Developing robotic grasping of delicate objects using a sensorised soft physical twin

Catarina Pires
catarinamlcpires@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

November 2022

Abstract

As the world population keeps growing at an unprecedented rate, reports of labour shortage in the agriculture sector present an increasing threat to the global food chain. Consequently, the role of robotics in agriculture is becoming ever more important, in particular for tasks such as harvesting. However, the research process of robotic harvesting is challenging as, traditionally, extensive field tests are required. Focusing on raspberry harvesting, this project builds upon previous work where soft robotics techniques are applied to develop a sensorised physical twin that simulates the harvesting process allowing the development of a harvesting robot in the lab before deployment in the field. This provides a new approach and methodology for robotic harvesting research. In this project computer vision methods are used to develop the algorithms for fruit identification and localisation. A control method was implemented according to the raspberry's characteristics, and the physical twin was used, coupled with a search algorithm, to find a feasible harvesting force interval. The trained robotic system was then taken to the field to evaluate its performance compared with the lab experiments, obtaining a 60% success rate for no damage and 80% for minimal damage. Using this approach, the successful transfer from lab to field using a soft sensor gripper is demonstrated.
Keywords: Agricultural Robotics, Selective Harvesting, Soft Robotics, Raspberry Harvesting, Lab Training

1. Introduction

Despite the world population reaching almost ten billion people by 2050 [1] there is a growing labour shortage in the agricultural sector due to an ageing farmer population and urbanisation. In addition, the recent COVID-19 pandemic has increased losses due to the hesitance of growers to permit large groups of day labourers on their farm, limiting the amount of crops that can be harvested [2].

As a result, there has been an increasing interest in the automation of agricultural tasks. High value crops, such as raspberries, ripen heterogeneously and require selective harvesting of only the ripe fruits, maintaining the remaining fruits and plant intact to not compromise future production. Consequently, selective harvesting has proven challenging to mechanise, making it one of the most labour intensive and expensive tasks on the farm [3]. There have been some attempts at developing robotic solutions for raspberry harvesting [2], and other soft fruits [4], yet their development is limited by the challenges associated with testing and developing robotic solutions. The interactions between the gripper and the raspberry are complex to simulate, forcing researchers to turn to real world experiments, which delay development and are constrained by the limited time window in which raspberries are ripe and ready for harvest. With these issues in mind, K. Junge and J. Hughes [5] developed a sensorised physical twin of a raspberry that allows for in lab robot training. This physical twin presents similar mechanical and dynamic behaviour to a real raspberry, and provides a measure of damage that enables the researcher to evaluate the performance of the robotic picking experiments. The aim of this work is then to put this physical twin hypothesis to the test by implementing a control policy on a robotic gripper based solely on physical twin experiments, and then examine how well these results transfer onto real raspberry picking on the field.

1.1. Related Works

Agricultural robots: The agricultural environment is one of the most unpredictable and unstructured, therefore it presents many adversities that make robotisation difficult. Selective harvesting in particular presents several challenges for current robotic technology, such as variable outdoor environmental

conditions, complex tree structures, fruit clusters and occlusion, inconsistency in fruit shape and size, and fruit sensitivity to damage [6].

Soft robotics provides a novel option for automatic harvesting by leveraging the use of compliant grippers [7], which enable a wider task versatility that is not found in traditional rigid bodied robotic systems. These soft robotic systems are ideally suited for grasping [8] and manipulating delicate objects [9] with complex, dynamic shapes. Some tasks also require the right amount of force to be applied, dictating a force-based rather than position based approach to control. An example in literature that follow the characteristics described is the tendon driven soft robotic gripper for blackberry harvesting developed by Gunderman et al. [2].

The challenge of autonomously harvesting soft fruits is also being tackled by a number of companies with the intent of making this technology commercially available. For example, Agrobot [10] is a company with robots that can find and grab the stem, and Fieldwork Robotics Ltd [11] developed a raspberry picking robot which has now been commercially deployed in two locations in Portugal [12].

Fruit detection: Robotic vision in agriculture requires robustness to changes in illumination, weather conditions, image background, and object appearance. It must be able to detect the presence of individual fruits, discriminate them from the rest of the scene, and localise them in space.

In some of the earliest fruit detection studies, black and white cameras were used, taking advantage of texture and geometry as the main features for identification [13], using methods such as the Circular Hough Trasnform (CHT). However, the use of colour cameras provides the opportunity for colour based segmentation of fruit, in addition to the geometric and texture information. A wide variety of distinctly coloured fruits such as apples [14] have been the focus of many researchers. Nonetheless, colour analysis falls short when chromatic differences between the fruit and the background foliage are more subtle. In these cases spectral reflectance can constitute an effective discriminatory factor [15].

The use of calibrated stereo vision has been a greatly used solution for localisation in recent years [16]. Although, the use of time of flight sensors, infrared sensors [17] or laser sensors [18] paired with RGB cameras are also popular choices, acknowledging that depth information obtained purely visually may suffer inaccuracies.

In addition, recent work in this deep learning has shown great advancements in object detection and classification tasks [19], and has been applied to agriculture with excellent results such as the DeepFruit network [20]. Yet, the performance of any machine learning method depends heavily on the quality and quantity of training data sets, and large data sets do not currently exist for real-world agricultural settings [21]

Physical Twin: A physical twin aims to artificially replicate a system that is challenging to simulate. The concept of physical twins, or "phantoms", has been widely explored for training in the medical field. For example, L. Scimeca et al. [22] use an abdominal phantom to train a robotic arm to perform medical palpation for tumour diagnosis, and M. J. Luk et al. [23] introduce a soft robotic phantom that simulates the cervix behaviour during labour for midwifery and obstetrics training. Moreover, these twins can be sensorised to provide feedback. These approaches have been shown to have a significant impact on enabling increasingly rapid testing and development.

2. Robotic Set-up

Sensorised Gripper: The gripper consists of two 3D printed Polylactic acid (PLA) fingers with tips in silicone Dragon Skin™ 10 Fast that move horizontally through the action of a Dynamixel XM430-W210 motor. Each finger is connected to a 500g load cell, and both load cells are then connected to an Arduino Nano Every. The load cells measure the compression and pulling forces acting on the raspberry, F_{c_G} and F_{p_G} respectively. A Raspberry Pi camera, an OAK-D camera (RGB camera coupled with two monochromatic cameras for stereo vision) and a Time-of-Flight (TOF) sensor were used for fruit detection. Finally, a Jetson Nano was employed to run all the code rather than on a separate computer to simplifying the connections between all components. All other structural or mechanical components were 3D printed PLA or laser cut. The gripper was mounted on the Universal Robots™ UR5 robotic arm, and for the field tests the whole system was transported on a Panther - outdoor autonomous mobile robot by Husarion. The whole set up is shown in Figure 1.

Raspberry Physical Twin: The raspberry physical twin's consists of a Dragon Skin™ 10 Fast silicone "soft fruit" and a 3D printed PLA "receptacle". A fluidic sensor which monitors the pressure changes induced by the deformation of the fruit was used to measure the compression force P both on and off the receptacle. A Neodymium magnet embedded in both the fruit and the receptacle was used to simulate the pulling force necessary for harvest, allowing for maximum pulling force tuning. The pulling force F_{p_R} is measured by a 1kg load cell connected to the receptacle by a 3D printed Thermoplastic Polyurethane (TPU) part, aiming to repli-

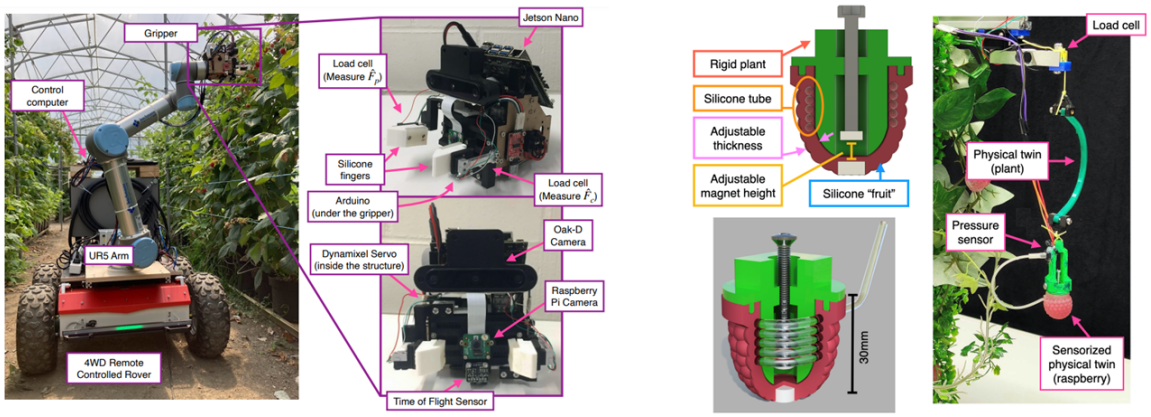

Figure 1: Hardware: [Right] Components of the raspberry harvesting robot, [Left] Physical twin: (Left) 3D render of the raspberry twin [5], (Right) real physical twin set up

cate the stiffness of the raspberry plant. Finally, for computer vision purposes, fake leaves and plants, and other fake silicone non-sensorised raspberries were added to the set up. Figure 1 shows the physical twin set-up.

3. Software and Implementation

3.1. Computer Vision

Fruit detection: Raspberries are approximately round bright red fruits, which stand out from their surroundings in nature. Thus, colour-based segmentation coupled with shape detection were considered appropriate methods for this application. Once a frame is captured by the RGB camera, it is converted into the HSV space and a copy of it is converted into grey scale. Due to raspberries' characteristic colour, by applying a threshold to the Hue channel it is possible to find which pixels of the image belong to raspberries. Therefore, a binary mask can be created, which consists of an array of boolean values where the pixels that are not part of raspberries in the image will be attributed "False" and the ones that do will be attributed "True". Combining this mask with the grey scale image, results in an image with all the raspberries in grey scale over black background, keeping both texture and shape information. In addition, a Gaussian blur is applied to the resulting image to filter out any noise. Lastly, the CHT was applied to the final image to find the location, in pixels, of the raspberries in the image. The pipeline of fruit identification is illustrated by Figure 2.

Fruit localisation and Visual Guidance: The CHT not only outputs the pixel coordinates of the detected raspberry, but also the radius of the corresponding circumference. This can be used to estimate distance since the closer the raspberry, the bigger the radius.

The algorithm begins by choosing a raspberry to be harvested according to which is closest to the centre of the frame. Next, the arm moves horizontally and vertically until the chosen raspberry's position is within a chosen proximity radius from the centre of the frame. The arm moves at a constant speed of $0.01\,\mathrm{m\,s^{-1}}$ and the proximity radius chosen was 5 pixels. After alignment, the arm moves forward at constant speed, registering the calculated radii of the raspberry as it moves. Once the threshold of 40 pixels is achieved the arm stops and aligns the gripper's fingers with the raspberry by moving the gripper up 6.3cm. The end-effector moves towards the raspberry until it is well positioned between the fingers. This is sensed by either the TOF sensor or by the Raspberry Pi camera positioned between the fingers. The frames obtained from the Raspberry Pi camera are processed in a similar way to what was previously described, but without applying the CHT. Instead, the mean pixel value of a black and white image was used as the indicator of when the raspberry is in place. This image is derived from the binary mask, where pixel entries that were "True" have the value 255 (white) and the ones that were "False" have the value 0 (black). When the raspberry is between the gripper's fingers the camera can only see the raspberry, therefore the mean of the black and white image should be very close or equal to 255. The further the raspberry is the more black pixels will start showing up, consequently decreasing the mean value of the image.

3.2. Controller

Gripper control: The harvesting task was divided into two phases according to how the stiffness of the raspberry varies: a) grasping and pulling, and b) holding. The detachment of the fruit from the receptacle signals the transition from one phase to the other. The change in stiffness implies that the robot must be extra gentle with the harvested raspberry as soon as it detaches from the plant. This discontinuity coupled with non-linearities derived from the surface interactions between the fruit and the gripper, make the modelling of the system, and

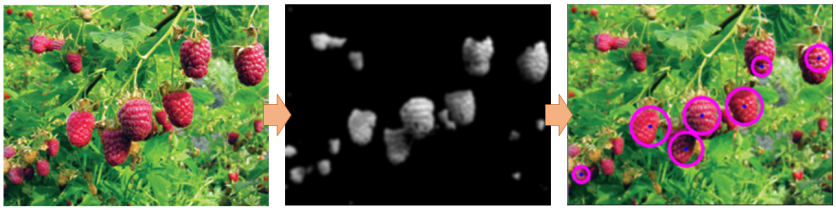

Figure 2: Illustration of the raspberry detection pipeline

therefore controller design, challenging. To accommodate the two states of the raspberry, two PID controllers were designed to regulate the compression force: controller 1 would be active during phase a), and once phase b) was detected, the controller would be switched into controller 2. Since the switching condition is irreversible, the transition between controllers does not cause any instability issues. Both controllers were tuned according to the Ziegler-Nichols method and force control was implemented, as opposed to position control. The compression force applied by the gripper is compared with a reference to obtain the force error, which constitutes the input of the controller, that sends the desired angular velocity to the motor, moving the fingers accordingly.

Off Receptacle Detection: The detachment moment is when the change in stiffness occurs and the raspberry has the highest probability of getting damaged. Once the raspberry has been grasped with the desired force, the arm begins to move down, pulling it off the plant. As the arm moves down the vertical force increases steadily, until it suddenly drops during the moment of detachment. This was monitored by keeping a record of $F_{PG\,i-1} - F_{PG\,i}$, i.e. the symmetric of the gradient of the pulling force measured by the gripper's load cell. Moreover, with every new sample that was added to this array, the standard deviation was calculated. If the value of $F_{PG\,i-1} - F_{PG\,i}$ was bigger than a certain threshold, in this case 20g, and bigger than b times the standard deviation, then it was assumed that detachment had occurred. In this implementation $b = 4$.

Arm control: The robotic arm was controlled by using the *ur-rtde* library [24]. Once the raspberry has been approached and grasped with the desired grasping force, the arm moves down at a constant speed, stopping as soon as the detection "off receptacle" is signalled.

3.3. Lab-based Training

Figure 3 proposes the high level optimisation flow of how the sensorised physical twin can be used together with a search algorithm ("Control Update Policy") to optimise the harvesting process.

After designing the closed loop controller for the gripper, the challenge remains in determining the

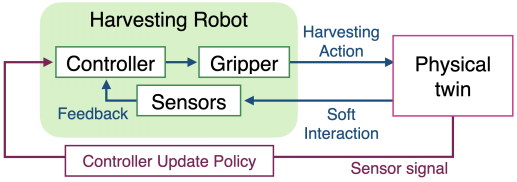

Figure 3: Block diagram of the physical twin assisted optimisation in the lab

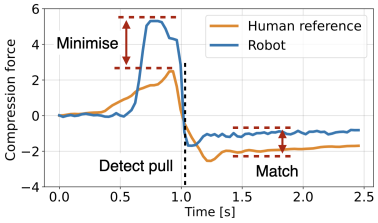

Figure 4: Goal of the optimisation of F_{sp1} and F_{sp2}

force setpoint for each of these controllers, F_{sp1} and F_{sp2}. Thus, a gradient descent inspired heuristic was employed, where the objective function aimed to minimise the maximum compression force when comparing the robot's performance with the human's, as illustrated by Figure 4. The physical twin served as the common medium for this comparison. Note that the training is performed using real equipment, therefore an algorithm that leverages more exploitation, rather than exploration is preferred, taking into consideration the wear and tear of the material and the robotic joints. The algorithm was run simulating multiple raspberry ripeness levels to obtain a feasible harvesting force interval. This was achieved by tuning the magnetic screw on the twin, iterating for "Low", "Medium", and "High" pulling forces.

It was observed, that due to the silicone's restoration properties the pressure sensor values showed some variance between picks, when the fruit was off the receptacle. Since the harvested raspberry should be gently held, but firmly enough not to fall off the gripper, it was decided to set $F_{sp2} = 40$g from the beginning and maintain it constant for all iterations. With F_{sp1} initialised as a high value, the raspberry is harvested, using controller 1 for the grasping and pulling, followed by controller 2

and F_{sp2} as reference once the "off receptacle" has been detected. During the harvesting process the measured P, F_{p_R}, F_{p_G}, and F_{c_G} are stored. Aiming to increase accuracy, each iteration of the algorithm consists of five pickings. This means that the twin is harvested and put back in place by the robot five times before F_{sp1} is updated. At the end of each trial the robot verifies if the raspberry was successfully harvested or if it slipped, and registers the trial as either successful or unsuccessful.

After the five trials, the maximum P_{robot_i} values are averaged and compared with the human reference. It is worth noting that the human reference underwent a similar data processing, i.e. five harvests of the twin were performed by a human and averaged for each raspberry ripeness. The value of F_{sp1} is then updated according to equation 1.

$$F_{sp1_{i+1}} = F_{sp1_i} - \gamma(\bar{P}_{robot_i} - \bar{P}_{human}) \quad (1)$$

Where $\bar{P}_{robot_i}$ denotes the mean value of the pressure sensor measurements for iteration i of the robot picking, $\bar{P}_{human}$ is the mean value of the pressure sensor measurements for the human harvesting reference, and γ acts as a learning rate. However, this update rule is only applied if the iteration is considered a success, meaning that all five trials were able to successfully remove the raspberry from the receptacle. Otherwise the iteration is considered a failure and F_{sp1} is updated according to equation 2. Where F_{sp1_j} denotes the last registered successful F_{sp1} force, reached in iteration j, n_{fails_i} is the number of failed trials in iteration i, and n_{trials_i} is the number of trials in iteration i. Knowing that F_{sp1_j} succeeded and F_{sp1_i} did not, this condition aims to find the limit force between failure and success.

$$F_{sp1_{i+1}} = F_{sp1_i} + (F_{sp1_j} - F_{sp1_i})\frac{n_{fails_i}}{n_{trials_i}} \quad (2)$$

Finally, if the robot performs better than the human, it is desired to encourage the algorithm to search further for the limit force. Therefore the update function is equation 3, where ΔF is a random number between 1 and 10.

$$F_{sp1_{i+1}} = F_{sp1_i} - \Delta F \quad (3)$$

Slip detection: Attempting to simulate touch, a slip detection algorithm was implemented making use of the Raspberry Pi camera similarly to what is used for visual guidance. A binary mask is applied to the frame collected by the camera, and the average value of the pixels is calculated. Since the raspberry is right in front of the camera, this binary mask should be mostly white and the average value should remain fairly constant during pull, granted the raspberry stays in place. But, if it starts to slip, more black pixels will appear in the binary image. By comparing how the average pixel value changes from frame to frame, the robot is able to identify if the raspberry might be about to slip its grasp. Once slip is detected the robot stops in place, increases the F_{sp1}, and then retries to pull the raspberry.

4. Results

4.1. Lab Tests

Computer Vision: Frames are 8-bit files, and therefore hue varies between 0 and 255 for each pixel. Consequently, for raspberry detection the hue values considered were from 0 to 10, and from 150 to 255. Regarding the CHT, there were also some parameters that required to be tuned. According to OpenCV's documentation.

- `min_dist`: Minimum distance between detected centers. For this application it was defined as 30.
- `param_1`: Upper threshold for the internal Canny edge detector. For this application it was defined as 100.
- `param_2`: Threshold for centre detection (it was observed that the function seemed to be more "strict" for high *param_2* values, not recognising some of the circles on frame, and less "strict" for low values giving more noisy results). For this application it was defined as 17.
- `min_radius`: Minimum radius to be detected. For this application it was defined as 5.
- `max_radius`: Maximum radius to be detected. For this application it was defined as 50.

The minimum distance between detected circles was chosen in a way to avoid the same raspberry being detected twice, but in different places. The maximum and minimum radii related parameters were defined such that the algorithm would detect raspberries at a range of distances, yet being low and high enough, respectively, so as to filter out noisy detection.

Table 1, shows the performance of this algorithm in the lab. These results were obtained using the physical twin set up with 4 fake raspberries, and by analysing a number of 13s videos. The field 'Frequency of detection' denotes the percentage of frames in which n amount of raspberries were detected.

The detection of 0, 1 and 2 raspberries happened mostly during the first frames, right after the camera was turned on and was still unfocused and not and dark. Once light, contrast, and focus had stabilised, the algorithm was able to detect all the raspberries most of the time, having more difficulty with the most occluded raspberry.

Detected raspberries	Frequency of detection
0	1%
1	6.33%
2	4.33%
3	20%
4	68.33%

Table 1: Fruit detection in the lab

Controller: The gain values obtained for each of the controllers are shown in Table 2.

Controllers	K_P	K_I	K_D
Controller 1	0.05	$8e^{-6}$	0
Controller 2	0.3	$1e^{-4}$	0

Table 2: Gains for controllers 1 and 2

Notice that the derivative gain ended up being equal to zero, thus the controllers were, in fact, PI controllers. The response of the controllers while the robot harvests the raspberry twin is shown in Figure 5. The top left plot displays in yellow F_{sp1} and F_{sp2}, and the switch moment between controllers. In blue it is shown the gripper's response, i.e. F_{c_G}. The top right plot shows the pulling force, F_{p_G}, applied to the raspberry during the harvest and the moment when this force suddenly decreases signalling the transition from phase a) to phase b). Finally, the bottom plot shows how the distance between the gripper's fingers varies during harvest.

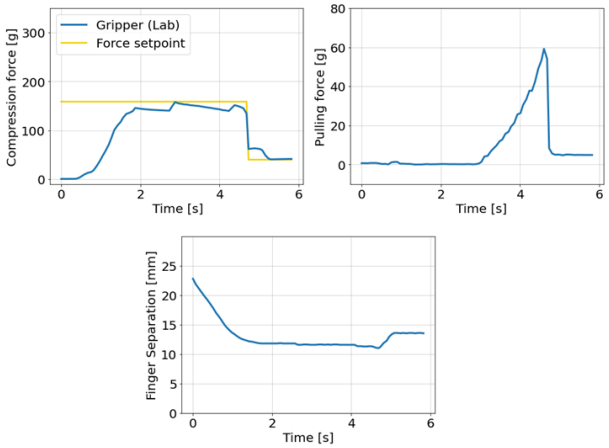

Figure 5: Harvesting of the twin in the lab: system's response to force setpoints (top left), pulling force measured by the gripper's load cell over time (top right), distance between the gripper's fingers over time (bottom)

In Figure 5 it is possible to see the fingers made contact with the raspberry at time 0.5s when F_{c_G} begins to increase. Shortly before the 2s mark the raspberry has been grasped, as the distance between the fingers stabilises. Then, $F_{c_G} = F_{sp1}$ occurs at around the 3s mark, after which the arm begins to move down, shown by the increase in F_{p_G}. The pulling force maximum is reached close to the 4.5s mark with the detachment moment occurring right after. At this point the controllers are switched, and the gripper responds accordingly by slightly releasing the raspberry to avoid damage (bottom plot), and $F_{c_G} = F_{sp2}$ occurs shortly after the 5s mark.

Lab-based Training: The human references employed are shown in Figure 6. The top plot shows the values obtained from the pressure sensor (arbitrary units), P_{human}, and the bottom plot presents the pulling force measured by the load cell on the physical twin set up.

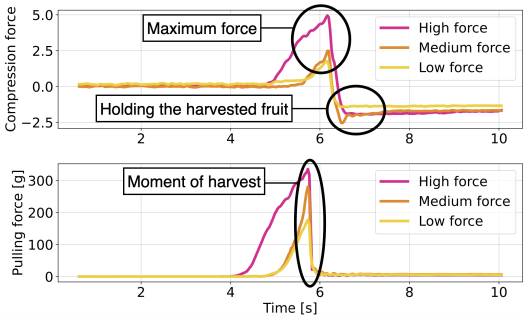

Figure 6: Measurements obtained from human harvests of the sensorised physical twin for three different ripeness levels

The maximum values obtained for both the compression force and the pulling force are presented in Table 3. The pressure sensor maximum values were used as baseline for the comparison with the robot pickings. The search algorithm was run for 15 iterations for each ripeness level with the γ value presented in equation 1 set to 10, and the number of iterations used as the algorithm's stopping criterion. Moreover, the arm was programmed to put the silicone fruit back in place after every picking trial, to allow for autonomous training.

Force level	Compression force	Pulling force [g]
Low	1.75	177.34
Medium	2.49	280.48
High	4.95	335.44

Table 3: Maximum pressure sensor and load cell values for the human references

The training results obtained for the different ripeness levels are shown in Figure 8. The top plots show how F_{sp1} evolved as iterations went on and the

plots below show the relative error between $\bar{P}_{human}$ and $\bar{P}_{roboti}$, calculated according to equation 4, for each iteration. F_{sp1}, was initiated equal to 350g for all three training experiments.

$$\frac{\bar{P}_{roboti} - \bar{P}_{human}}{\bar{P}_{human}} \tag{4}$$

Black dots represent iterations that were considered failures. For visualisation purposes the relative error of these iterations was represented as equal to the previous successful iteration. Figure 7 shows a comparisons between the human reference, $\bar{P}_{human}$, and the robot's pressure sensor values for the first defined F_{sp1} (left), and the "optimal" F_{sp1} (right) when picking the twin with the lowst pulling force. In these plots it is possible to observe how the robot starts by harvesting the raspberry way too harshly with the initially defined force, but by the end of the iterations the reference value is well matched by the robot.

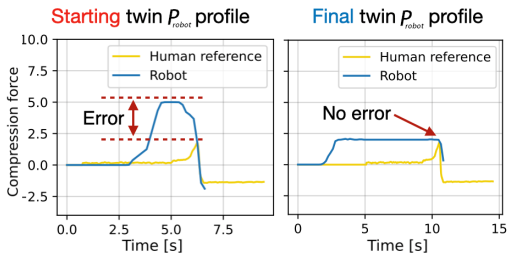

Figure 7: Pressure sensor measurements comparison between the start and end of training

All three F_{sp1} plots show a clear downward tendency as training goes on. The steepest descent is displayed by the lowest pulling force raspberry, and the most gradual descent is featured by the highest pulling force raspberry. This was not surprising as all training experiments were initialised in the same conditions, i.e. $F_{sp1} = 350$g, and thus it is expected that the grasping force needed to pick the high pulling force raspberry would be closer to the initial value, then the grasping force needed to pick the low pulling force raspberry.

On the same note, it is possible to observe that the relative error plot of the low pulling force raspberry also shows a clear decreasing trend. On the other hand, the medium pulling force raspberry's relative error ended up oscillating the most out of the three and the high pulling force raspberry's relative error barely changed at all compared to the other two. This is most probably derived from variance in the pressure measurements from one trial to another due to material restoration properties. Since pressure sensor measurements are relatively small, oscillations translate into bigger relative errors. Moreover, the monotony of the high pulling force raspberry's relative error plot comes from the fact that the grasping force needed to pick it ends up being relatively close to the initial value.

Force level	"Optimal" F_{sp1} $[g]$	Absolute Error	Relative Error [%]
Low	158.64	0.50	28.73
Medium	216.42	1.05	42.09
High	293.43	0.68	13.83

Table 4: Lab-based training results

The experiments with the low pulling force raspberry were successful most of the time with a 20% failure rate (3 out of 15), and the medium and high pulling force raspberry experiments failed equally often presenting a 33.3% failure rate (5 out of 15). The "optimal" values for F_{sp1} and the relative error, as well as the absolute error, $\bar{P}_{robot_{optimal}} - \bar{P}_{human}$, for each ripeness level are presented in Table 4. The robot was able to harvest the low pulling force raspberry with the least absolute error, but the harvest of the high pulling force raspberry showed the least relative error. The harvest of the medium pulling force raspberry was the one that the robot had the most difficulty to learn.

4.2. **Field Tests**

Computer Vision: In the field, the vision related methods struggled the most to adapt. Problems occurred when bright spots showed up on frame, caused by direct sunlight. To overcome this, value (brightness) thresholding was added to the pipeline, by cutting off pixels with value above 240. The CHT parameters were kept the same.

Detected raspberries	Frequency of detection	Chosen raspberry detected	Frequency of detection
1	52.60%	Yes	87.64%
2	47.40%	No	12.36%

Table 5: Raspberry detection in the field

Table 5, show the performance of this algorithm in the field. These results were obtained by analysing a number of 4s camera alignment videos where 2 raspberries were clearly visible. The videos were taken of the same raspberries, but starting in slightly different positions. On the left side of Table 5 it is shown how often n amount of raspberries were detected, and on the right how often the chosen raspberry is detected, regardless of other raspberries being detected as well. The times when 1 or 2 raspberries were detected in frame, overall, was very close to 50/50, nonetheless the raspberry that was meant to be harvested was detected almost 90% of the time.

Regarding the approach algorithm, the use of the Raspberry Pi camera to detect when the raspberry

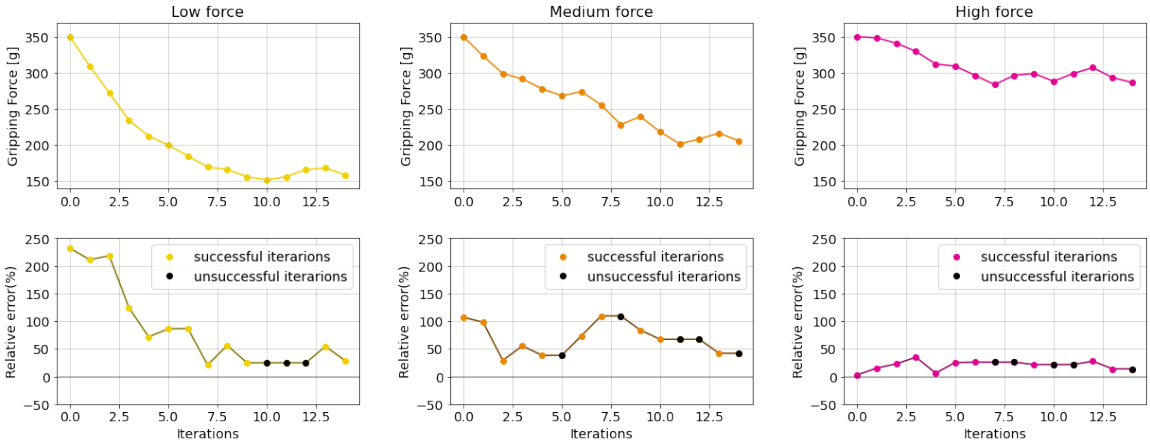

Figure 8: Lab-based training results for different simulated ripenesses

was close enough to the fingers proved to be more reliable than the TOF.

Controller: The robot was tested on over 60 raspberries over two days. Here, a representative set of 25 harvested raspberries is analysed. Out of these, 22 were removed off the receptacle, of which 2 were highly damaged (extremely squashed, or were dropped) and 5 got some damage (slightly squashed, small "rips" in the fruit). Failures related to control were mostly caused by bad positioning of the fingers, failure in the slip detection, or failure in detecting the detachment moment. Table 6 presents the field harvesting data, showing a 60% success rate for no damage and 80% for minimal damage.

Harvest number	Harvest off plant	Damage of fruit	Attempts	Controller strength
1	Fail	-	1	Low
2	Success	Some damage	1	Low
3	Success	Good	1	Low
4	Success	Good	1	Low
5	Success	Good	1	Low
6	Success	damaged	1	Low
7	Success	Good	1	Low
8	Fail	-	1	Low
9	Success	Some damage	1	Low
10	Fail	-	2	Medium
11	Success	Good	1	Low
12	Success	Good	2	Medium
13	Success	Some damage	1	Low
14	Success	damaged	1	Low
15	Success	Good	1	Low
16	Success	Good	3	high
17	Success	Good	1	Low
18	Success	Good	1	Low
19	Success	Good	1	Low
20	Success	Good	1	Low
21	Success	Some damage	1	Low
22	Success	Good	1	Low
23	Success	Good	2	Medium
24	Success	Good	3	High
25	Success	Some damage	3	High

Table 6: Harvesting data of raspberries in the field

Due to the conical shape of the raspberries, the finger positioning sometimes lead to slipping of the raspberry, resulting in failure. Positioning the fingers closer to the top of the receptacle achieved considerably better results. It was also observed that the pulling direction played an important role in the success of the pick. The robot was programmed assuming that the raspberries would be hanging vertically, moving down to produce the pulling action. However, some raspberries hang in awkward angles., in which cases some "rips" occurred as the pulling force was not parallel with the raspberry's orientation.

The slip detection algorithm ended up struggling in the field. At time it was overly sensitive, leading to damaged raspberries, while others not quick enough to react (harvests 1, 8 and 10 in Table 6). Even though slip detection was successful a handful of times, as shown in harvests 12, 16, 22, 23 and 25 (Table 6), it was set aside for the remaining of the trials to properly test the controllers. Slip detection was failing due to shifting light directions resulting from movement of the raspberries during the pulling process.

When the detachment moment is not detected there is no switch between the controllers, ultimately resulting in the squashing of the raspberry. The separation of a real raspberry from its receptacle is more gradual than a separation between two magnets, which most probably lead to the off receptacle detection failures, especially for very ripe raspberries (6 and 14 in Table 6).

The F_{c_G} and F_{p_G} measurements were used to compare the in-field performance of the controller trained in the lab, shown in Figure 9. The top plots display F_{c_G}, F_{sp1} and F_{sp2}, and the bottom plots show Fp_G. There is a match of key characteristics between the real fruit and physical twin, for both F_{c_G} and F_{p_G}. Moreover, there is a slower decrease rate of the compression force after the switching of the controllers, likely caused by the different Young modulus between the physical twin and the

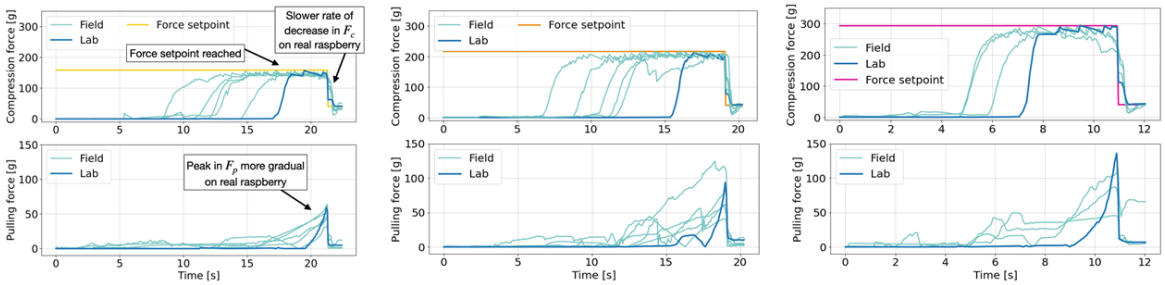

Figure 9: Lab vs field raspberry harvesting data: Low force (left), Medium force (middle), High force (right)

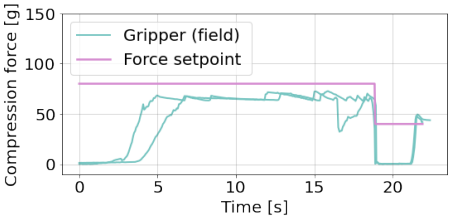

Figure 10: Harvesting with force lower than the optimal range

real fruit. Also, there is a more gradual increase of F_{p_G}, likely resulting from the unmodeled complex force-displacement characteristics of the plant, and a bigger variation of the pulling force peak for different raspberries.

Additionally, experiments done using $F_{sp1} = 80$g proved that using the forces within the found interval produced better results than forces outside of that range, Figure 10. It shows that after "detachment" F_{sp1} goes to zero before reaching F_{sp2}. This means that the gripper slipped from the raspberry and, in an attempt to reach F_{sp2}, the fingers kept moving closer until they met and reached the F_{sp2} by pressing on each other.

5. Conclusions

In this project, an autonomous raspberry harvesting robot was developed with the help of a sensorised physical twin. A computer vision pipeline that detects and localises raspberries, and a control method suitable to the raspberry's characteristics were implemented. The computer vision algorithm in the lab's set up was able to identify 75% or more of the raspberries on frame 88.33% of the time. In the filed, it was necessary to add Value thresholding, after which it was able to trace the chosen raspberry during movement 87.64% of the time. Two PID controllers were employed to regulate the compression force acting on the raspberry. The physical twin was used as a common medium to compare between human and robot harvests for lab training. In the field, the controller translated generally well with a 60% success rate for no damage and 80% for minimal damage. An efficient lab to field transfer of a harvesting robot by using a sensorised physical twin was shown, concluding that a robot trained in the lab will function well in the field as long as the scenario is similar enough to the lab.

5.1. Future Work

Making the system more robust to occlusions is something that should be taken into consideration both in fruit detection and how the fruit is approached. Slip detection, as well, still has a lot of potential. Not only in how slip is detected, but also in the re-grasping of the raspberry and the update rules.Finally, the introduction of more shape, size and colour diversity in the physical twin set-up is also worth exploring.

Acknowledgements

This project was done in collaboration with the CREATE Lab at EPFL under a grant from the the Swiss European Mobility Program. I would like to thank PhD student Kai Junge for the technical support provided during the development of this work.

References

[1] Department of Economic and United Nations Social Affairs, Population Division. World population prospects 2022. https://population.un.org/wpp/. Accessed 22-09-2022.

[2] Anthony L Gunderman, Jeremy Collins, Andrea Myer, Renee Threlfall, and Yue Chen. Tendon-driven soft robotic gripper for berry harvesting. *arXiv preprint arXiv:2103.04270*, 2021.

[3] Gert Kootstra, Xin Wang, Pieter M Blok, Jochen Hemming, and Eldert Van Henten. Selective harvesting robotics: current research, trends, and future directions. *Current Robotics Reports*, 2(1):95–104, 2021.

[4] Eduardo Navas, Roemi Fernández, Delia Sepúlveda, Manuel Armada, and Pablo Gonzalez-de Santos. Soft grippers for auto-

matic crop harvesting: A review. *Sensors*, 21(8):2689, 2021.

[5] Kai Junge and Josie Hughes. Soft sensorized physical twin for harvesting raspberries. In *2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)*, pages 601–606. IEEE, 2022.

[6] Abhisesh Silwal, Joseph R Davidson, Manoj Karkee, Changki Mo, Qin Zhang, and Karen Lewis. Design, integration, and field evaluation of a robotic apple harvester. *Journal of Field Robotics*, 34(6):1140–1159, 2017.

[7] Daniela Rus and Michael T Tolley. Design, fabrication and control of soft robots. *Nature*, 521(7553):467–475, 2015.

[8] Taimoor Hassan, Mariangela Manti, Giovanni Passetti, Nicolò d'Elia, Matteo Cianchetti, and Cecilia Laschi. Design and development of a bio-inspired, under-actuated soft gripper. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3619–3622. IEEE, 2015.

[9] Jianshu Zhou, Shu Chen, and Zheng Wang. A soft-robotic gripper with enhanced object adaptation and grasping reliability. *IEEE Robotics and automation letters*, 2(4):2287–2293, 2017.

[10] Agrobot. Robotic harvesters. https://www.agrobot.com/e-series. Accessed: 25-09-2022.

[11] Fieldwork Robotics Ltd. https://fieldworkrobotics.com/, Sep 2021. Accessed: 25-09-2022.

[12] Tony Quested. Robocrop company plucks double success in Portugal. *Business Weekly*, Apr 2022. https://www.businessweekly.co.uk/news/agriculture/robocrop-company-plucks-double-success-portugal, Accessed: 25-09-2022.

[13] Martha Cardenas-Weber, Amots Hetzroni, and Gaines E Miles. Machine vision to locate melons and guide robotic harvesting. *Paper-American Society of Agricultural Engineers (USA)*, 1991.

[14] Abhisesh Silwal, Aleana Gongal, and Manoj Karkee. Apple identification in field environment with over the row machine vision system. *Agricultural Engineering International: CIGR Journal*, 16(4):66–75, 2014.

[15] Ting Yuan, Wei Li, Qingchun Feng, and Junxiong Zhang. Spectral imaging for greenhouse cucumber fruit detection based on binocular stereovision. In *2010 Pittsburgh, Pennsylvania, June 20-June 23, 2010*, page 1. American Society of Agricultural and Biological Engineers, 2010.

[16] Huanyu Jiang, Yongshi Peng, and Yibin Ying. Measurement of 3-d locations of ripe tomato by binocular stereo vision for tomato harvesting. In *2008 Providence, Rhode Island, June 29–July 2, 2008*, page 1. American Society of Agricultural and Biological Engineers, 2008.

[17] Ya Xiong, Cheng Peng, Lars Grimstad, Pål Johan From, and Volkan Isler. Development and field evaluation of a strawberry harvesting robot with a cable-driven gripper. *Computers and electronics in agriculture*, 157:392–402, 2019.

[18] Qingchun Feng, Wei Zou, Pengfei Fan, Chunfeng Zhang, and Xiu Wang. Design and test of robotic harvesting system for cherry tomato. *International Journal of Agricultural and Biological Engineering*, 11(1):96–100, 2018.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[20] Inkyu Sa, Zongyuan Ge, Feras Dayoub, Ben Upcroft, Tristan Perez, and Chris McCool. Deepfruits: A fruit detection system using deep neural networks. *sensors*, 16(8):1222, 2016.

[21] Raymond Kirk, Grzegorz Cielniak, and Michael Mangan. L* a* b* fruits: A rapid and robust outdoor fruit detection system combining bio-inspired features with one-stage deep learning networks. *Sensors*, 20(1):275, 2020.

[22] Luca Scimeca, Josie Hughes, Perla Maiolino, Liang He, Thrishantha Nanayakkara, and Fumiya Iida. Action augmentation of tactile perception for soft-body palpation. *Soft robotics*, 9(2):280–292, 2022.

[23] Michelle Jennifer Luk, Derek Lobb, and James Andrew Smith. A dynamic compliance cervix phantom robot for latent labor simulation. *Soft robotics*, 5(3):330–338, 2018.

[24] Anders Prier Lindvig. ur-rtde. https://sdurobotics.gitlab.io/ur_rtde/. Accessed: 01-03-2022.