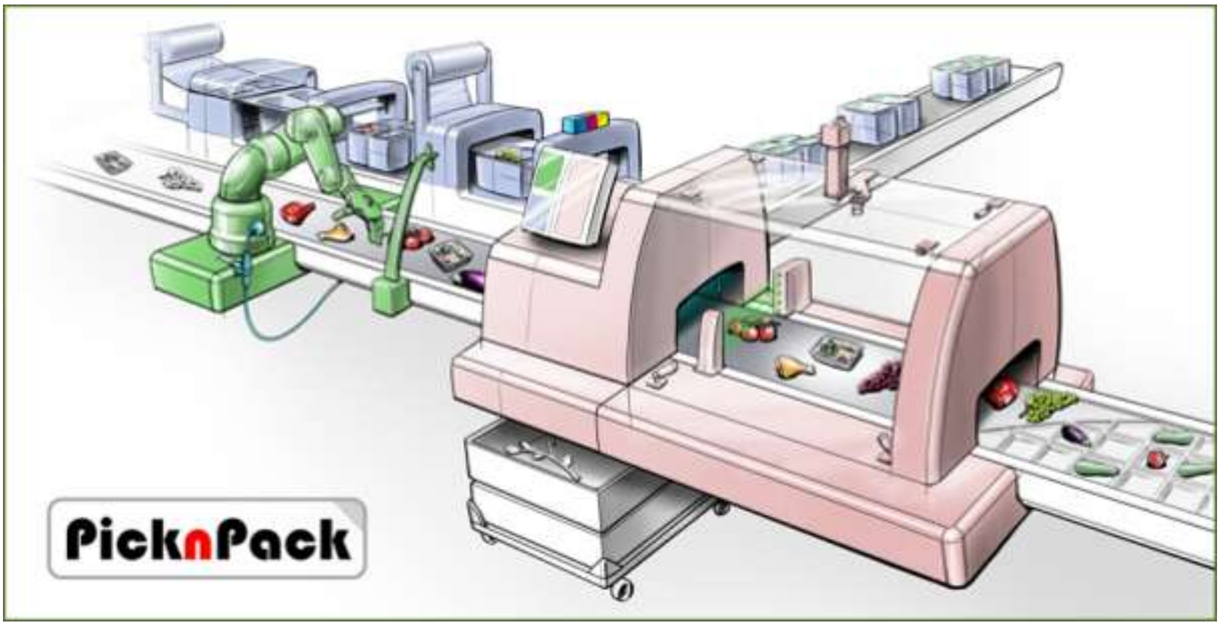
D5.4– Report of robot performance in production line

DLO, Lacquey, MAREL, Fraunhofer, Tecnalia

29/02/2016



Flexible robotic systems for automated adaptive packaging of fresh and processed food products

The research leading to these results has received funding from the European Union Seventh Framework Programme under grant agreement n° 311987.

Table of Contents

Dissemination level		
PU	Public	X
PR	Restricted to other program participants (including the EC Services)	
RE	Restricted to a group specified by the consortium (including the EC Services)	
CO	Confidential, only for members of the consortium (including the EC Services)	

1 Introduction 2
2 Modules performance 4
2.1 Cable robot 4
2.1.1 Communications 4
2.1.2 Control implementation 6
2.1.3 Kynematics 10
2.2 IPL robot 13
2.2.1 Communication line control 13
2.2.2 Vine tomatoes 14
2.2.3 Chicken breast 18
2.2.4 Grape vines 21

1 Introduction

This document is a deliverable for the Pick and Pack project, the "Report of robot performance in production line", the fourth deliverable for the Work package 5.

The objective of this deliverable is to describe the robot performance in production line conditions. In the previous deliverable D5.3 it was described the robot performance in test conditions and in this document it is described the robot performance too, once its final configuration has been achieved.

The production line what it is named in the scope of this project, is the Picknpack line and it includes the different modules developed in the project.

As WP5 has two main modules, this deliverable is divided in 2 parts: Pickable (cable robot) and IPL (Marel) Robot.

Figure 1: Picknpack line general view (Wageningen)

IPL Robot is located in Wageningen now, integrated with Pickable line.

Figure 2: IPL robot general view in Wageningen

Pickable has its own conveyor, so can be set up in an independent way, before its shipment to Wageningen.

Figure 3: Pickable robot general view

2 Modules performance

2.1 Cable robot

2.1.1 Communications

After the last implementations, Tecnalia Cable Robot is totally integrated with the PnP complete line. Due to hardware constrains, this implementation has been implemented in a second computer different from the robotic control computer with no final integration of the complete system. It is able to receive and send communications signals/events and data, and thanks to that, it is able to accomplish these expected functionalities:

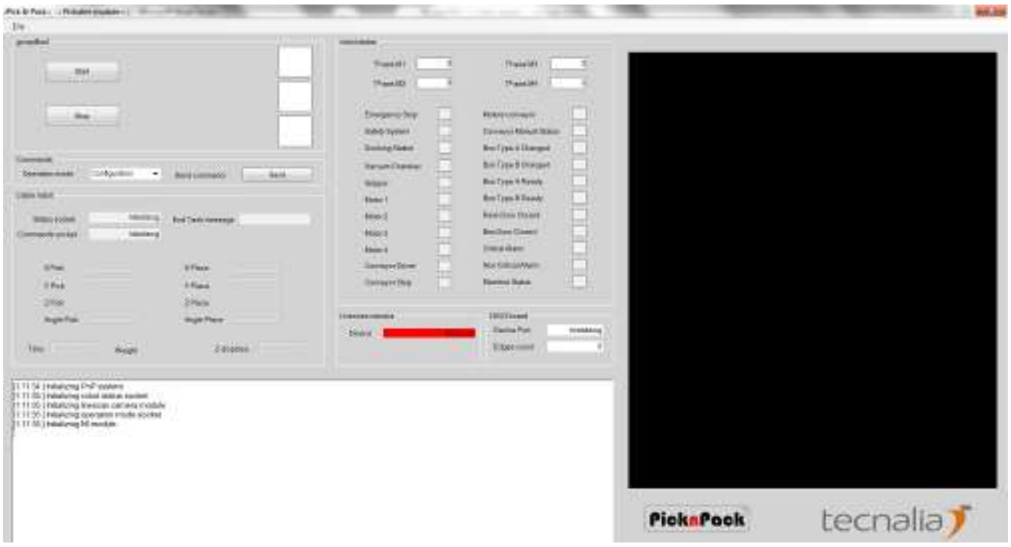
- Direct and "clean" connection to the PnP line, without not changes needed, as long as the subnet accomplishes the Zyre based network requirements: UDP and broadcast allowed, and port 5760 opened.
- Flexible line configuration changes implemented by Cable Robot in realtime thanks to world data model reception, parse and process. This allows real time changes related to batch, tray and punnet types and their dimensions.
- Cable Robot data shipment to the Line in order to be shown in general line GUI. This data includes for example information about final crates used.

The following picture shows (see computer screens) Cable robot integrated with PnP general line controller, receiving and sending signals/events and data in real time:

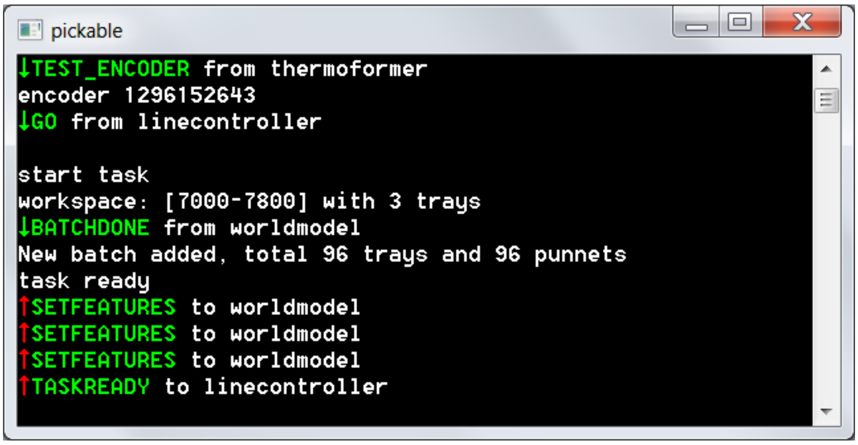
Figure 4: Cable robot integrated with PnP general line controller

If we analyse the image, it can be been:

1. The Pickable own GUI running in the Vision PC:



2. In the same machine the communication thread responsible for communicate Pickable with the rest of the line:



3. In the laptop the line controller is running and tracking all the process, and the line GUI can be seen:

2.1.2 Control implementation

As Pickable is an innovative robot patented by Tecnalia, its modelling and control is very challenging and has to take into account many constraints: 1/ Models precision while they has to be as simple as possible to be implemented in a real time environment, and at the same time 2/ developing a control strategy that takes into account the fact that cables are able only to pull and not to push. Always maintaining the cables in tension is then very important.

Many steps have characterised the development and implementation of the control of Pickable robot. They can be summarized in the following table:

Step 1: Simulation of the models and their transcription and validation from MATLAB to C++
Step 2: Implementation and testing of model within the real time RTX environment
Step 3: First communication tests with motors and encoders within RTX and control in torque of each motor separately (without platform)
Step 4: Implementation of the tensioning strategy of the cables
Step 5: Pick and place cycles

Table 1: The different steps of implementing the control strategy

<u>Step1:</u> *Simulation of the models and their transcription and validation from MATLAB to C++*

Different functionalities have been developed and simulated, among them the trajectory generation and the kinematic models of the robot (inverse kinematics and Jacobian matrix).

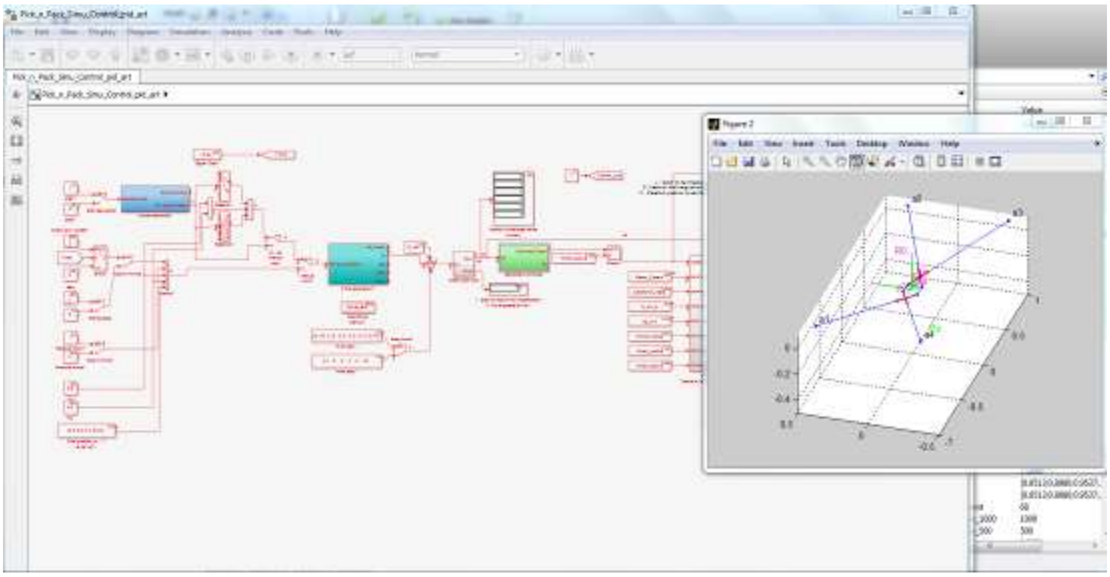


Figure 5: Graphical simulation of the robot kinematics with Matlab/Simulink

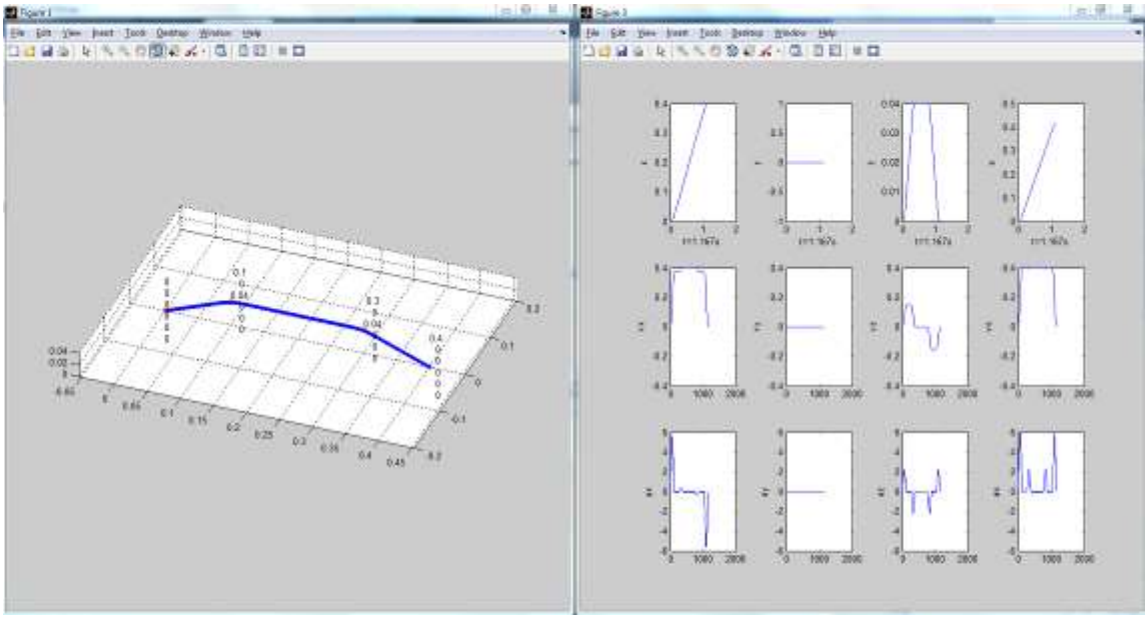


Figure 6: Simulation of the trajetory generation module within MATLAB

Once the models and functions have been validated on Matlab/Simulink, they have been then transcripted in C++ and validated within Visual Studio environment.

<u>Step2:</u> *Implementation and testing of model within the real time RTX environment*

The real time environment used for the control of Pickable is RTX. Its main advantage is its ability to transform a hard real-time system on a standard PC Hardware, transforming Windows into a Real-Time Operating System (RTOS).

Many examples and sample projects enables to start with the RTX environment. This led to an easy start with a simple RTX example and then all the kinematic models, trajectory generation and control functions were implemented within Visual Studio under an RTX environment.

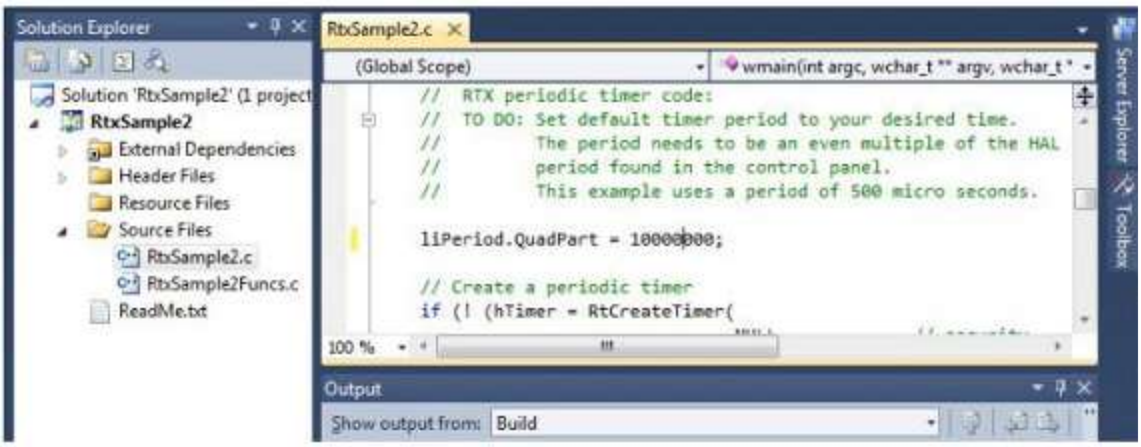


Figure 7: RTX getting started example

Step3: *First communication tests with motors and encoders within RTX and control in torque of each motor separately (without platform)*

The specificity of the control of Pickable robot is that that it has to take into account the use of cables as an actuating element, as they cannot loose their tension during the movement of the robot. Otherwise, this can lead to many vibrations and an instability of the robot.

For the control of Pickable, an innovative control strategy developed by Tecnalia has been developed and takes into account the necessity of always keeping in tension the cables. In our case, the use of the standard control of the drives is not possible. That is why a crucial step in our case is to totally open the controllers that can directly send torque control references. Once this parametrization of torque control achieved in the ETEL drives, the first step is to control the robot in position and tune the gains in order to have a good performance in terms of setlling time, overshoot and static error.

Using a simple Ziegler Nichols method, we have been able to have a first set of parameters. Starting from that, we adjusted empiracaly the parameters in order to have a most suitable behaviour. An example of step signal reference test si shown in the following figure.

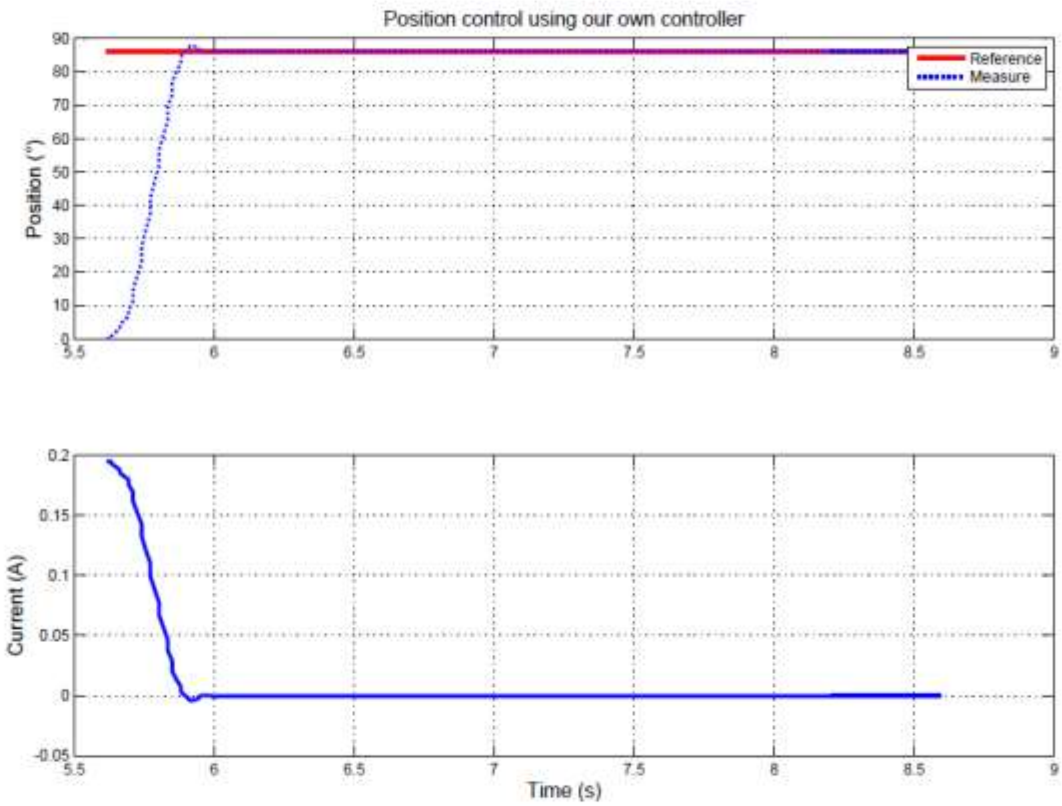


Figure 8: Example of a step signal control of one axis (torque mode + own control strategy)

Steps 4 and 5: *Implementation of the tensioning strategy of the cables and Pick and place cycles*

When the platform was mounted, the first test was to put in tension the cables during the movement. This enables us to make the first movements and then to tune the gains in order to achieve the best performances in terms of good tracking of trajectories.

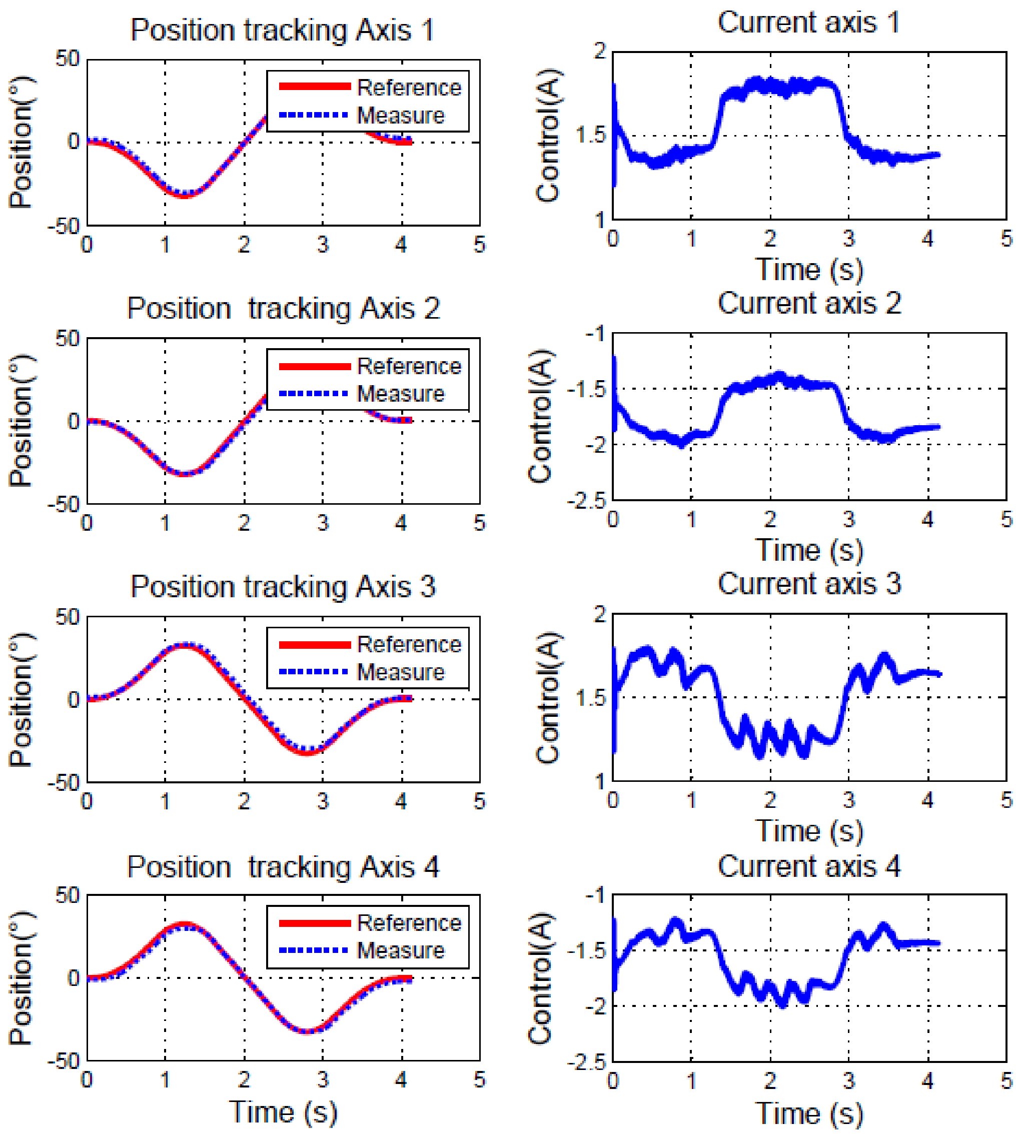


Figure 9: A tracking test with our own control + trajectory generation module (within the RTX environment)

The last step of the tuning was to reduce the overshoot and retune the control of the motors 5 and 6 that are embedded on the platform. As these motors are controled using Comet software from ETEL. It was possible, to easily perform this tasks after an identification and autotuning process. The results are shown in the two following curves that show the performances before and after tuning.

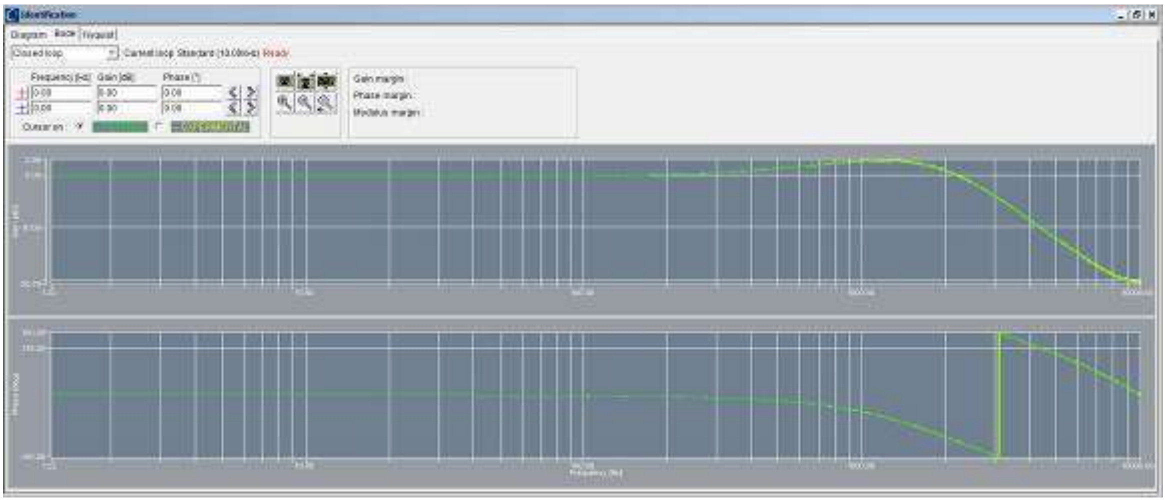


Figure 10: Closed loop bode diagram before tuning the gains of motor 6 (high overshoot)

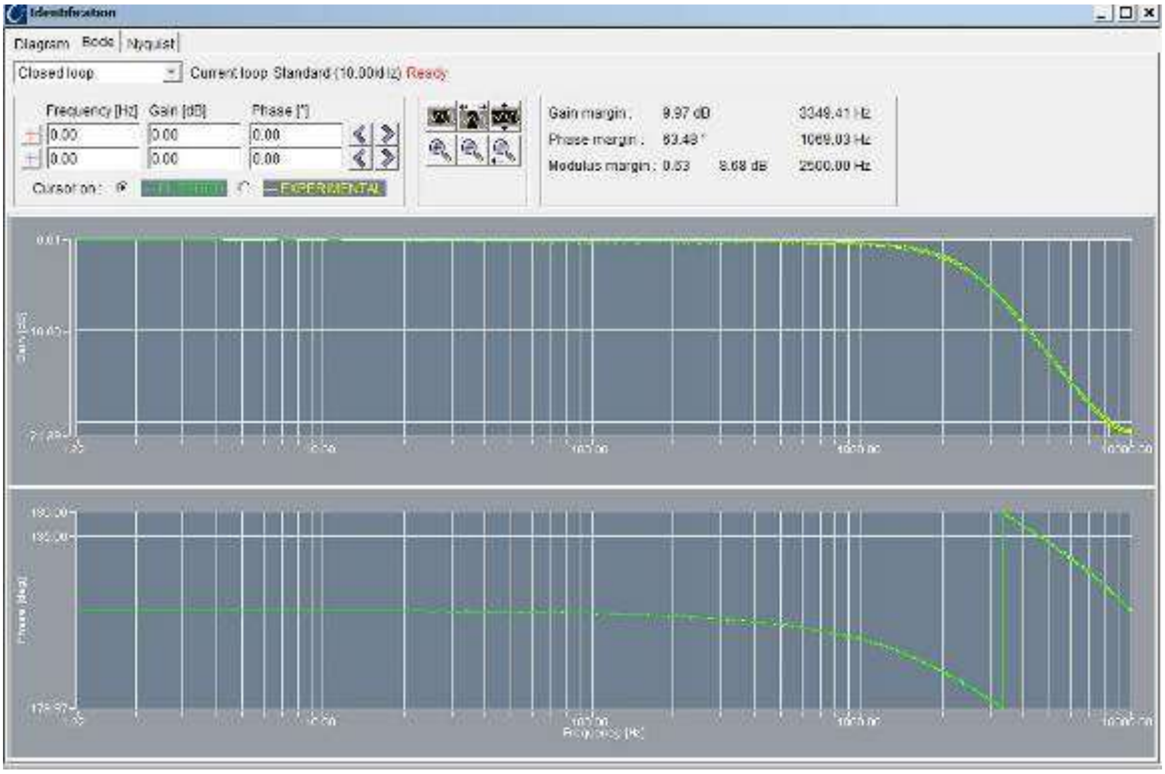
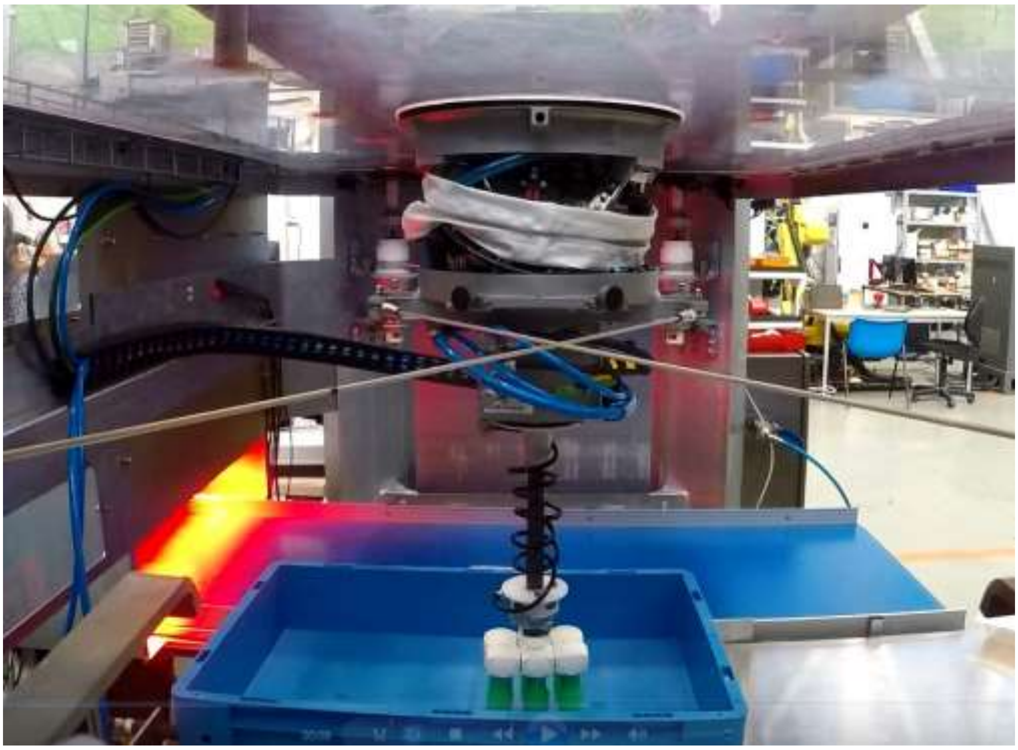


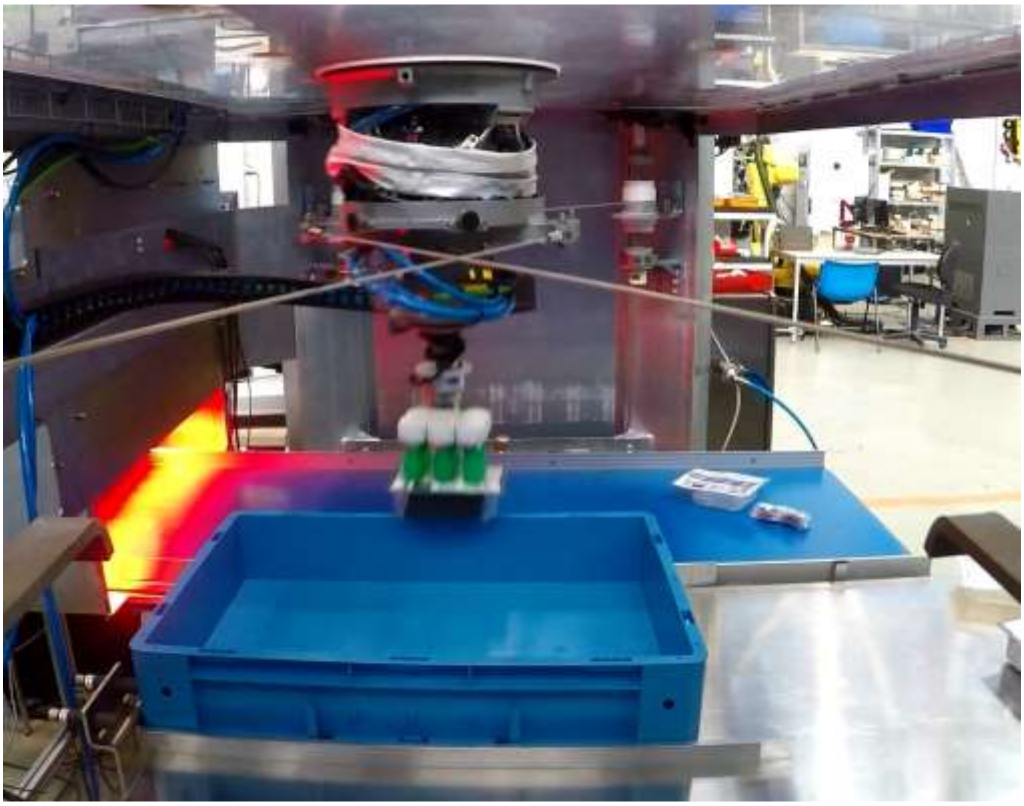
Figure 11: Closed loop bode diagram after tuning the gains of motor 6 (no overshoot)

As a conclusion, all these tests enable us to execute pick and place tests at the targeted cadence of 30 cycles/min.

2.1.3 Kynematics

The following images are examples of cycles of Pickable:



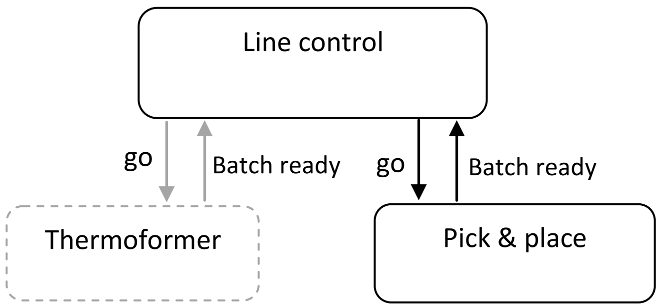


To see the robot performance access to 'Pickable by Tecnalia' in the Youtube webpage.

2.2 IPL robot

The Pick & Place robot is tested in the PnP line without a moving web of trays (simulated Thermoformer). Three products were tested: vine tomatoes, chicken breasts and grape vines. The quantitative and qualitative results as well as further improvements are described in the next sections.

2.2.1 Communication line control



In the conducted tests the line control module simulates the Thermoformer. When the Pick & Place robot is ready to act, that's when it's able to pick the first product, it sends a " ready message" to the line control. When all modules are ready, the line control can ask the Thermoformer to make the first batch. The line control receives a batch "ready message" from the simulated Thermoformer and immediately sends a "go message" with or without data of the batch to the Pick& Place robot. *In the start " start up* phase" , that's before the first batch is in the working area of the Pick & Place robot, the "go message" is not containing any batch data, and the Pick & Place robot immediately sends "task ready message", so the Thermoformer can make the next batch. When the first batch arrives in working area of the robot the "go message" is containing tray data of the batch and the robot can start to act. (The working area of the robot is configured in the line controller set-up.) The Pick&Place robot picks the product in the bin, moves the gripper to a pouch position and places the product. When the product is placed, the robot sends product data (Bin Id, Image no., weight, etc.) to the world model, the key is the tray and punnet ID's. After all the trays in the batch have been filled, the Pick & Place module signals the line control it has finished by sending a "task ready message", whereupon the cycle repeats.

The first version of the communication between the Line control / world model and the Pick & Place robot was tested in the period December to February. 1/ A continues production were the vison data were simulated and more than 2000 products (350+ batches) processed with success. 2/ By testing the vision and the different grippers, the line controller were used to simulate the thermoformer movements.

Ultimo February a new version of the line controller were released, see description above. This was implemented and tested with success primo march.

By the tests the Bin Id were simulated, because the RFID receiver is not yet mounted and the signal from this receiver trough the line control system to the Pick & Place robot is therefore not tested.

The "hard stop message" / "soft stop message" function is not yet implemented because if the robot is stopped , it has to be started again at the Robot HMI. Change of new product (batch layout) is not yet tested.

2.2.2 Vine tomatoes

Good results were obtained for the vine tomatoes. A summary of the quantitative results is given in Table 2. *The pick success in the bin* was 95% (20/21). This measures whether the vision software indicates correctly the position where the gripper should grab the vine tomato. *The grip success in the bin* was 90% (19/21). This measures whether the gripper holds, drops or has an unstable hold of the vine tomatoes once the gripper closed and moved up. *The transport success from bin to the web of trays* was 100% (19/19). This measures whether the gripper holds the vine tomatoes during the transportation to the pouch. Only in one instance the vine tomatoes were damages.

Pick success: in bin.

Tomatoes. 7 bins, 22 vine tomatoes, Single layer: singled out		Correct	Not correct
Vision software judgement	Pick	21	0
	No Pick	0	1

Grip success: in bin.

Tomatoes. 7 bins, 21 tomatoes, Single layer: singled out		Arm/Gripper action			
		Hold	Unstable grip	Drop	Missed/not in gripper
Gripper Action	Pick	19	0	2	0

Transport success: from bin to WoT

Tomatoes. 19 tomatoes Single layer: singled out		Arm/Gripper action		
		Hold	Unstable grip	Drop
Gripper Action	transport	19	0	0

Table 2: Vine tomatoes, quantitative results

Observations

1. Bunches with sufficient free space around get grapes well in general.
    a. Sometimes, a bunch does not get grasped well and slides off the gripper.
    b. Sometimes, a bunch slides somewhat in the gripper, but remains grasped.
2. The fingers, before grasping, and after placing the product, do not fully open.

3. The gripper, especially the upper parts with the motor block and the axis for the finger joints, is too wide. Bunches close to the sides of the crate can therefore not be picked.
4. The dragging motion, necessary to create free space for concatenated bunches, fails. Some observations:
    a. Setting the height for dragging is delicate. A little too high means that the fingers will miss the tomatoes. A little too low means that the fingers will collide with the tomatoes.
    b. When one of the tomatoes sticks out of the row, the fingers will collide with it.
    c. What happens during dragging
        i. The robot moves to the sides (usually)
        ii. First, the fingers at one side touch the tomatoes and the fingers open more
        iii. When fingers are maximally open, the bunch starts to move
        iv. Usually, the bunch does not move enough (because of the fingers opening)
        v. Then, the gripper goes directly down, whereas the bunch is now not centred in the gripper
        vi. The tomatoes get damaged
5. The fingers are often not nicely aligned, causing some fingers to collide with the tomatoes.
6. The fingers now have sticky coating at both sides. The outside should preferably be frictionless.

Actions

1. WUR
    a. Setting the opening of the grippers is now based on the average position of the tomatoes at either side of the stalk. If one of the tomatoes is out of line, the aperture is miscalculated, resulting in collisions with the tomatoes. Instead, calculate the bounding box and use that to calculate the aperture.
2. Lacquey
    a. Create improved fingers, with a small bend and with rubber click-on material at the insides of the fingers. See fingers that are used in the Koppert project. This creates better grip on the tomato bunches.
    b. Create a flexible, compliant tip of the fingers, to allow for some robustness in determining the z-position of the dragging action. The above-mentioned design already has a similar functionality.
    c. Fingers are now oriented outwards when objects of 100-1200mm wide are grasped (see figure 12). This causes the product to be pushed out. By changing the position of the rotation axes for the fingers, a more profitable orientation with respect to the object can be obtained. This will also create a better finger orientation for the grapes.
    d. The rotation axes are now sticking out. To prevent collisions of these parts of the gripper with the crate, we have to use a large collision bounding box. The

smaller the collision bounding box the better. If the rotation axes are placed inside, see Figure 12 and also previous point. This could be improved.

e. Footprint of the gripper seems unnecessarily big (see also previous point). One issue is the motor block, which is now placed somewhat too low. The distance from motor block to bottom of the finger tips should be at least the height of the crate, so that it cannot collide with the crate and the fingers can get closer to the sides of the crate. Temporary solution could be to lower the crate edges by putting a plate on the bottom of the crates.

f. With the previous, mind that the Marel robot cannot move upwards much more than it does now.

3. Marel

   a. Full opening of the gripper (before grasping and after placing the product) is now position controlled. This should become torque control, so that the fingers fully open, and thus also realign.

   b. After dragging the bunch to the sides, the fingers are misaligned. To make sure that the gripper is again nicely aligned with the product, the gripper should move back a little, see Figure 13. NB. Only movements in the direction perpendicular to the orientation of the truss should be compensated, see Figure 14.

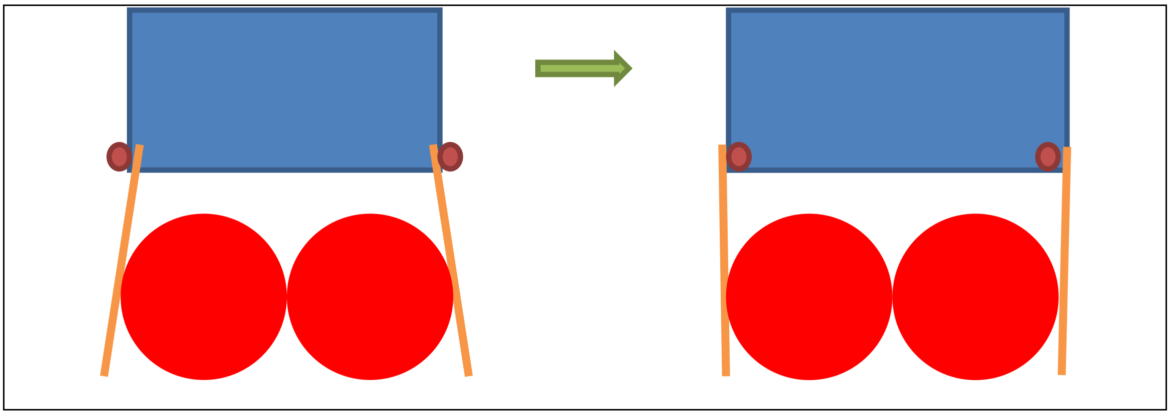


Figure 12: Different rotation point to improve the orientation of the finger with respect to the product

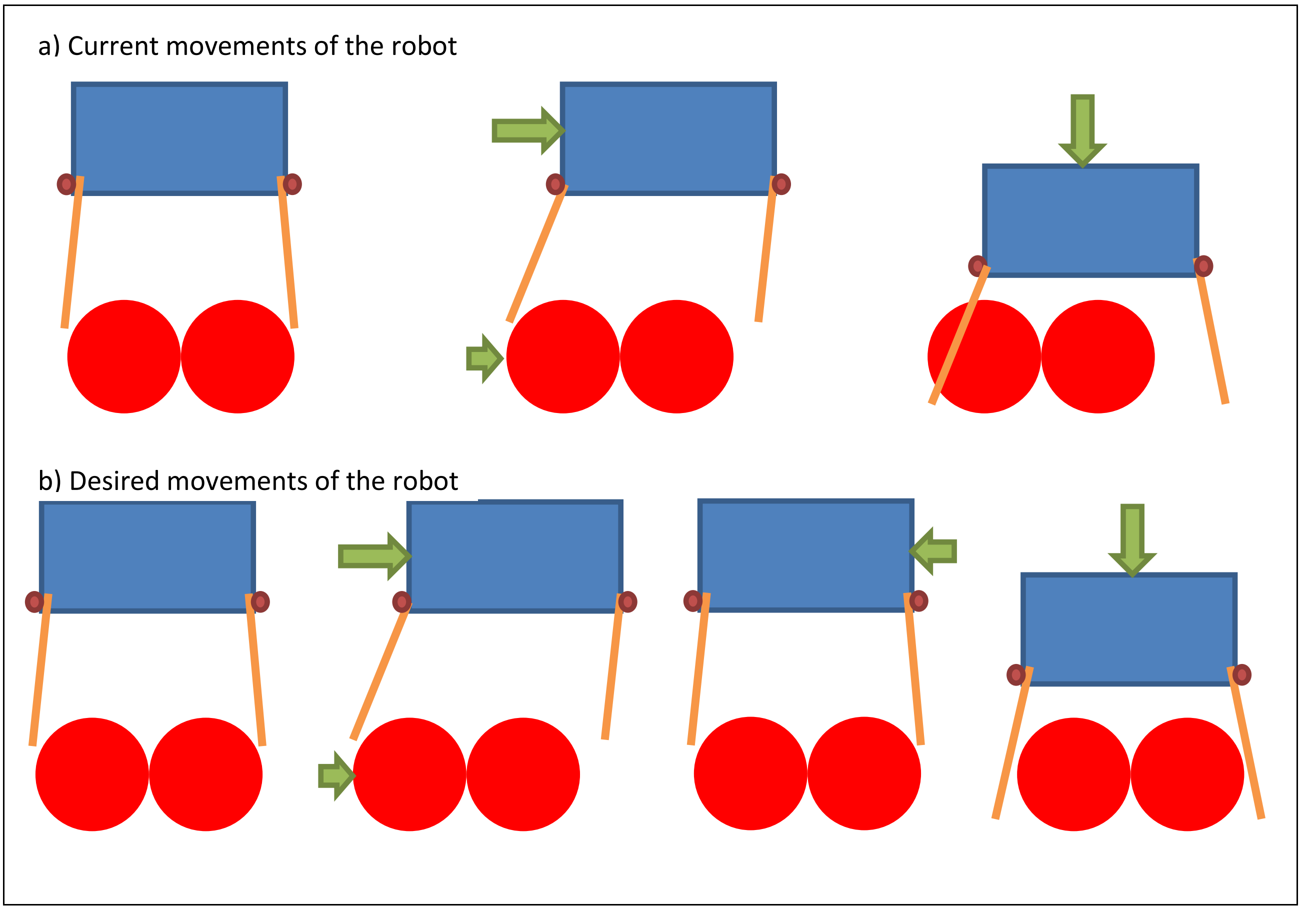


Figure 13: The current movements of the robot (a) result in splashed tomatoes. In the desired (b) situation, the robot should move the gripper slightly back, in order to recentre above the product before grasping.

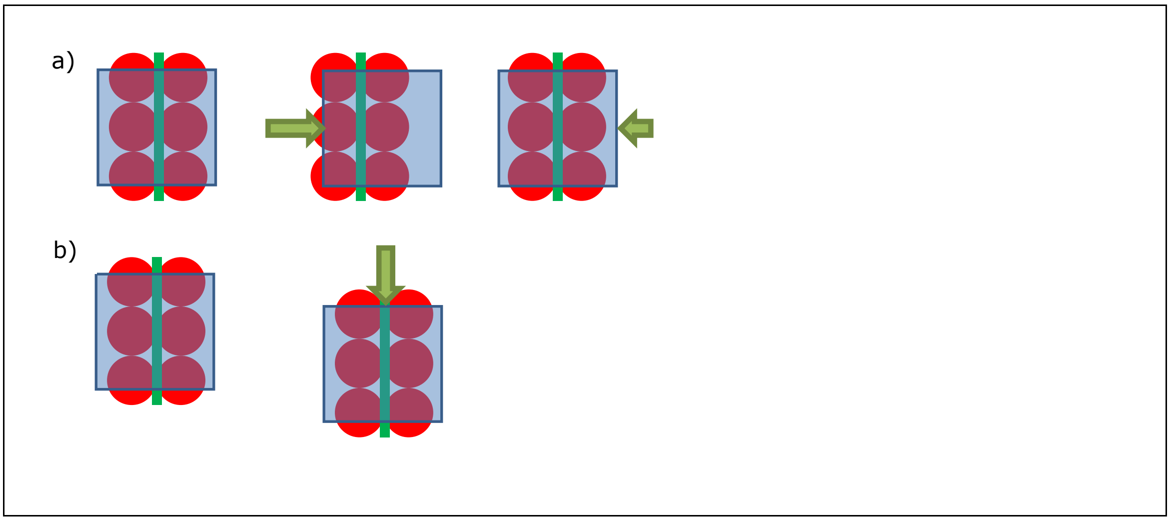


Figure 14: The movement that the gripper needs to make backwards depends on the movement during dragging in the direction perpendicular to the orientation of the truss (a). Movements in the same orientation (b) do not need to be compensated.

2.2.3 Chicken breast

Three tests with the chicken breasts where conducted: 1) a single layer of chicken breasts that were singled out, 2) a single layer of chicken breasts that were touching and, 3) multi-layered chicken breasts. The quantitative results are shown in Table 3, Table 4 and Table 5, respectively.

Moderate results were obtained for *the pick success in the bin*, 60% (24/40) for the single layer singled out, 61% (20/33) for the single layer touching and 63% (22/35) for the multi-layer condition. *The grip success in the bin* and *the transport success from bin to the web of trays* was 100% (24/24, 20/20 and 22/22) in all conditions.

Pick success: in bin.

Chicken breast 5 bins, 40 pieces Single layer: singled out		Correct	Not correct
Vision software judgement	Pick	24	6
	No Pick	0	10

Grip success: in bin.

Chicken breast 5 bins, 24 pieces Single layer: singled out		Arm/Gripper action			
		Hold	Unstable grip	Drop	Missed/not in gripper
Gripper Action	Pick	24	0	0	0

Transport success: from bin to WoT

Chicken breast 24 pieces Single layer: singled out		Arm/Gripper action		
		Hold	Unstable grip	Drop
Gripper Action	transport	24	0	0

Table 3: Chicken breast – single layer & singled out: quantitative results

Pick success: in bin.			
Chicken breast 3 bins, 33 pieces Single layer: touching		Correct	Not correct
Vision software judgement	Pick	20	3
	No Pick	0	10

Grip success: in bin.					
Chicken breast 3 bins, 33 pieces Single layer: touching		Arm/Gripper action			
		Hold	Unstable grip	Drop	Missed/not in gripper
Gripper Action	Pick	20	0	0	0

Transport success: from bin to WoT				
Chicken breast 33 pieces Single layer: touching		Arm/Gripper action		
		Hold	Unstable grip	Drop
Gripper Action	transport	20	0	0

Table 4: Chicken breast – single layer & touching: quantitative results

Pick success: in bin.			
Chicken breast 4 bins, 35 pieces Multi-layer		Correct	Not correct
Vision software judgement	Pick	22	0
	No Pick	0	13

Grip success: in bin.					
Chicken breast 4 bins, 35 pieces Multi-layer		Arm/Gripper action			
		Hold	Unstable grip	Drop	Missed/not in gripper
Gripper Action	Pick	22	0	0	0

Transport success: from bin to WoT				
Chicken breast 35 pieces Multi-layer		Arm/Gripper action		
		Hold	Unstable grip	Drop
Gripper Action	transport	22	0	0

Table 5: Chicken breast – multi-layer: quantitative results

Observations:

1. Grasping loose breast, single layer
   - Grasping of breasts with some space around goes rather well
   - In some cases, the breast rotates in the gripper. When the breast is thin, the breast slips out of the gripper
2. Grasping concatenated breasts, single layer
   - Some successes, also when grasping off centre
   - Sometimes, the fingers penetrate a neighboring breast (wrong visual analysis)
   - Would be good if the fingers of the gripper could penetrate between two breast, creating space, without damaging the breasts
3. Grasping breasts on a pile
   - When breasts are on a pile, the robot is picking too low, causing the fingers to penetrate breasts lower on the pile

Actions:

1. WUR
   a. Also check for possibilities to pick off the midline of the product, for instance to place the (improved) fingers in between two concatenated chicken breasts
   b. Check, together with Marel, why the robot is picking too low in the multi-layer scenario
   c. Fingers are sometimes in collision with neighbouring breasts. Why? Some options to look at:
      i. Maybe part of the object is not visible in the depth map. In that case, use also the colour image
      ii. Maybe we have the wrong gripper dimensions
   d. Collision detection should be done not only based on the open position, but on the complete closing trajectory.
   e. We now centre the gripper with respect to its opening position. Should be with respect to the closing position
   f. Make sure to never select a target that is not at the top of the pile
   g. Improve segmentation of concatenated and multi-layer situations
   h. Improve speed of image capture by using area of interest, see *Table*.

2. Lacquey
    a. Change the finger plate so that the opposite fingers can close more, while keeping the same opening aperture
    b. Get the two fingers at the same side closer together
    c. Create fingers that are thinner at the bottom 2cm, to fit more easily between chicken breasts
    d. See if there is a solution to have two opening positions (half and full), to have some more control on where and how to place the gripper
    e. How to prevent the breast from rotating in the gripper.

Test nr	Chicken breast number	Capture (msec)	Crate detection (msec)	Grasp position (msec)	Total (msec)
1	1	811	118	295	1224
1	2	330	108	76	514
1	3	330	126	78	534
1	4	335	134	144	613
1	5	307	122	86	515
1	6	383	126	68	577
2	1	794	58	282	1134
2	2	306	50	78	434
2	3	302	66	81	449
2	4	319	50	83	452
2	5	287	60	109	456
2	6	325	65	85	475
2	7	312	61	61	434
2	1	794	58	282	1134

Table 6: Speed test vision software for chicken breast

2.2.4 Grape vines

The grape vines could not be grasped successfully therefore no quantitative results are reported.

Observations

1. Gripper is going to the correct location and with the correct orientation
2. None of the grapes were successfully grasped. The bunches get too thin when the gripper closes. On the other hand, a bunch is quickly too wide (>130mm) when it is lying flat on the ground.

Actions

1. WUR
    a. Allow grapes bunches to slightly touch each other. Improve the segmentation of these situations
2. Lacquey
    a. Change the fingers so that they can scoop the bunches, i.e., give support from below
        i. Fully change finger, and/or
        ii. Create an addon finger tip. NB. The Marel robot can only move a little higher. Check with Marel how much long the fingers can be, in order not to collide with the rim of the crate