AUTOFHS: An Automatic Fruit Harvesting System

O/E: Human's Protection System/Agriculture

Luigi Capogrosso
University of Verona
luigi.capogrosso@studenti.univr.it

Abstract

In this work, I, firstly, provide an overview of the worldwide development and current status of precision-agriculture technologies. Some of the scientist's contributions are flying robots, forester robots, and, in the majority of cases, mobile robots. Secondly, I talk about the physical Human-Robot Interaction (HRI) and cooperation. In detail, I will focus on safety, an important field of HRI because robots can perform powerful movements that can cause hazards to humans surrounding them.

So, this paper proposes the development of an automatic fruit harvesting system by combining a low-cost robotic arm, a low-cost stereovision camera placed in the gripper tool, and a cubic metal gate for human protection. In particular, the robotic arm works on an agricultural collection wagon drawn by a tractor.

***Keywords**: Human Protection System, Agriculture, Robotic Arm, Fruit Harvesting.*

Figure 1. Graphic representation of the proposed solution.

1. Introduction

The agriculture industry is demanding technological solutions focused on automating agriculture tasks in order to increase production and benefits while reducing time and costs. With the population expected to reach 9 billion, agricultural production must increase if it is to meet the increasing demands for food. In particular, these technological solutions are mostly based on the application of *Sensor-Based Technologies*. A comprehensive description can be found in [17] where the most recent research focused on solving agriculture and forestry tasks by sensors is summarized. Nowadays, scientists have the goal of creating *Automatic Farms*, where all of the work will be done by machines.

Another modern and innovative technology to take care of these challenges is *Robotics*, one of the five technology trends that are driving innovations in the modern Fourth Industrial Revolution [11]. Robotics can play an important role in increasing the efficiency of agriculture productivity given limited land, water, and labor resources. The number of agricultural robots, *Agrobots*, is increasing each year. The jobs they can do are also increasing with new technology in hardware and software. Robots are milking cows, shearing sheep, picking fruit, weeding, spraying, and cultivating. The new robots are getting smaller and smarter. In particular, agricultural robots can be classified into the following several groups:

- Harvesting or picking;
- Planting;
- Weeding;
- Pest control;
- Maintenance.

The rest of the paper is organized as follows: In Subsection 1.1 a deeper state of the art is given. In Sections 2, 3, and 4 the system is detailed. Finally, in Section 5, conclusions are drawn.

1.1. A Deeper State of the Art

Robotics can play an important role in increasing the efficiency of agriculture productivity given limited land, water, and labor resources. They can be employed in various agricultural environments including large area cultivation, orchards, horticulture, nurseries, etc. Many experimental models have been built, and a few have been produced commercially. In [20] a significant amount of work is detailed regarding technology in agricultural area.

Some intelligent devices are being used in grading and sorting operations. The mostly two used installations are in California, each has an operating capacity of approximately 480 fruit per minute. One is used in a lemon packinghouse, and the other is an orange packinghouse.

In Japan, Kondo and Kawamura [10] have developed one of the first famous robots for harvesting tree fruit. They use a video camera mounted near the hand of the robot to guide it to the fruit. The system was mounted on a mobile battery-powered platform.

Numerous methods exist for manipulating robotic arms position, such as speech and gesture-based controller [19] or sensor-based interfaces [12], to name a few. In the research work published in [5] an algorithm for real-time gesture recognition is used to control the robot. The robot recognizes a particular set of commands which translates to doing some specific movements.

Automation of agriculture tasks has improved all phases of the industrial process, from the pre-harvest to the harvest and post-harvest stages.

For example, in the pre-harvest stage, the application of biochemicals in orchards has been automated with the aim of controlling weeds [18] and improving pesticide applications [9].

In the case of the harvesting stage, the already proposed systems [6] require an estimate of the position and size of the fruits in the trees [4] and other parameters such as its ripeness stage [15]. These estimates can be performed by using a stereo vision system [16].

Finally, in the case of the post-harvesting stages, the most important tasks are related with the estimate of fruit production [7] and quality [14] by detecting skin defects [3] or by validating fruit variety [8].

2. Problem Statement

The main obstacle to this kind of system presented in Section 1 and Subsection 1.1 is that farms are a part of nature and nature is not uniform. It is not like the robots that work in factories building cars. Factories are built around the job at hand, whereas, farms are not. Robots on farms have to operate in harmony with nature. For example, robots in factories don't have to deal with uneven terrain or changing conditions.

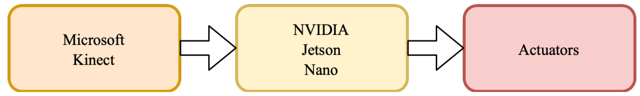

Figure 2. The block diagram of the robotic arm.

Other problems presented in [5] and [19] are the methods that are used to manipulate the robot arm, in particular, we have the speech-based and gesture-based control. However, these cannot be used for complex manipulation in agriculture where, for example, the noise level is high.

Regarding physical HRI and safety, numerous research studies have been done so far. Safety issues in human-robot collaborative areas are important both for human operators and equipment since robots are strong enough to cause harm to operators. Some EU-funded research projects have investigated HRI and safety issues, especially those related to physical interaction and lightweight industrial and non-industrial robots. An example of the investigation into physical HRI is the Saphari project (`http://www.saphari.eu`). In this project, undesired collision and unintended contacts are prevented by collision avoidance algorithms. The team has, therefore, applied only software prevention which, however, we know is not enough.

In addition, the resulting decrease in rural population has meant that the number of farms has also decreased. Thus, nowadays, we have an increasing concentration of land in the hands of a few owners. These, in particular, year after year, have more and more difficulty in finding a sufficient number of people to work on the land.

3. Proposed Solution

The new contribution of this research, showed in Figure 1, consists of:

1. A low-cost robotic arm;
2. A low-cost stereovision system placed in the gripper of the robotic arm in order to detect and locate the fruits;
3. A cubic metal gate with a door opening and closing mechanism is used for human protection;
4. To obtain a full operate in harmony with nature the robotic arm is placed on an agricultural collection wagon drawn by a tractor;
5. To manipulate the position of the robotic arm, a joystick-based solution is proposed.

All these five points allow us to avoid the problems discussed in Chapter 2.

The workflow of the system architecture has 3 blocks as shown in Figure 2. Data from the physical world enters the system through the Microsoft Kinect video camera and it is processed by an NVIDIA Jetson Nano. The processed data is given as output in the form of the motion of the actuators present in the robotic arm.

In particular, I have identified 3 different stages in the environment:

- The **safety** stage. Is when the robotic arm is not in operation and the opening and closing door mechanism of the cubic metal gate is closed. It is, therefore, all those periods used by the operator to move between different campaigns.
- The **possibility-critical** stage. This is when the opening and closing door mechanism of the cubic metal gate is opening and the robotic arm picks up a fruit.
- The **critical** stage: Is all the situation where the opening and closing door mechanism of the cubic metal gate is opening and the robotic arm pick up a fruit, but, we have that the operator got off the tractor for some reason.

All these three stages can be managed with Artificial Intelligence (AI) applied through the information retrieved by the video cameras.

3.1. Microsoft Kinect Sensor

The Microsoft Kinect [1] video camera is composed of the *Depth Sensor* and the *RGB Camera*. The first one is used to capture the motion of an object as depth data and skeletal data. To get depth data from the received light pattern, the whole frame is compared to a "flat surface" stored in the memory of the Kinect, which was defined earlier during calibration. Thus, the distance of each point to the camera is calculated and thus the 3D depth data is generated.

3.2. Microcontroller

For this, I use the NVIDIA Jetson Nano [2] microcontroller. This will need to have the necessary APIs installed in it for communication with the Kinect. To detect the apple positions, I used the SSD [13] approaches.

SSD is a method based on CNN, which detects objects in an image using a single deep neural network. The other possible detection methods are Faster R-CNN, and You Only Look Once, among others.

For all the fruits detected by the SSD, I choose the fruit that was closest to the robot arm, and then, with the joystick I move the robotic arm.

3.3. Actuators

I use a low-cost robotic arm with similar functions to a human arm. The links of such a manipulator are connected by joints allowing either rotational motion. The terminus of the kinematic chain of the manipulator is called the end-effector and it is analogous to the human hand.

The robotic arm was built keeping the end application in mind, i.e., agriculture. Servo systems were used at each of the joints to allow precise movement and control over the arm.

Figure 3. The 3D view of the model.

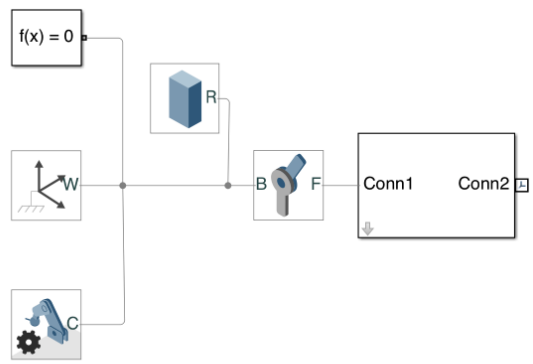

Figure 4. Simscape Multibody schematic of the whole system.

4. Modelling and Simulation

The implementation of the system is divided into two parts: hardware implementation and software implementation. The hardware implementation includes the components presented before: Microsoft Kinect, NVIDIA Jetson Nano, and the robotic arm. The software implementation includes, instead, the modeling of the opening and closing door mechanism of the cubic metal gate and the kinematics of the robotic arm.

4.1. The Opening and Closing Door Mechanism

For modeling the opening and closing door mechanism of the cubic metal gate I used the *Simscape Multibody* framework. This provides a simulation environment for 3D mechanical systems such as robots, vehicle suspensions, construction equipment, etc. It is possible to model systems using blocks representing bodies, joints, constraints, force elements, and sensors.

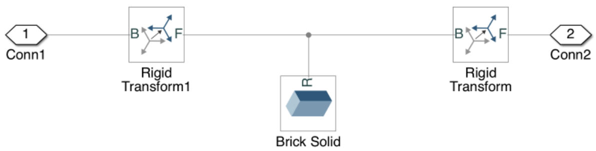

Figure 5. Simscape Multibody schematic of the door.

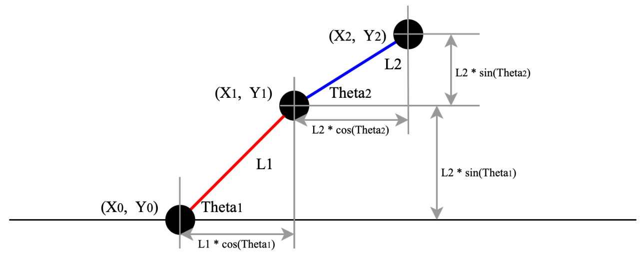

Figure 6. Robotic arm challenge.

4.1.1 Door Mechanism System Modelling

At the MATLAB command line, enter `smnew`. A Simscape Multibody model template with commonly used blocks opens.

Then, delete the `Simulink-PS Converter`, `PS-Simulink Converter`, and `Scope` default blocks because they are not used in this work. In particular, I use:

- Two `Brick Solid block`. This block is a shape with surfaces to the reference X, Y, and Z axes;
- A `Revolute Joint block`. This block represents a joint with one rotational degree of freedom;
- Two `Rigid Transform block`. This block applies a time-invariant transformation between two frames. The transformation rotates and translates the follower port frame (F) with respect to the base port frame (B). It is usually used with `Brick Solid block` to model rigid bodies.

Figure 5 represents the schematic of the door. In particular, in the `Brick Solid block` dialog box, I specified the following parameters:

- **Dimensions**: `[20, 1, 30] (cm)`;
- **Density**: `2700 (kg/m^3)`;
- **Color**: `[0.25, 0.40, 0.70] (R G B)`.

These parameters define the simple link's physical properties, such as shape, mass, and appearance. Instead. In the `Rigid Transform block`, I specified the following parameters:

- **Method**: `Standard Axis`;
- **Axis**: `[-X/+X]`;
- **Offset**: `L/2`.

These parameters specify the locations of two end frames.

In Simscape Multibody it is possible to build a complex system using several simple models. The physical parameters of these simple models usually need to be adjusted to fit different design requirements. To simplify the parameter adjusting process, you can create a `Subsystem blocks`. `Subsystem blocks` allows you to update many parameters in a single place. So, I generated the `Subsystem block` of the Figure 5 as we can see in Figure 4. In particular, Figure 4 represents the schematic of the whole system.

To complete the entire system I added a `Revolute Joint block` and another `Brick Solid block` to the model. In this `Brick Solid block` dialog box, I specified the following parameters:

- **Dimensions**: `[4, 4, 4] (cm)`;
- **Color**: `[0.80 0.45 0] (R G B)`.

The `Revolute Joint block` uses the common Z axis as the joint rotation axis. To ensure that the door moves, it is necessary to change the gravity vector so it no longer aligns with the Z axis. To do this, I set:

- **Gravity**: `[0 -9.81 0]`.

4.1.2 Door Mechanism System Simulation

Run the model. The *Mechanics Explorer* opens with a front view of the simple link model. To see the 3D view of the model (Figure 3), click the isometric view button. From the Mechanics Explorer menu bar, select `Tools > Video Creator` and then, save the video of the simulation. The simulation video is attached as a supplementary material.

4.2. The Kinematics of the Robotic Arm

A classic problem with robotic arms is getting the end-effector, the mechanism at the end of the arm responsible for manipulating the environment, to where you need it to be. In our case, the end-effector is a gripper and we want to pick up an object. We know where that object is relative to the robot, so we have to determine the joint angles that get the end-effector to where we want it to be.

Forward Kinematics refers to the use of the kinematics equation of a robot to compute the position of the end-effector from specified values for the joint parameters. The kinematics equations of the robot are used in a lot of fields such as robotics, computer games, and animation.

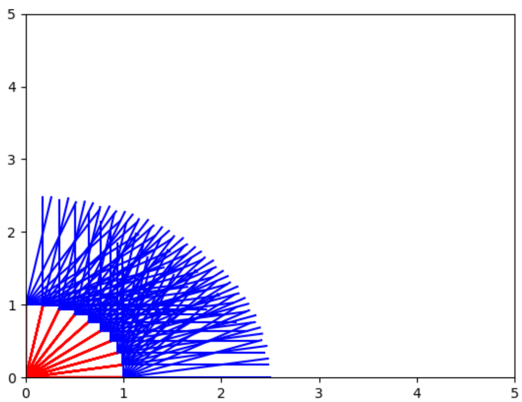

Figure 7. Result of the simulation of the robotic arm.

The reverse process that computes the joint parameters that achieved a specified position of the end-effector is known as *Inverse Kinematics*.

In the Figure 6 a schematic of robotic arm is shown. The arm contain two links, `link1` and `link2` and the length of these both links are `L1` and `L2` respectively. It is consider that `link1` is fixed to the horizontal axis and making an angle θ_1. One hand of `link2` is connected to `link1` and make the θ_2 angle. Is is consider that the starting point coordinate of `link1` is (X_0, Y_0) and the end point coordinate of `link1` is (X_1, Y_1). Similarly for `link_2`, start and end point coordinates are (X_1, Y_1) and (X_2, Y_2) respectively. Let $X_0 = 0$ and $Y_0 = 0$, using trigonometry formula I can say that:

- $X_1 = L1 * cos(\theta_1)$;
- $X_2 = X_1 + L2 * cos(\theta_2)$;
- $Y_1 = L1 * sin(\theta_1)$;
- $Y_2 = Y_1 + L2 * cos(\theta_2)$.

4.2.1 Simulation of the Kinematics of a Robotic Arm

To create a program that automatic solve this problem, firstly I set the following input data:

- **Arm length**: `L1` and `L2`;
- **Angle**: θ_1 and θ_2;
- **Position of coordinate**: X_0 and Y_0.

After giving this data I calculate the value of the other coordinates by using the above equation.

By using the `for` loop I get the position of the links for different angles. I get the graphical form of the result of problem (Figure 7) using the Algorithm 1. Again, the source code is attached as a supplementary material.

Algorithm 1: Simulation of the robotic arm

```
Result: The graphical form of data.

# Input arm length initialization.

# Theta start and end values initialization.

# Input position of (X0, Y0) initialization.

# Number of theta values initialization.

# Define angle variable
theta1= [], theta2= []

# Values of angle find out.
for i in range(0, nTheta) do
    thetaValue
    theta1.append(thetaValue)
    theta2.append(thetaValue)
end

for i in theta1 do
    for j in theta2 do
        Calculate coordinates (x1, y1)
        Calculate (x2, y2)
        Plot
    end
end
```

5. Results, Discussion and Conclusion

This study presents *AUTOHFS*, an Automatic Fruit Harvesting System.

The system is composed using a low-cost robotic arm, a low-cost stereovision camera placed in the gripper tool, and a cubic metal gate for human protection. To obtain a full operate in harmony with nature the robotic arm is placed on an agricultural collection wagon drawn by a tractor. To manipulate the position of the robotic arm, a joystick-based solution is proposed.

The paper first proposes an overview of the worldwide development of scientific contributions such are flying robots, forester robot, and, in the majority of cases, mobile robots. Then, discuss the main obstacles of this kind of system and I propose a new solution to innovate the state-of-the-art.

At the end, the paper proposes the door mechanism system modeling and simulation of the kinematics of a robotic arm.

References

[1] Azure kinect. https://azure.microsoft.com/it-it/services/kinect-dk/. Accessed: 2021-05-29. 3

[2] Nvidia jetson nano. https://developer.nvidia.com/embedded/jetson-nano-developer-kit. Accessed: 2021-05-13. 3

[3] Anna Adamiak, Artur Zdunek, Andrzej Kurenda, and Krzysztof Rutkowski. Application of the biospeckle method for monitoring bull's eye rot development and quality changes of apples subjected to various storage methods—preliminary studies. *Sensors*, 12(3):3215–3227, 2012. 2

[4] Rafael V Aroca, Rafael B Gomes, Rummennigue R Dantas, Adonai G Calbo, and Luiz MG Gonçalves. A wearable mobile sensor platform to assist fruit grading. *Sensors*, 13(5):6109–6140, 2013. 2

[5] Christopher Assad, Michael Wolf, Adrian Stoica, Theodoros Theodoridis, and Kyrre Glette. Biosleeve: A natural emg-based interface for hri. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 69–70. IEEE, 2013. 2

[6] Johan Baeten, Kevin Donné, Sven Boedrij, Wim Beckers, and Eric Claesen. Autonomous fruit picking machine: A robotic apple harvester. In *Field and service robotics*, pages 531–539. Springer, 2008. 2

[7] D Font, T Pallejà, M Tresanchez, M Teixidó, D Martinez, J Moreno, and J Palacín. Counting red grapes in vineyards by detecting specular spherical reflection peaks in rgb images obtained at night with artificial illumination. *Computers and electronics in agriculture*, 108:105–111, 2014. 2

[8] D Font, M Tresanchez, T Pallejà, M Teixidó, D Martinez, J Moreno, and J Palacín. An image processing method for in-line nectarine variety verification based on the comparison of skin feature histogram vectors. *Computers and Electronics in Agriculture*, 102:112–119, 2014. 2

[9] Emilio Gil, Jordi Llorens, Jordi Llop, Xavier Fàbregas, and Montserrat Gallart. Use of a terrestrial lidar sensor for drift detection in vineyard spraying. *Sensors*, 13(1):516–534, 2013. 2

[10] Naoshi KONDO and Noboru KAWAMURA. Methods of detecting fruit by visual sensor attached to manipulator fruit detecting experiment and simulation by computer. *JOURNAL of the JAPANESE SOCIETY of AGRICULTURAL MACHINERY*, 47(1):60–65, 1985. 2

[11] Heiner Lasi, Peter Fettke, Hans-Georg Kemper, Thomas Feld, and Michael Hoffmann. Industry 4.0. *Business & information systems engineering*, 6(4):239–242, 2014. 1

[12] Soo-Chul Lim, Kyoobin Lee, and Dong-Soo Kwon. Human friendly interfaces of robotic manipulator control for handicapped persons. In *Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003)*, volume 1, pages 435–440. IEEE, 2003. 2

[13] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Computer vision–eccv 2016. *Lecture Notes in Computer Science*, 9905:21–37, 2016. 3

[14] Yande Liu, Xudong Sun, Hailiang Zhang, and Ouyang Aiguo. Nondestructive measurement of internal quality of nanfeng mandarin fruit by charge coupled device near infrared spectroscopy. *Computers and Electronics in Agriculture*, 71:S10–S14, 2010. 2

[15] Muhammad Makky and Peeyush Soni. Development of an automatic grading machine for oil palm fresh fruits bunches (ffbs) based on machine vision. *Computers and electronics in agriculture*, 93:129–139, 2013. 2

[16] Yasir M Mustafah, Rahizall Noor, Hasbullah Hasbi, and Amelia Wong Azma. Stereo vision images processing for real-time object distance and size measurements. In *2012 international conference on computer and communication engineering (ICCCE)*, pages 659–663. IEEE, 2012. 2

[17] Gonzalo Pajares, Andrea Peruzzi, and Pablo Gonzalez-de Santos. Sensors in agriculture and forestry. *Sensors (Basel, Switzerland)*, 13:12132–9, 09 2013. 1

[18] Manuel Perez-Ruiz, Jacob Carballido, Juan Agüera, and Antonio Rodríguez-Lizana. development and evaluation of a combined cultivator and band sprayer with a row-centering rtk-gps guidance system. *Sensors*, 13(3):3313–3330, 2013. 2

[19] Neo Ee Sian, Kazuhito Yokoi, Shuuji Kajita, Fumio Kanehiro, and Kazuo Tanie. Whole body teleoperation of a humanoid robot development of a simple master device using joysticks. *Journal of the Robotics Society of Japan*, 22(4):519–527, 2004. 2

[20] Michal Stočes, Jiří Vaněk, Jan Masner, and Jan Pavlík. Internet of things (iot) in agriculture-selected aspects. *Agris on-line Papers in Economics and Informatics*, 8(665-2016-45107):83–88, 2016. 2