second bottleneck is the filling ratio. The employees at location A) (Figure 1) are not able to fill all the forks of the sorting machine. The filling ratio is defined as (full forks)/(total number of forks) over a certain timespan. The third bottleneck is the bunching station limitations. The bunching stations are not able to accept all roses which pass by in the sorting machine. This is caused by some actions in the bunching station, during which the bunching station cannot accept any roses. For instance, when a string is wrapped around the stems or when a bunch is transferred to the buffer belt. When a bunching station cannot accept a rose, the rose will make an extra round in the sorting machine. The fourth bottleneck is the belt system (buffer belts, V-belt and sealer). When multiple buffer belts are full in a short time span, the V-belt and sealer cannot handle all the bunches from the buffer belts immediately. When a buffer belt and the accessory bunching station is full, the bunching station cannot accept roses anymore. The fifth and last bottleneck doesn't cause a limitation in the current sorting systems, but makes it possible to enlarge the theoretical maximum capacity. This may be done by adding an extra rose attachment location at the other end of the sorting machine. With this adjustment, the forks of the sorting machine may process up to two roses per round.

Discrete event simulation

In discrete event simulation (Cassandras & Lafortune, 2010) or a discrete event system, traffic units (entities) flow through the system(Schriber & Brunner, 2001). These entities respond to events. Events are occurrences that change the state of the model. Many events may occur in a model, it starts with the event of creating entities. The entities move between servers, where operations are executed and queues where the entities can wait. Further there are control elements that control the logic and delays based on the systems state.

The Matlab environment using Simulink and SimEvents is used to construct the simulation model. This is a commonly used program for discrete event modelling (Gray, 2007; Van 't Ooster, Bontsema, Van Henten, & Hemming, 2012). In Simulink a mathematical model may be constructed as a block diagram. The library SimEvents in Simulink contains the blocks where entities can flow through.

Simulation model

In reality, a rose sorting system consists of multiple standard machines, which may be composed in several ways. This is also the case in the simulation model. For each standard machine, a subsystem is created. This subsystems may be composed in several ways. Per subsystem, there are variables which may be edited to simulate different simulations. The variables are listed below.

1) Sorting machine

- Fork flow: number of forks per hour.
- n_fork: number of forks in sorting machine.
- Filling ratio: ful forks / total number of forks
- Rose datasheet: sheet with roses and accessory quality classes which have to be processed in the simulation.

2) Bunching station (times for action within bunching station)

- Time fork change 1
- Time fork change 2
- Time create bunch
- Time move arm out
- Time move arm back
- Error log: an error log can be added to a bunching station. In an error timeout the bunching station cannot accept roses.

3) buffer belt

- Time to move bunches to the end of the buffer belt

4) V-belt

- No variables

5) Sealer

- No variables

Verification Matching and Validation

The steps of verification, matching and validation are based on the structure of Grimmelius (2005).

In the verification, the behavior of a single bunching station is tested and the priority rules of the buffer belts are tested.

For matching the simulation model, a basic situation is simulated. Afterwards each input variable is changed and the direction of change of the output variables, processing time and number of roses rejected, is checked.

To execute the validation of the simulation model, an existing situation of a greenhouse is

simulated. The situation of Meewisse Roses is taken to execute the verification. Meewisse has a sorting machine with 20 bunching stations. Bunching stations 3 – 14 are equipped with buffer belts, the rest of the bunching station is not equipped with buffer belts. Bunching station & buffer belts 3 – 7 are equipped with one V-belt and sealer, and bunching stations and buffer belts 8 – 14 are equipped with another V-belt and sealer.
The most important output parameter is the processing time. The simulation has a 3% deviation in processing time compared to reality. The number of roses offered per bunching station showed an average deviation of 2.5% with a maximum of 10.7%

Simulations

For the case PORTA NOVA 3, 7 different situations are simulated. Each simulation is executed 10 times. For situation 1) Basic, the entire setup of machines and all the input variables are discussed.

1) Basic

In the basic setup of PORTA NOVA 3, the sorting machine is extended to 36 bunching stations (Figure 2). 18 bunching stations are positioned on both sides of the sorting machine. All bunching station are equipped with a buffer belt. Per 6 bunching station and buffer belts, one V-belt and sealer is equipped.

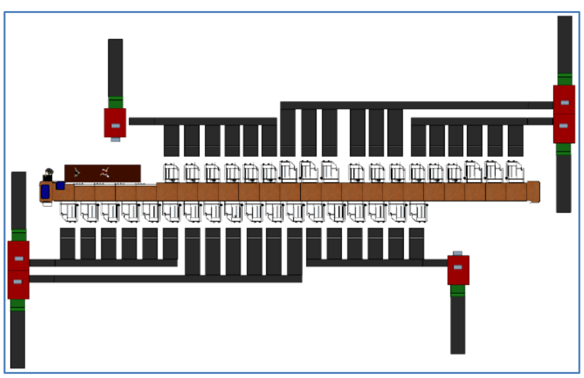


Figure 1 basic setup of PORTA NOVA III

The input variables are defined as follows:

- Fork flow: 9500 Forks/hour. This is similar to the current situation in PORTA NOVA I & II.
- n_fork: 416 forks.
- Filling ratio: 84%. Similar to current situation in PORTA NOVA I & II.
- Rose dataset: The rose datasets from one day from PORTA NOVA I & II combined. This sheet containes 90.192 roses.
- Time fork change 1: 1.89 seconds.
- Time fork change 2: 4.93 seconds.
- Time create bunch: 3.79 seconds.
- Time move arm out: 3.79 seconds.
- Time move arm back: 1.89 seconds
- Error log: no error log is used
- Time to move bunches to the end of the buffer belt: 13 seconds.

2) Filling ratio

For situation 2 and the following situations, only the differences from the basic situation are discussed. In this situation the filling ratio bottleneck is removed. This means that a filling ratio of 100% is used

3) Fork flow

In this situation, the fork flow bottleneck is reduced. The fork flow is enlarged to 12.000 forks/hour, which is the maximum of the current sorting machine.

4) Bunching station busy

In this situation, the bunching station actions bottleneck is removed. All the time variables in the bunching station subsystems are set to 0 seconds.

5) Belt system flooding

In this situation, the belt system bottleneck is removed. To remove this bottleneck, the entire belt system is removed.

6) Two attachment locations

In this situation, a second attachment location is added to the sorting machine.

7) Feasible

In this situation, all feasible improvements are combined. A sorting system setup without belt system and with a second attachment location is used. The input variables which are changed are:

- Fork flow: 12.000 forks/hour
- Filling ratio: 87%
- Time fork change 1: 0.3 seconds
- Time fork change 2: 0.3 seconds

- Time create bunch: 1.5 seconds
- Time move arm out: 3.0 seconds
- Time move arm in: 4.2 seconds

3. Results

The results are shown in bar graphs. Each situation is executed 10 times and the difference in result is displayed with error bars.
The first and most relevant result is the time to process all 90.192 roses in the dataset. This is shown in Figure 3.

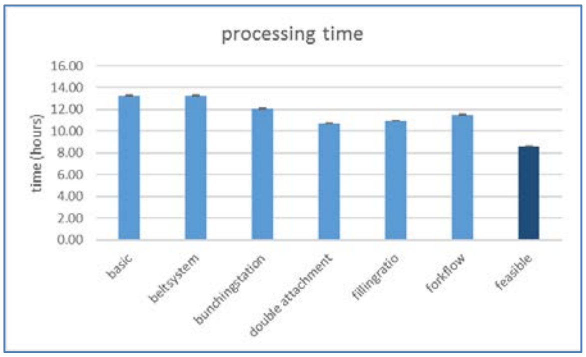


Figure 2 time to process all roses in datasheet per situation

Other results, which provide more insight in the reason of the time reduction are shown in Figure 5 and Figure 5. In Figure 4, the number of roses offered accepted and rejected, per situation is shown.

Figure 3 number of roses offered, accepted and rejected per situation

The number of roses rejected is split up in different causes and is shown in Figure 5.

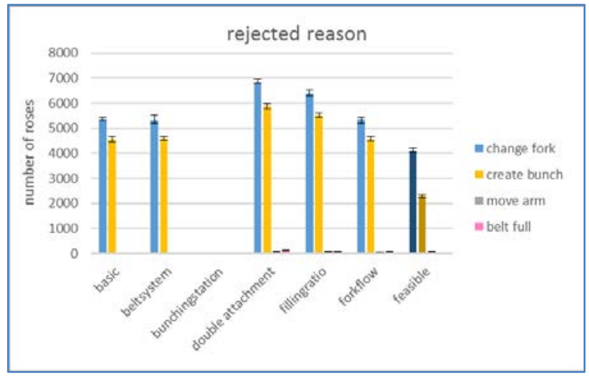


Figure 1 number of roses rejected per reason of rejection per situation

4. Discussion

Five situations show an improvement in processing time relative to the basic situation, except the belt system. The belt system situation did not show an improvement. In the basic situation no roses are rejected because of the belt system (Figure 5). This explains why removing the belt system does not reduce the processing time.
In the situation where the bunching station bottleneck is removed the processing time is reduced by 9%. In Figure 5, it can be seen that this time reduction is caused by the decrease of rejected roses.
Using a second attachment location, the processing time is reduced by 19%, however the number of roses rejected is increased instead of decreased.
With a 100% filling ratio also the processing time is decreased and the number of roses rejected has increased.
When the fork flow is increased, the number of roses rejected per reason is nearly equal to the basic situation. However the processing time decreases.
With all feasible adjustment, the processing time decreases from 13.3 to 8.6 hours and the number of roses rejected decreases from 9965 roses to 6457 roses. In this situation it takes 0.6 hours to process the 6457 roses.

5. Conclusion

1) With discrete event simulation, the advantages and disadvantages of machine adjustments can be identified clearly.

2) With the feasible machine adjustment, the processing time can be reduced from 13.3 hours to 8.6 hours. This is not the desired reduction to 8 hours.

3) Introducing an extra attachment location, increase fork flow, increase filling ratio and decrease time for bunching station action all decrease the processing time.

6. Recommendations

The verification is executed on data of only one day in reality. It is recommended to verify the model on multiple days at multiple greenhouses with different setups.

For further decrease of processing time, multiple options are left.

The fork flow can be increased further. It is interesting to investigate what would be the next limitation for the fork flow when the software of she sorting machine is updated.

The filling ratio is not jet 100%, so it is possible to further increase.

A little time may be saved with further improvement of the bunching stations.

The last option is to add even more attachment locations. At this moment this is not possible because the IRISS system can only be installed at the end of the sorting machine.

7. References

Cassandras, C. G., & Lafortune, S. (2010). *Introduction to Discrete Event Systems* (2nd ed.). New York: springer.

G-fresh. (2016). Retrieved December 15, 2016, from http://www.g-fresh.nl/nl/nieuws/20161117-grower-of-the-week-porta-nova-with-red-naomi

Gray, B. M. A. (2007). Discrete event simulation: A review of simevents. *Computing in Science and Engineering*, *9*(6), 62–66.

Grimmelius, H. T. (2005). *Condition monitoring for marine refrigeration plants*. Delft University of technology.

IRISS. (2016). Retrieved February 15, 2017, from https://www.4mt.nl/iriss

Porta Nova. (2017). Retrieved January 26, 2017, from http://www.portanova.nl/

Schriber, T. J., & Brunner, D. T. (2005). Inside Discrete event Simulation Software: How It Works and Why It Matters. In *Simulation Conference, 2009 Proceedings of the Winter* (pp. 158–168).

Van 't Ooster, A., Bontsema, J., Van Henten, E. J., & Hemming, S. (2012). A discrete event simulation model on crop handling processes in a mobile rose cultivation system. *Biosystems Engineering*, *112*(2), 108–120.

Verbeek, J. (2016). Hoe de rozenkassen langzaam verdwijnen uit het Nederlandse landschap. Retrieved from https://fd.nl/economie-politiek/1169913/hoe-de-rozenkassen-langzaam-verdwijnen-uit-het-nederlandse-landschap

Appendix B, IRISS

This Appendix provides some background information about IRISS.

IRISS is the new Intelligent Rose Inspection and Sorting System (Intilligent Rozen Inspectie en Sorteer Systeem) developed by 4More Technology. IRISS is more accurate and is able to distinguish more quality classes for the roses. Also a user friendly user interface and information to analyze the roses and subdivision of roses in quality classes is provided. With the option to create more quality classes the demand for extra outlets on the sorting machine arose.

IRISS consists of camera's, computers with image processing software, databases and a user interface. The various parts may be installed on the current sorting machines. This is shown in Figure Appendix B1.

Figure Appendix B1 IRISS installed on sorting machine

Measurement improvements

In Table Appendix B1 the measurement differences between the old system and IRISS are depicted.

- IRISS is able to measure more characteristics. The old system is able to measure the bud height, bud width, stem thickness and stem length. IRISS also measures the color, ripeness, stem bend and neck angle.
- The quality of the images is much better.
- IRISS also has better image processing software. This is shown by the measurements of the bud width. IRISS is able to distinguish the drooping leaf on the left from the bud, while the old system is not able to do that.
- IRISS takes pictures from multiple directions. Using those images, a 3 dimensional shape is calculated. An additional advantage of the camera's on multiple angles is that if the desired information cannot be measured from one image, it is probably available in the other image. In Table Appendix B1 it is shown that the bottom of the stem is behind a leaf so that the stem length cannot be calculated. In the image from the other direction the bottom of the stem is clearly visible.

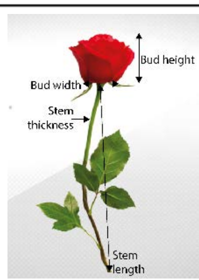
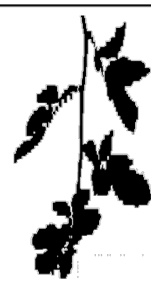
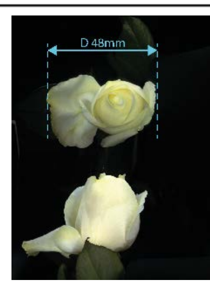
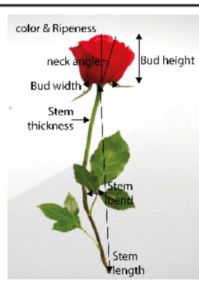
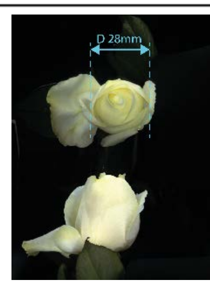
	characteristics	Image quality	Image processing	Dimensions
Old system				
IRISS				

Table Appendix B1 measurement differences, old system - IRISS

User interface

In the old system the quality classes and other settings and data may be accessed via the computer shown in Figure Appendix B2. IRISS is accessible via a Windows pc with user interface. In Figure Appendix B2 the screen to define the quality classes is shown. In the main table the quality classes are defined. Each row represents a quality class. In the columns the minimum and maximum for each property is defined. The number of rows, the numbers in the table and the order of the rows may be changed by the user. The quality class with the highest value per rose is at the top and the quality class with the least value per rose is at the bottom of the table. The properties of each rose scanned by IRISS are compared to each rule shown in this screen, starting at the upper rule. When the properties of the rose doesn't match the first rule, it is compared to the second rule and so on. The rose is classified in the first rule where all constraints are satisfied.

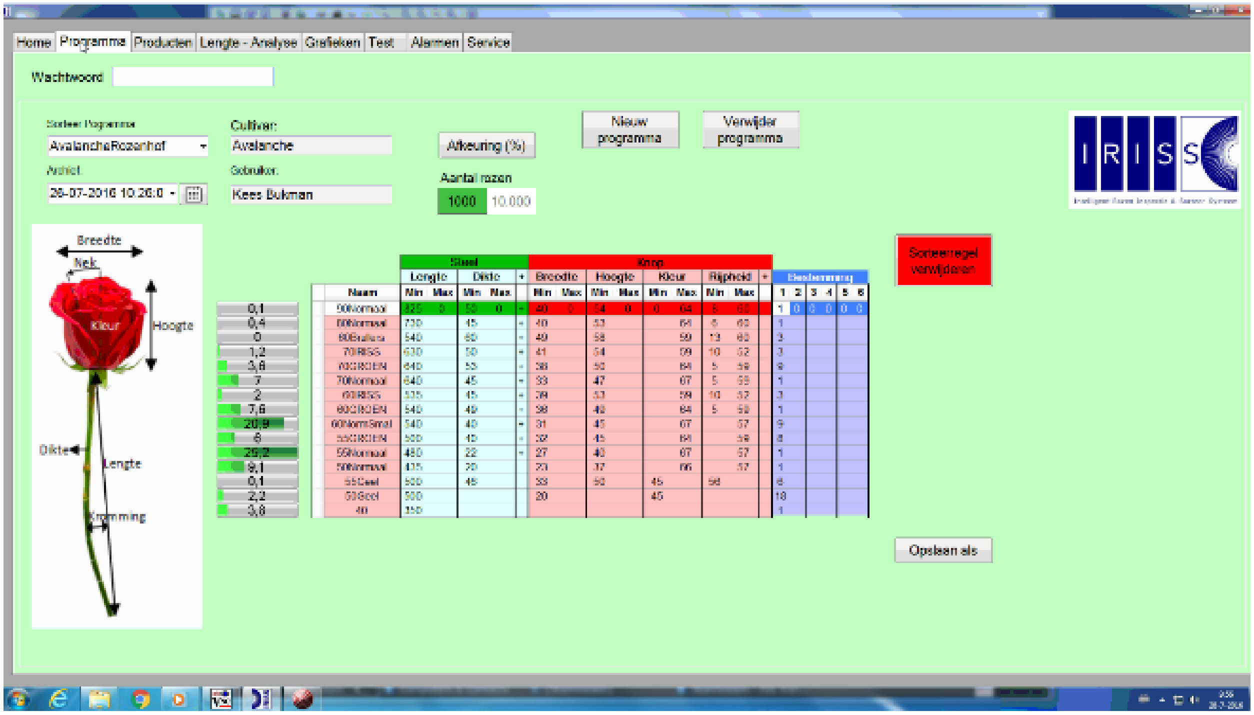


Figure Appendix B2 IRISS user interface

In the other tabs there is more information available to keep an eye on the roses being processed. For instance it shows how many roses are being cut to a shorter length. This happens because some roses are rejected at a certain quality class as result of the measurement of another property than the stem length. This is not desired, because most times it means loss of value. Via IRISS it can be seen how many roses are rejected at a certain quality class per property and in which quality class the roses end up. More tools are available to manage the qualification process.

Computational time

In the old system the image processing was the limiting factor for the speed in terms of Roses per hour of the sorting machine. In the IRISS system the image processing is not the limiting factor anymore. This means that the speed of the sorting machine may be increased. In some greenhouses the sorting machine runs already at a higher speed.

Effect

The use of IRISS results in less waste in terms of money, but also in terms of cut off stems, and the ability to define more quality classes. Another large advantage is the overview of the sorting process, that is

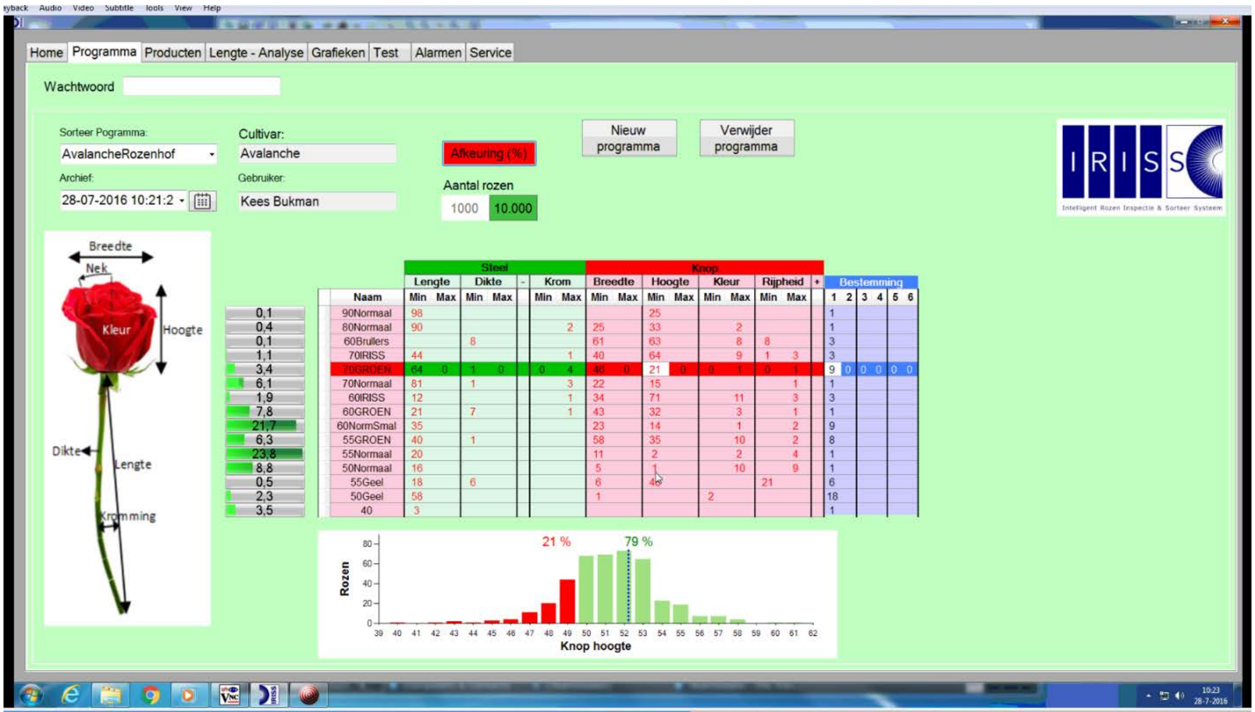


Figure Appendix B3 IRISS GUI with histogram of bud height in quality class "70GROEN"

provided by IRISS. An example is shown in Figure Appendix B3. The user clicked on bud height in the rule “70 GROEN”. A histogram appears, that shows the bud height of the roses that fulfill all other constraints within the quality class “70 GROEN”. With this histogram insight in the effect of changing the minimum bud height is provided.

Appendix C detailed simulation results

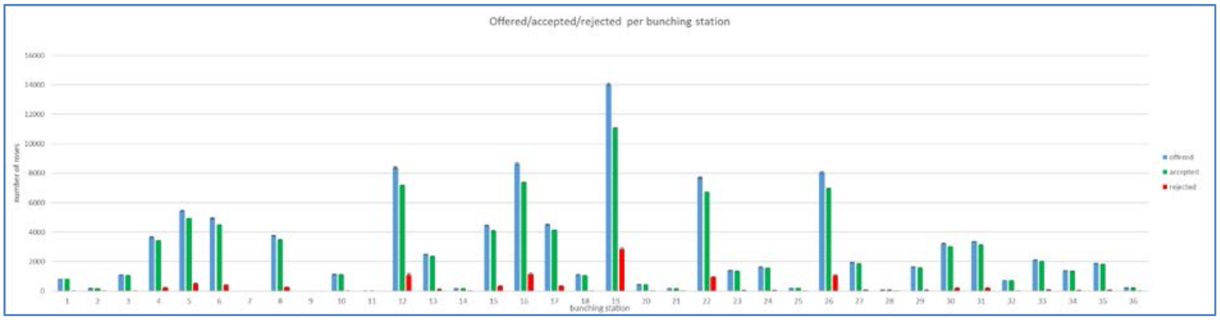


Figure Appendix C1 Basic situation, number of roses offered, accepted and rejected per bunching station

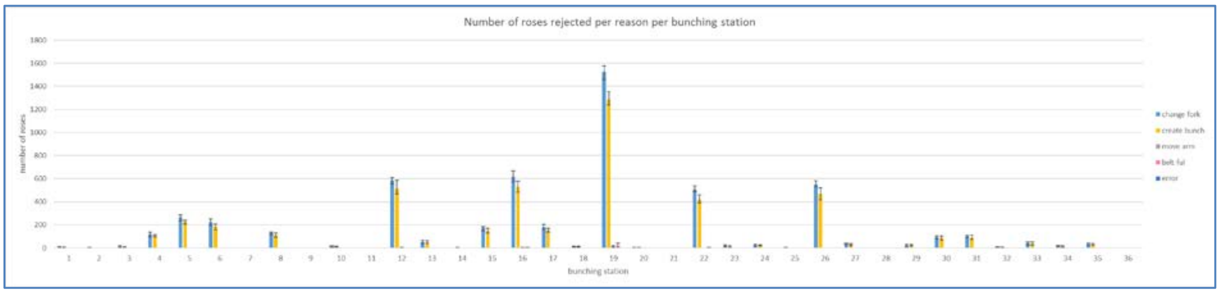


Figure Appendix C2 Basic situation, number of roses offered per reason, per bunching station

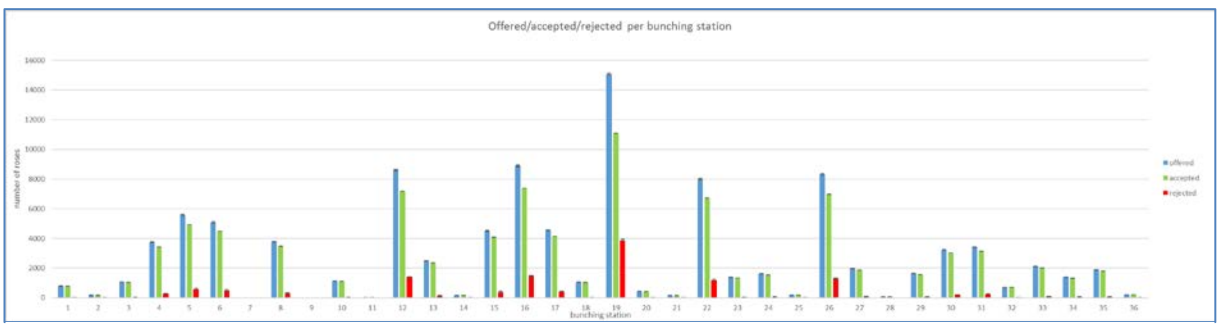


Figure Appendix C3 double attachment, number of roses offered, accepted and rejected per bunching station

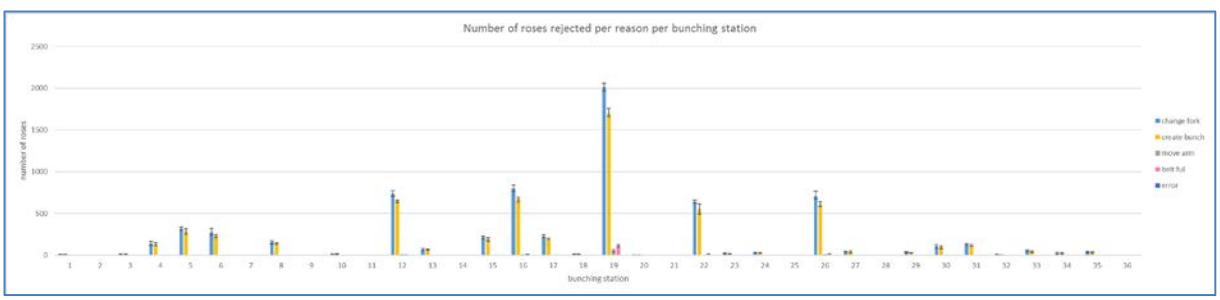


Figure Appendix C4 double attachment, number of roses offered per reason, per bunching station

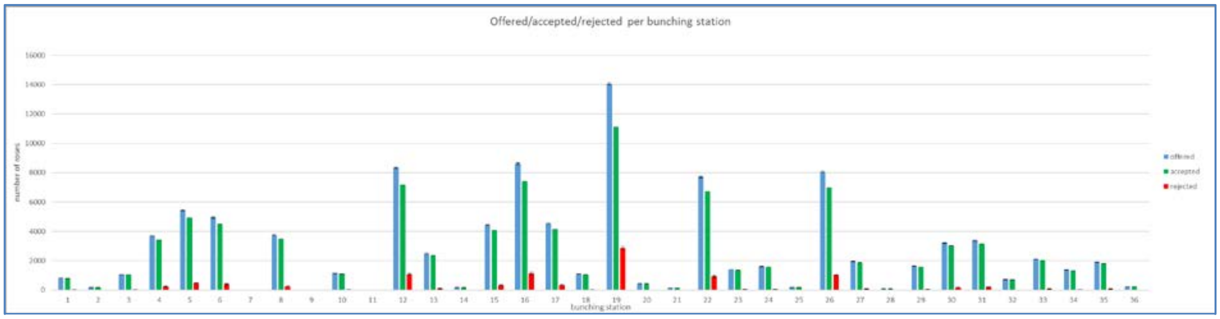


Figure Appendix C5 beltsystem, number of roses offered, accepted and rejected per bunching station

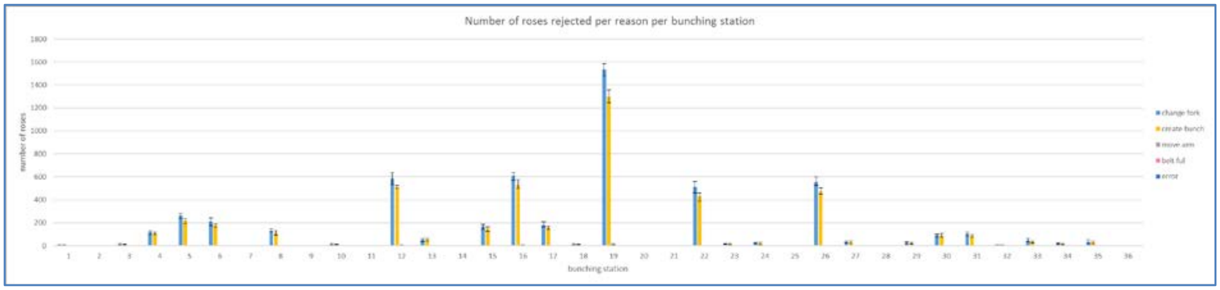


Figure Appendix C6 beltsystem, number of roses offered per reason, per bunching station

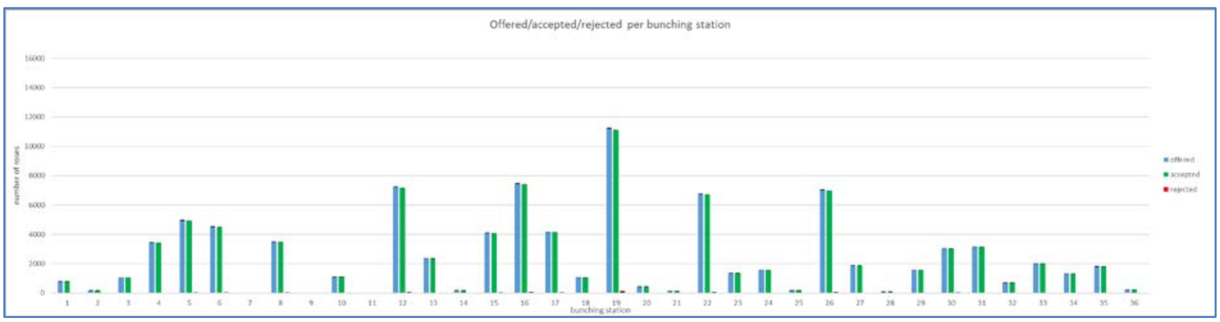


Figure Appendix C7 bunchingstationbusy, number of roses offered, accepted and rejected per bunching station

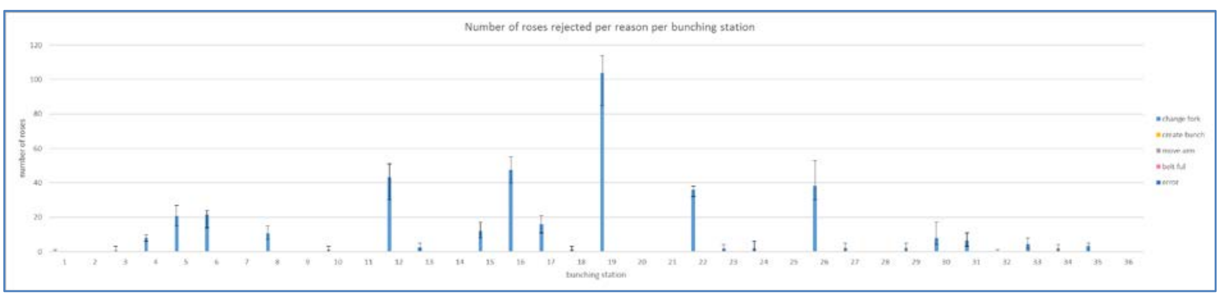


Figure Appendix C8 bunchingstationbusy, number of roses offered per reason, per bunching station

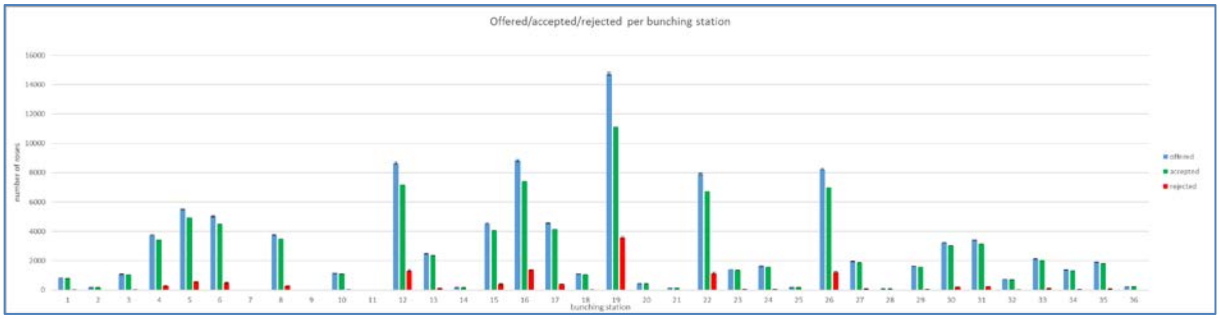


Figure Appendix C9 fillingratio, number of roses offered, accepted and rejected per bunching station

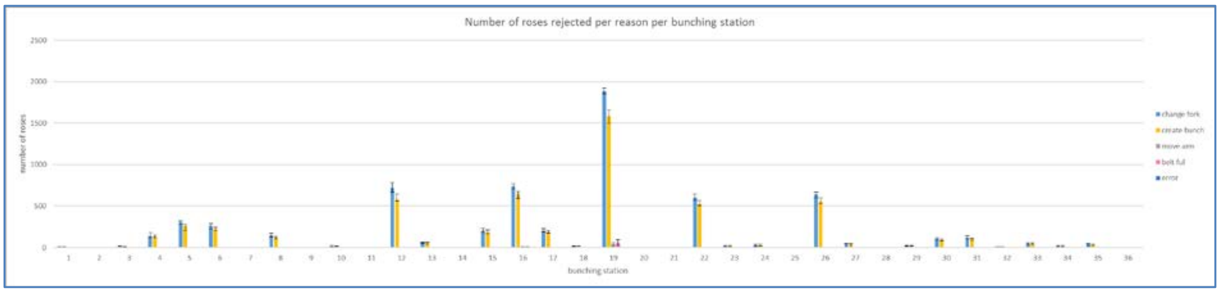


Figure Appendix C10 fillingratio, number of roses offered per reason, per bunching station

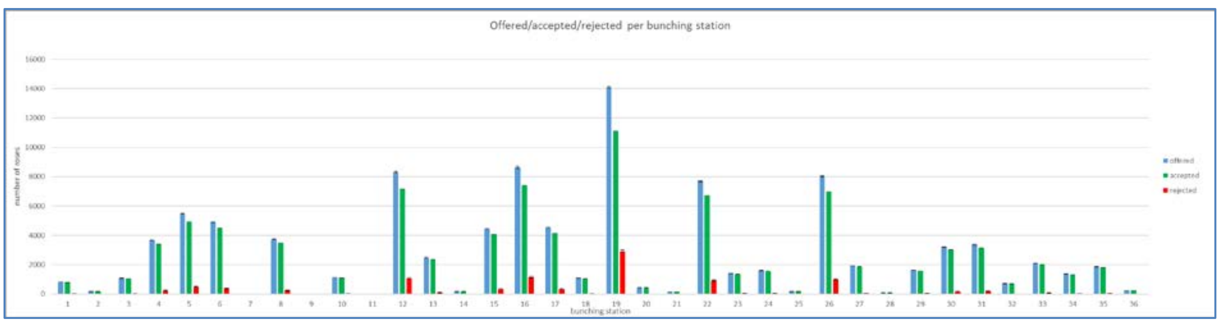


Figure Appendix C11 forkflow, number of roses offered, accepted and rejected per bunching station

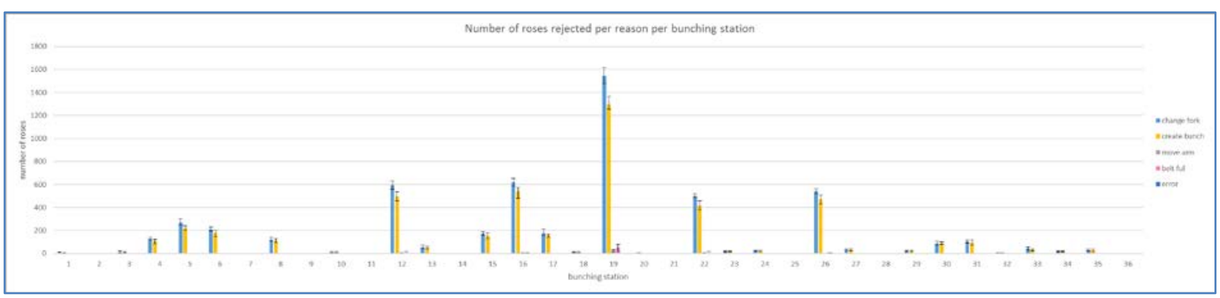


Figure Appendix C12 forkflow, number of roses offered per reason, per bunching station

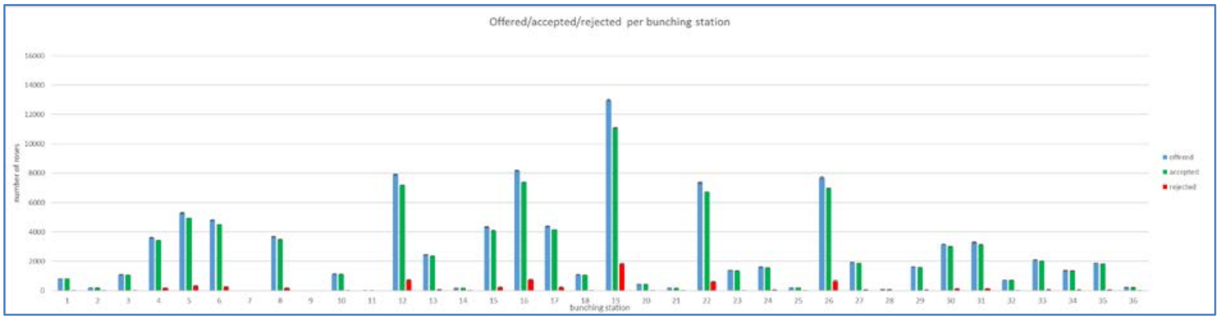


Figure Appendix C13 feasible, number of roses offered, accepted and rejected per bunching station

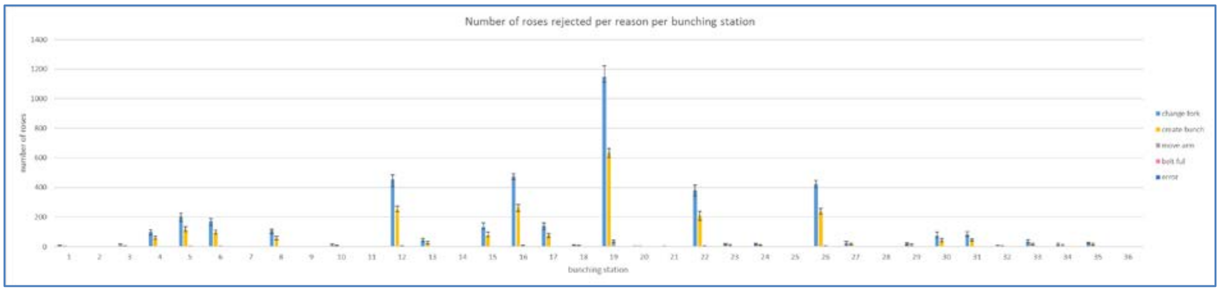


Figure Appendix C14 feasible, number of roses offered per reason, per bunching station

Appendix D bunching station actions

During the measurements that are executed to be able to construct a simulation it has been noticed that the bunching station may act in different ways. It has been found that these machines perform differently at different greenhouses. From these observations it was expected that the bunching stations may perform better than they currently do.

In this chapter at first the different parts and processes within a bunching station are discussed. Thereafter the differences between greenhouses are shown and at last the possibilities and restrictions to optimize the bunching station are described.

Parts and processes

As explained in chapter 1, a bunching station collects the roses of one quality class. So the sorting machine exterts the roses at the right bunching station. In Figure , the moving parts of the bunching station are shown.

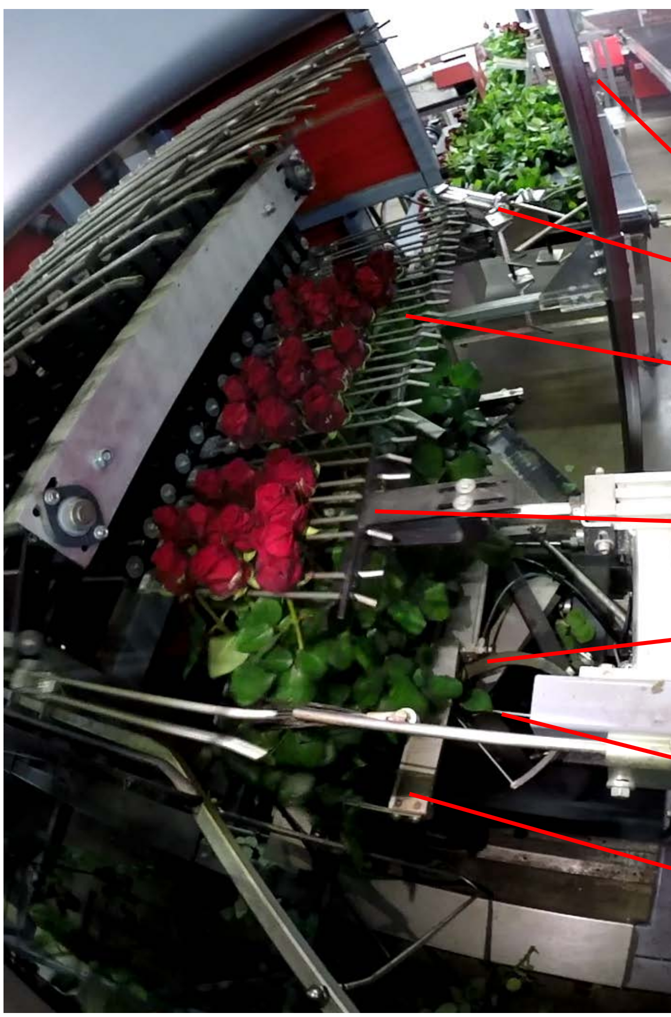


Figure Appendix D1 bunching station parts

In Figure it is shown that this bunching station makes bunches of 10 roses. On the first collector fork of each bunch 3 roses are collected, at the second collector fork 4 roses are collected and at the third collector fork 3 roses are collected. Each bunch is followed by 1 empty collector fork. This is called a 3-4-3-0 configuration. Other common configurations are 2-3-3-2-0-0- and 5-5-5-5-0.

Some parts of the bunching station can move such as the bud pusher, stem pusher, collector forks etc. Three different sequences of movement of these parts may be distinguished.

- Normal collector fork change
- Collector fork change and exert bunch
- Collector fork change and attach string around stem of the bunch (create bunch)

In time lines (Figure) the order of movements in the bunching station for each sequence is shown. To measure these times a video is recorded of all these processes and analyzed by watching the video's in slow motion. To be able to know when the bunching station can or cannot accept roses the sorting machine is set such that all roses are sent to one bunching station.

It can be seen that during a part of the processes no roses may be accepted by the bunching station. In this case during the time no roses may be accepted 8 roses in the sorting machine pass the bunching station during the normal fork change, 9 roses during fork change and attach string and 9 roses pass during collector fork change and exert bunch. In the latter process there is also some time during which no fork change may occur.

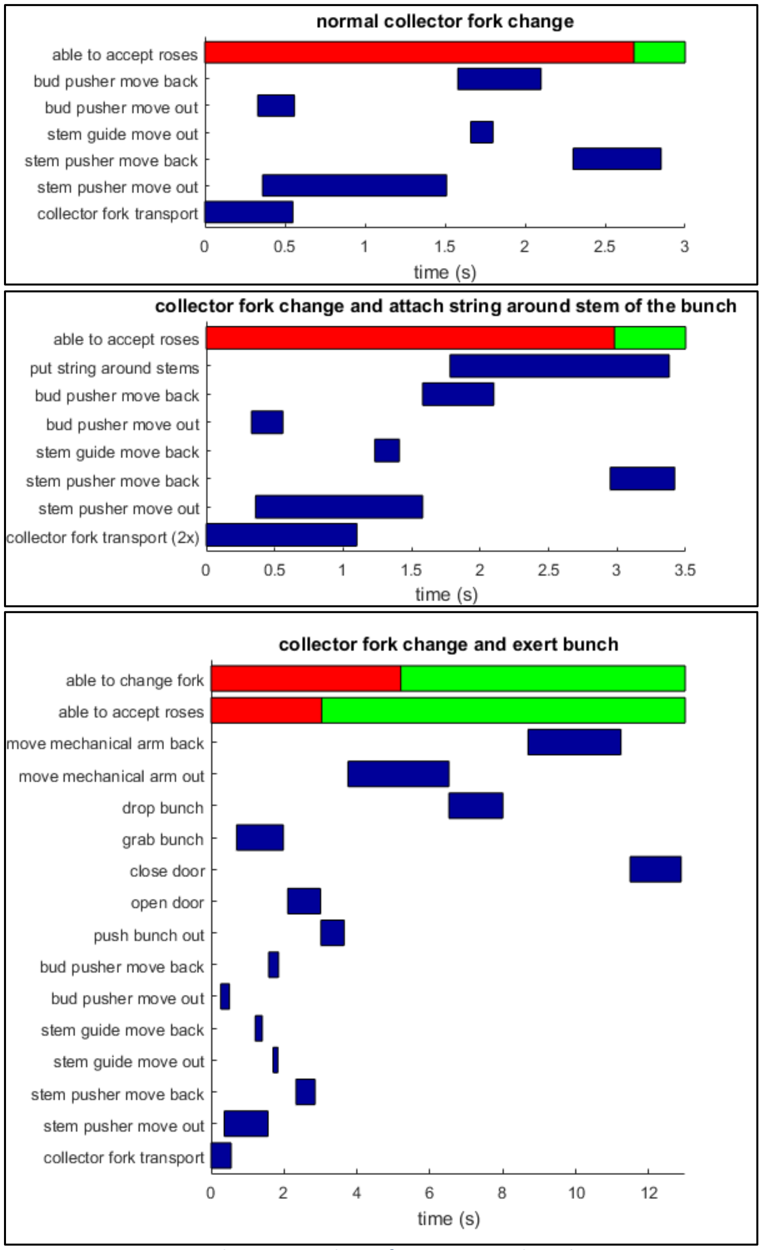


Figure Appendix D2 Time line of processes in bunching station

Differences between greenhouses

The only thing important for the discrete event simulation is the number of roses that could not be accepted during the fork change processes. Therefore this is measured in multiple greenhouses and on multiple bunching stations within the greenhouses. A distinction has to be made between normal bunching stations and speeded up bunching stations. In a speeded up bunching station, the electromotor, that provides the motion of the collector fork is changed for a better one and a variable frequency drive is added. The variation in the number of roses, that cannot be accepted during fork change is shown in Table Appendix D1.

It is notable that there are differences between the bunching stations in different greenhouses, but there are also differences between bunching stations, that are within one greenhouse.

At first the differences between the bunching stations within the same greenhouse are discussed. These differences were not that large. In terms of number of roses rejected, the maximum difference between bunching stations within one greenhouse was 2. These deviations seemed to have two causes. Most of the moving parts in the bunching station operate on compressed air. The pressure used on each activator is tunable with an adjustable valve. In this way the time for each movement may be influenced. The second reason is deferred maintenance. Some pistons are worn, which results in longer movement times.

	Normal bunching station		Speeded up bunching station	
process	Rejected roses		Rejected roses	
	minimum	maximum	minimum	maximum
Normal collector fork change	3	10	2	2
Collector fork change and exert bunch	4	11	2	2
Collector fork change and attach string around stem of the bunch (create bunch)	9	13	6	6

Table Appendix D1 Number of rejected roses per process

The differences between the bunching stations from different greenhouses are much bigger. This has another cause. In the machine the starting time of each movement is adjustable. Big differences in these settings are present without a known reason.

Because of these differences the presumptions arose that the bunching stations may perform better.

Bunching station optimization

Optimizing the bunching stations is done by adjusting the settings and the compressed air adjustable valves. The processes are recorded on video and analyzed in slow motion. The results of this optimization is shown in Table .

During the optimization some restrictions occurred. The company 4More Technology cannot edit the software of the bunching station jet. Therefore there is no complete freedom in adjusting the starting time of the processes and not all dependencies are totally clear. This is most clear in the process of the collector fork change and attach string around the stem of the bunch. In Figure it is shown that the movements are almost the same as a normal collector fork change except the extra movement of the binder. In the optimized situation the processes are still very similar. However at the collector fork change and attach string 8 roses are rejected and at a normal fork change 2 roses are rejected. The cause is probably in the software. There is no physical limitation to accept roses earlier. If the software could also be edited, the number of roses rejected could be reduced from 8 to 2.

	Normal bunching station	Speeded up bunching station
process	Rejected roses	Rejected roses
Normal collector fork change	2	1
Collector fork change and exert bunch	2	1
Collector fork change and attach string around stem of the bunch (create bunch)	8	5

Table Appendix D2 Number of roses rejected per process, optimized