sensors

MDPI

Article

Recycling and Updating an Educational Robot Manipulator with Open-Hardware-Architecture

Antonio Concha Sánchez [1], Juan Felipe Figueroa-Rodríguez [1], Andrés Gerardo Fuentes-Covarrubias [1], Ricardo Fuentes-Covarrubias [1] and Suresh Kumar Gadi [2,*]

[1] Facultad de Ingeniería Mecánica y Eléctrica, Universidad de Colima, Coquimatlán Colima 28400, Mexico; aconcha@ucol.mx (A.C.S.); jfigueroa_@ucol.mx (J.F.F.-R.); fuentesg@ucol.mx (A.G.F.-C.); fuentesr@ucol.mx (R.F.-C.)

[2] Facultad de Ingeniería Mecánica y Eléctrica, Universidad Autónoma de Coahuila, Torreón, Coahuila 27276, Mexico

* Correspondence: Research@SKGadi.com; Tel.: +52-(1)-449-182-0153

Received: 28 February 2020; Accepted: 6 March 2020; Published: 18 March 2020

Abstract: This article presents a methodology to recycle and upgrade a 4-DOF educational robot manipulator with a gripper. The robot is upgraded by providing it an artificial vision that allows obtaining the position and shape of objects collected by it. A low-cost and open-source hardware solution is also proposed to achieve motion control of the robot through a decentralized control scheme. The robot joints are actuated through five direct current motors coupled to optical encoders. Each encoder signal is fed to a proportional integral derivative controller with anti-windup that employs the motor velocity provided by a state observer. The motion controller works with only two open-architecture Arduino Mega boards, which carry out data acquisition of the optical encoder signals. MATLAB-Simulink is used to implement the controller as well as a friendly graphical interface, which allows the user to interact with the manipulator. The communication between the Arduino boards and MATLAB-Simulink is performed in real-time utilizing the Arduino IO Toolbox. Through the proposed controller, the robot follows a trajectory to collect a desired object, avoiding its collision with other objects. This fact is verified through a set of experiments presented in the paper.

Keywords: open-source hardware; educational robot; artificial vision; decentralized control; recycled robot; robot control

1. Introduction

Robot manipulators are one of the most widely used mechatronic systems in the industry, whose applications include the assembly of elements, as well as the welding and painting of parts. Due to its great usefulness in the industry, it is very important to study its kinematics, dynamics, and automatic control in engineering careers related to mechatronics and robotics. A characteristic of robot manipulators is that they are usually manufactured with a closed architecture in their automatic control. Once a robot meets its end-of-Life, it is resold, reused, or recycled, which are known as the "3Rs" [1]. A manipulator is usually classified as unusable equipment when its controller is damaged. The reason is that the cost of its reparation can be expensive. In this case, it could be convenient to propose a low cost methodology to re-manufacture the robot, where its mechanical components can be reused and its control system is redesigned using an open architecture.

In the literature, there are several motion controllers for robot manipulators, some of which are recycled and are employed in experimental educational platforms to validate the theory seen in class. Bomfim et al. [2] re-manufactured the controller of a robot manipulator for the automotive

industry, whose trajectories are designed with the MATLAB and Mach3 programs. Sanfilippo et al. [3] and Soriano et al. [4] recycled robotic arms for automation engineering education. The robot in [3] is useful for student academic training, whose controller cabinet was developed by students using a PLC architecture. On the other hand, the robot in [4] is built with recycled LEGO pieces, and it is controlled with an Arduino Mega board, which is programmed using the Simulink Support Package for Arduino Hardware. Yen et al. [5] developed a low-cost collaborative robot that employs a virtual force sensor and stiffness control to safety collision detection and low-precision force control. The authors of [6–8] presented educational robot manipulators, whose movements are carried out by means of radio control servomotors that have controllers that cannot be modified. The manipulator in [6] is operated from a graphical interface, while the robotic arm in [7] has two cameras to detect, collect, and move objects. On the other hand, Cocota et al. [8] described the design and development of a 4-DOF manipulator with a low cost of approximately USD 150. Robot manipulators based on Dynamixel servomotors are developed in [9,10], where Rivas et al. [9] presented the control system of a 6-DOF manipulator controlled through the Python software, whereas Kim and Song [10] designed a mechanism to counterbalance the gravitational torques of a 5-DOF robot arm. Manzoor et al. [11] developed an experimental platform called AUTAREP, which consists of a robotic arm, model ED7220C, from the ED Corporation. The authors of [11] replaced the original controller of the manipulator, which has a closed architecture, with a controller manufactured by them. This controller is described in [12], has an open architecture, is programmed through a graphical user interface (GUI), and it has been implemented as: PID regulator [13], computed torque controller [14], and optimal regulator [15]. On the other hand, sliding mode controllers, an adaptive regulator, and neural networks are, respectively, proposed in [16–21], for tracking control of robot manipulators employed for educational and research purposes. In the literature, the development of virtual or simulated robot manipulators is also proposed; for example, the authors of [22–25] presented robust controllers validated in simulations with the so-called PUMA 560 robot manipulator. However, these manipulators usually do not contemplate friction and backlash, which are present in a real manipulator and cause tracking errors, limit cycles, and other problems that directly affect the manipulator's motion control.

This article presents an experimental educational platform based on a recycled 4-DOF robotic arm with gripper, which is employed to teach and study its kinematics, dynamics, and automatic control. The recycled robot reuses the mechanical parts and motors of a manipulator from ED Corporation, model ED7220C, whose controller was damaged. Since its repair cost is high, a in-house design is considered. The proposed experimental platform is an upgraded version of our first work described in [26] to which several capabilities has been added such as: a force sensor inside the robot gripper to detect objects; artificial vision to locate objects and to pick them up according to its shape and color; an anti-windup technique to a PID controller to improve transient response of the movements; a fine tuning of the controller gains to reduce tracking errors; a graphical interface to interact with a user, and trajectory planning to avoid the collision of the robot with objects. All the programming of the recycled robot is carried-out in MATLAB Simulink. Its motion controller has a decentralized scheme that does not take into account the robot dynamics, and it is applied to five direct current (dc) motors coupled to the robot joints, whose positions are detected by the optical encoders. A parameter identification methodology based on the Recursive Least Squares method is also designed to estimate the parameters of the dc motors, which are subsequently employed to design their controllers and state observers that estimate the joint velocities. Data acquisition of the encoders is realized by two Arduino Mega boards. The communication between these boards and MATLAB-Simulink is carried out in real-time using the open-source ARDUINO IO Toolbox [27]. The proposed controller has the advantage that it is programmed with a visual environment based on block diagrams that has a higher level of abstraction than the programming language used in the AUTAREP platform [11]. Furthermore, in comparison with the on by Soriano et al. [4], the proposed motion controller is developed with a Simulink toolbox that reads encoder signals, thus simplifying their acquisition. Programming the controller and the artificial vision in Simulink has the advantage of monitoring all signals of the

controller by means of scopes, and of using blocks that facilitate the design of other control algorithms such as robust, optimal, adaptable, fuzzy, and neural networks, among others. It is worth mentioning that the proposed experimental educational platform is a key element of the Robotics Laboratory of the Faculty of Mechanical and Electrical Engineering (FIME) at the Universidad de Colima in Mexico, where undergraduate students validate the theory seen in courses of robotics and automatic control, and they also use the robot for research purposes. For example, it was used by three undergraduate students during their final degree projects, whose achievements are reflected in this manuscript. Similarly, the robot has also been used in internal workshops to motivate students to join and remain at the FIME, as well as to show them the importance of robotics and automatic control.

The article is organized as follows. Section 2 describes the architecture of the recycled robot manipulator. Its kinematics and dynamics are presented in Sections 3 and 4, respectively. Section 5 shows the parameter identification of the robot actuators, and their parameter estimates are used in Section 6 for the design of PID controllers and state observers. On the other hand, the trajectory planning, the artificial vision, the GUI interface, and experimental experiments are discussed in Sections 7–10, respectively. Finally, Section 11 establishes the conclusions of the manuscript.

2. Robot Architecture

The achitecture of the recycled robot manipulator is shown in Figure 1. It consists of a 4-DOF robotic arm, model ED7220C, developed by the ED Corporation from Korea. Its joints are shown in Figure 2, which are located at the body, shoulder, elbow, and wrist. The manipulator also has a gripper to collect objects, and, over it, there is a resistive force sensor, model FSR 402 from the Interlink Electronics company of USA. This sensor determines if an object is inside the gripper. All joints and gripper have limit switches, which indicate their minimum and maximum displacements. Moreover, these switches allow establishing the manipulator initial position. To achieve this position, the manipulator also has an 11.43 cm flex sensor from the American company SparkFun Electronics; this sensor is located at the elbow joint. The body, shoulder, and elbow joints are coupled to permanent dc motors, model DME38B50G-116 from the company Servo of Japan .In the sequel, these joints are denoted as q_1, q_2, and q_3, respectively. On the other hand, the wrist and gripper are, respectively, driven by DME38B50-115 and DME33B37G-171 dc motors also from Servo. The wrist joint, denoted as q_4, is actuated by a differential gear mechanism [28] coupled to two dc motors. Each motor of the robot includes an optical encoder to determine its position and is connected to a gearbox to increase its torque while reducing its speed.

Two Arduino Mega boards, from the Italian company Arduino, control the position of the motors. Each board acquires the position data of three motors, and it is communicated with a personal computer through a USB connection. The control signal of each motor is produced by the program MATLAB-Simulink from the American corporation Mathworks. This signal is communicated to the Arduino Mega board via the Arduino IO toolbox, which converts the control signal to Pulse Width Modulation (PWM). The PWM signals are fed to L298N Dual H-Bridge Driver Modules from the Chinese corporation Haitronic. These modules provide the power to the dc motors. On the other hand, a webcam from the Swiss company Logitech, model C525, provides artificial vision to the manipulator and permits obtaining the positions of the objects taken by the manipulator. Image processing is carried out in MATLAB-Simulink using the Computer Vision Toolbox. A user-friendly interface GUI, created in MATLAB, permits selecting the shape and color of the objects taken by the manipulator, whose kinematics is described below. It is worth mentioning that the main components of the proposed controller, such as webcam, Arduino boards, motor drivers, force and flex sensors, connectors, and cables, have a total cost of about 230 USD.

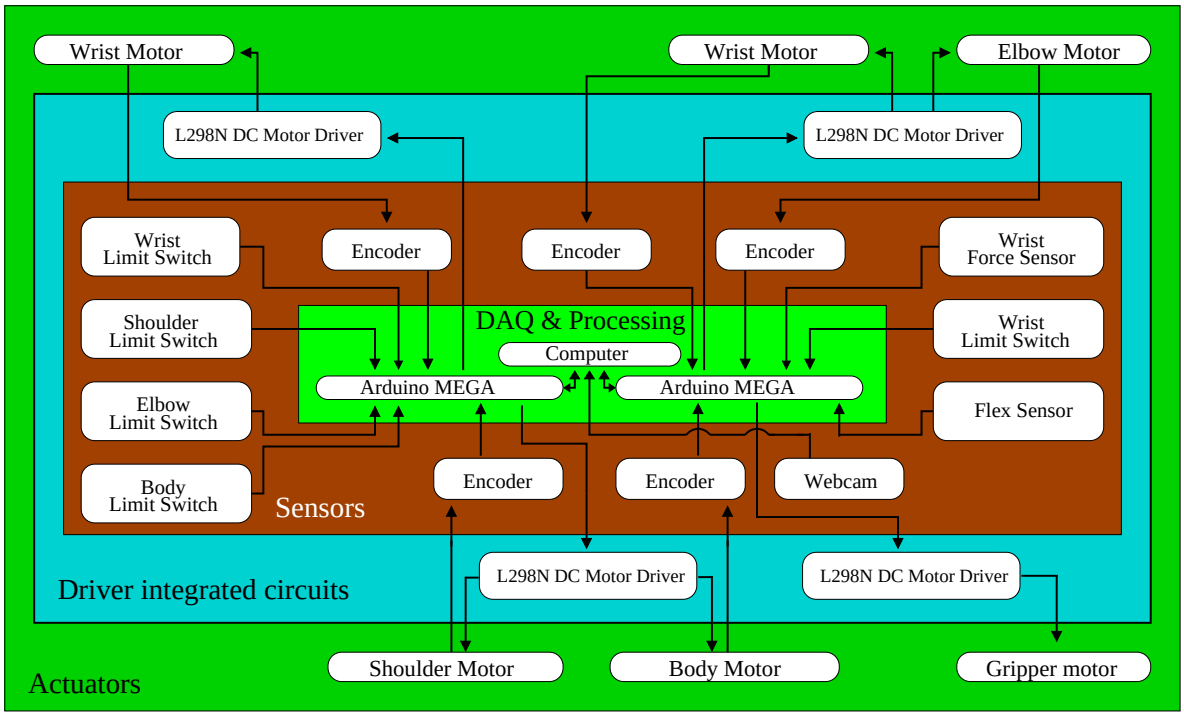


Figure 1. Robot architecture.

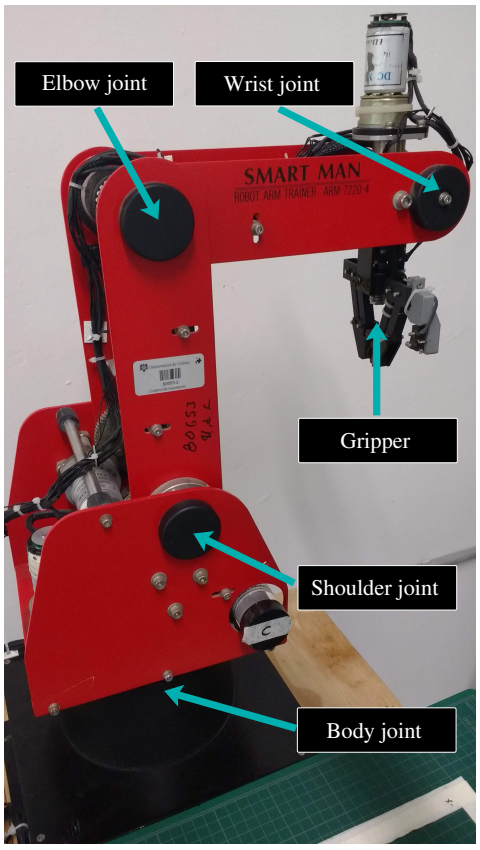


Figure 2. Recycled manipulator.

3. Robot Kinematics

Figure 3 shows the structure of the manipulator, as well as its 4-DOF q_1, q_2, q_3, q_4, which are the joint positions. This figure also shows the coordinates (x, y, z) of the end-effector, as well as the angle ϕ that specifies its orientation.

3.1. Forward Kinematics

Using the trigonometric relationships between the links and their lengths leads to the forward kinematics of the manipulator, which provides the position of the end-effector with respect to the joint positions. It is given by:

$$\begin{aligned} x &= C_1 \left(l_1 \cdot C_2 + l_2 \cdot C_{23} + l_3 \cdot C_{234} \right) \\ y &= S_1 \left(l_1 \cdot C_2 + l_2 \cdot C_{23} + l_3 \cdot C_{234} \right) \\ z &= d_1 + l_1 \cdot S_2 + l_2 \cdot S_{23} + l_3 \cdot S_{234} \end{aligned} \tag{1}$$

where $d_1 = 370\,\text{mm}$, $l_1 = l_2 = 220\,\text{mm}$, and $l_3 = 140\,\text{mm}$ are the lengths of the links. Furthermore, we used the following shorthand notations

$$C_i = \cos\left(q_i\right); \quad S_i = \sin\left(q_i\right); \quad C_{iwj} = \cos\left(q_i + q_w + q_j\right); \quad S_{iwj} = \sin\left(q_i + q_w + q_j\right)$$

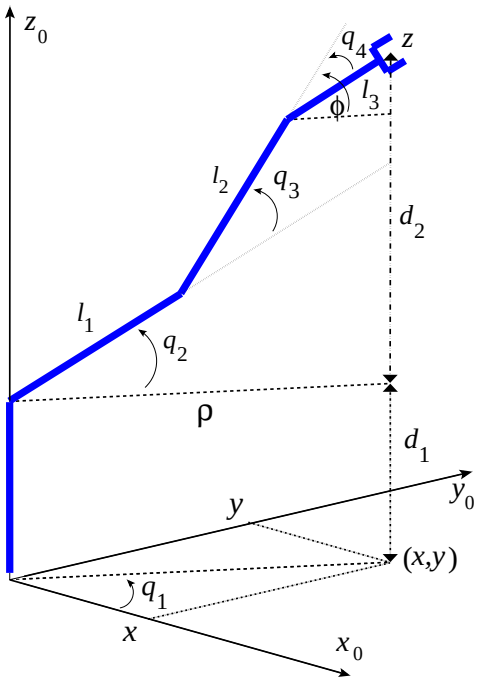


Figure 3. Kinematic scheme of the manipulator.

3.2. Robot Workspace

The workspace is the volume that can reach the end-effector, and it is constructed through the forward kinematics in Equation (1), as well as the range covered by the joints of the manipulator, as shown in Table 1.

Table 1. Range for the joints of the manipulator.

Joint	Lower Limit (°)	Upper Limit (°)
q_1	25	335
q_2	0	120
q_3	−66	116
q_4	−90	−45

Figure 4 shows the arm movement range in the plane xy at a height of 115 mm with respect to the robot base. The figure shows that this range encompasses a radius of approximately 400 mm. The objects manipulated by the robot are placed over a rectangle area A located in front of the robot base. The dimensions n and l of this area are determined analytically to maximize it, as follows. Note that n and l are given by:

$$l = r\sin(\alpha), \qquad p + n = r\cos(\alpha) \tag{2}$$

where $r = 400$ mm, $p = 200$ mm, and α is an unknown angle to be determined. Therefore, the area A can be written as

$$A = [r\cos\alpha - p][r\sin\alpha] \tag{3}$$

The derivative of Equation (3) with respect to α is given by

$$\frac{dA}{d\alpha} = r\left(r[\cos^2\alpha - \sin^2\alpha] - p\cos\alpha\right) = r\left(r[2\cos^2\alpha - 1] - p\cos\alpha\right) = r\left(2r\cos^2\alpha - p\cos\alpha - r\right) \tag{4}$$

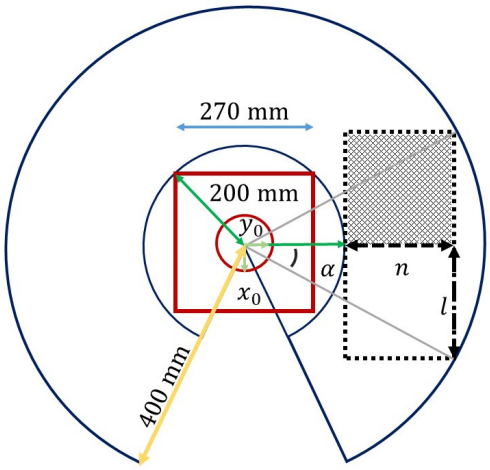


Figure 4. Arm movement range.

By equaling this derivative to zero produces the critical point $\alpha = 0.568$ rad = 32.5°, which gives the maximum rectangular area A. For simplicity, a value of $\alpha = 0.568$ rad = 30° is used that yields the dimensions n = 200 mm, l = 140 mm, and the area A shown in Figure 5, which is represented as a purple rectangle. Note that this figure also shows that the robot camera is located 585 mm above the area A.

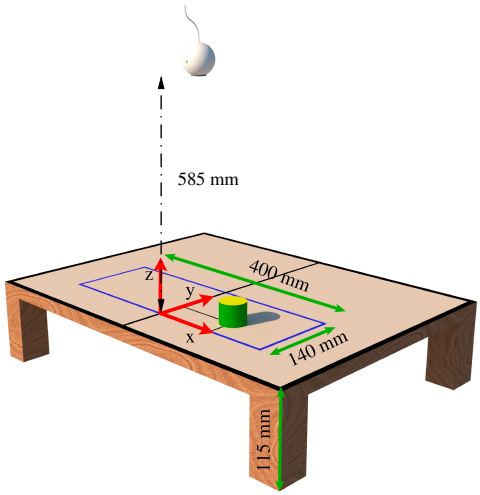


Figure 5. Dimensions of the area where objects are placed.

3.3. Inverse Kinematics

The section describes the inverse kinematics of the manipulator whose objective is to obtain the joint positions $q_1, i = 1, \ldots, 4$ so that the end-effector is placed in a specific position and orientation. The inverse kinematics of the robot is given by [29]:

$$\begin{aligned} d_2 &= z - d_1 \\ \rho &= \sqrt{x^2 + y^2} \\ p_{wx} &= \rho - l_3 \cos(\varphi) \\ p_{wy} &= d_2 - l_3 \sin(\varphi) \\ D &= \frac{p_{wx}^2 + p_{wy}^2 - l_1^2 - l_2^2}{2 l_1 l_2} \\ q_1 &= \operatorname{atan2}(y, x) \\ q_2 &= \operatorname{atan2}\left(p_{wy}, p_{wx}\right) - \operatorname{atan2}\left(l_2 \sin(q_3), l_1 + l_2 \cos(q_3)\right) \\ q_3 &= \operatorname{atan2}\left(\sqrt{1 - D^2}, D\right) \\ q_4 &= \phi - q_2 - q_3 \end{aligned} \tag{5}$$

where $\operatorname{atan2}(y^*, x^*)$ represents the arctangent of y^*/x^* and takes into account the sign of each argument to determine the quadrant corresponding to the angle between x^* and y^*.

The next section presents the dynamic equation of the manipulator that takes into account the torques required for the execution of the robot motion.

4. Robot Dynamics

The dynamic behavior of the manipulator is described by the following expression [29]

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{f}(\dot{\boldsymbol{q}}) = \boldsymbol{\tau} \tag{6}$$

where $\boldsymbol{q} = [q_1, q_2, q_3, q_4]^T$ is the vector of joint positions; $\dot{\boldsymbol{q}}$ is the angular velocity vector, $\boldsymbol{M}(\boldsymbol{q}) = \boldsymbol{M}(\boldsymbol{q})^\intercal \in \mathbb{R}^{4\times4}$ is called inertia matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} \in \mathbb{R}^{4\times1}$ represents a centrifugal and Coriolis force vector, and $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^{4\times1}$ is a gravitational forces vector. In addition, $\boldsymbol{f}(\dot{\boldsymbol{q}}) \in \mathbb{R}^{4\times1}$ is a frictional forces vector and $\boldsymbol{\tau} \in \mathbb{R}^{4\times1}$ is a vector of torques applied by the actuators at the joints.

4.1. Dynamic Model of the Actuators

The set of the joint actuators can be represented by the following matrix differential equation [30]:

$$\ddot{\boldsymbol{q}} + \boldsymbol{G}\dot{\boldsymbol{q}} + \boldsymbol{R}\boldsymbol{\tau} = \boldsymbol{K}\boldsymbol{u} \tag{7}$$

where

$$\begin{aligned} \boldsymbol{G} &= \text{diag}\,(a_1, a_2, a_3, a_4) \\ \boldsymbol{R} &= \text{diag}\left(\frac{1}{J_{m1}r_1^2}, \frac{1}{J_{m2}r_2^2}, \frac{1}{J_{m3}r_3^2}, \frac{1}{J_{m4}r_4^2}\right) \\ \boldsymbol{K} &= \text{diag}\,(b_1, b_2, b_3, b_4) \\ \boldsymbol{u} &= \text{diag}\,(u_1, u_2, u_3, u_4) \\ \boldsymbol{\tau} &= \text{diag}\,(\tau_1, \tau_2, \tau_3, \tau_4) \end{aligned} \tag{8}$$

a_i and b_i, $i = 1, 2, 3, 4$ are positive parameters. u_i, τ_i, J_{mi}, and r_i are the input voltage, load torque, motor inertia, and gear reduction ratio of the ith joint actuator, respectively. $\text{diag}(p)$ represents a diagonal square matrix with the elements of p in the main diagonal.

4.2. Mathematical Model of the Robot Manipulator with Actuators

Substituting τ of Equation (6) into Equation (7) yields

$$[\boldsymbol{R}\boldsymbol{M}(\boldsymbol{q}) + \boldsymbol{I}]\ddot{\boldsymbol{q}} + \boldsymbol{R}\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{R}\boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{R}\boldsymbol{f}(\dot{\boldsymbol{q}}) + \boldsymbol{G}\dot{\boldsymbol{q}} = \boldsymbol{K}\boldsymbol{u} \tag{9}$$

where $\boldsymbol{I}$ is the identity matrix of size 4×4. The previous model is considerably reduced when the gear ratios r_i, $i = 1, 2, 3, 4$ are high, i.e., $r_i \gg 0$. In this case, $\boldsymbol{R} \approx \boldsymbol{O}$ and Equation (9) approximates to:

$$\ddot{\boldsymbol{q}} + \boldsymbol{G}\dot{\boldsymbol{q}} = \boldsymbol{K}\boldsymbol{u} \tag{10}$$

The gear ratios r of the dc motors corresponding to the body, shoulder, elbow, and wrist of the manipulator are 720, 576, 576, and 133, respectively. Since the gear ratios of these motors are high, the dynamics of the manipulator in Equation (6) can be neglected. Therefore, an independent controller can be designed for each robot joint using the linear model in Equation (10).

Parameters a_i and b_i of the dc motor models are unknown, and they are estimated using the recursive least squares algorithm described in the following section.

5. Parameter Identification of the Actuators

The Recursive Least Squares algorithm (RLSM) [31] is used to identify the parameters of the robot actuators, which permit designing: (1) controllers to obtain high precision movements in the manipulator; and (2) state observers to estimate the motor speed. Moreover, the parameter identification is necessary to simulate the robot manipulator and to detect faults on it [32]. To estimate the parameters a_i and b_i, the actuators are operated in closed loop using a proportional controller and a sinusoidal reference input signal.

Since signals $\dot{q}_i(t)$ y $\ddot{q}_i(t)$ of the model in Equation (10) are not available, parameters a_i and b_i are estimated using only measurements of the motor voltage u_i and its position q_i. To this end, each uncoupled model in Equation (7) is filtered by means of the filter $H(s) = \lambda_2/(s^2 + \lambda_1 s + \lambda_2)$, $\lambda_1, \lambda_2 > 0$, which also attenuates measurement noise, thus minimizing its effect in the parameter identification algorithm. This filtering procedure produces:

$$\boldsymbol{z}_i(t) = \boldsymbol{\psi}_i^{\intercal}(t)\boldsymbol{\theta}_i \tag{11}$$

where

$$z_i(t) = \mathcal{L}^{-1}\left[s^2 H(s) Q_i(s)\right]$$
$$\boldsymbol{\psi}_i(t) = \begin{bmatrix} \mathcal{L}^{-1}\left[-sH(s)Q_i(s)\right] \\ \mathcal{L}^{-1}\left[H(s)U_i(s)\right] \end{bmatrix} \tag{12}$$
$$\boldsymbol{\theta}_i = \begin{bmatrix} a_i \\ b_i \end{bmatrix}$$

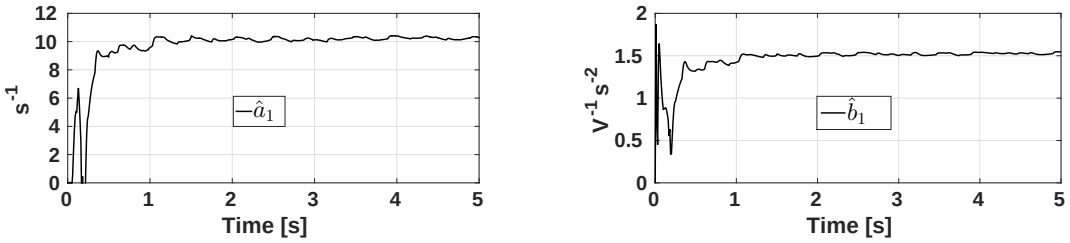
$\mathcal{L}$ and $\mathcal{L}^{-1}$ are the Laplace operator and its inverse, respectively; similarly, $Q_i(s) = \mathcal{L}[q_i(t)]$ and $U_i(s) = \mathcal{L}[u_i(t)]$.

Signals $z_i(t)$ and $\boldsymbol{\psi}_i(t)$ in Equation (12) are sampled every T_s seconds, and they are used by the RLSM given by [31]:

$$\begin{aligned} \widehat{\boldsymbol{\theta}}_i(k) &= \widehat{\boldsymbol{\theta}}_i(k-1) + \boldsymbol{P}_i(k)\boldsymbol{\psi}_i(k)\epsilon_i(k) \\ \boldsymbol{P}_i(k) &= \frac{1}{\gamma_i}\left[\boldsymbol{P}_i(k-1) - \frac{\boldsymbol{P}_i(k-1)\boldsymbol{\psi}_i(k)\boldsymbol{\psi}_i^T(k)\boldsymbol{P}_i(k-1)}{\gamma_i + \boldsymbol{\psi}_i^T(k)\boldsymbol{P}_i(k-1)\boldsymbol{\psi}_i(k)}\right] \\ \epsilon_i(k) &= z_i(k) - \boldsymbol{\psi}_i^T(k)\widehat{\boldsymbol{\theta}}_i(k-1) \end{aligned} \tag{13}$$

where $\hat{\boldsymbol{\theta}}_i(k) = \left[\hat{a}_i(k), \hat{b}_i(k)\right]^T$ is an estimate of $\boldsymbol{\theta}_i$. γ_i is called forgetting factor and satisfies $0 < \gamma_i \leq 1$. In addition, variable $\boldsymbol{P}_i(k) = \boldsymbol{P}_i^{\intercal}(k)$ is called covariance matrix.

To experimentally identify the parameters a_i and b_i, $i = 1, 2, 3, 4$, the RLSM was configured with the following values: $T_s = 0.02\,\text{s}$, $\gamma_i = 0.997$, $\boldsymbol{P}_i(0) = 1000\,\text{diag}[1, 1, 1, 1]$, $\lambda_1 = 20$, $\lambda_2 = 100$. Figure 6 shows the time evolution of estimates $\hat{a}_1$ and $\hat{b}_1$ corresponding to the base actuator. It is shown that the estimates converge in approximately 1 s. Table 2 shows the estimated parameters of each actuator and its corresponding joint.



(a) Parameter estimate $\hat{a}_1$. **(b)** Parameter estimate $\hat{b}_1$.

Figure 6. Parameters $\hat{a}_1$ and $\hat{b}_1$ estimated by the RLSM.

Table 2. Parameter estimates $\hat{a}_i$ and $\hat{b}_i$ of the actuators.

Actuated joint	Estimate $\hat{a}_i$ [s^{-1}]	Estimate $\hat{b}_i$ [$\text{V}^{-1}\text{s}^{-2}$]
Waist	$\hat{a}_1 = 10.2$	$\hat{b}_1 = 1.53$
Shoulder	$\hat{a}_2 = 13.1$	$\hat{b}_2 = 1.10$
Elbow	$\hat{a}_3 = 05.5$	$\hat{b}_3 = 0.88$
Wrist	$\hat{a}_4 = 17.0$	$\hat{b}_4 = 0.88$

6. Robot Control

A PID controller is used to regulate the position of the actuators. This controller is a modification of the basic Proportional Derivative Controller (PID), and it is employed to avoid the set-point kick phenomenon, which consists of abrupt changes of the control signal due to sudden changes of the reference input [33]. The PID controller is given by [33]:

$$u_i(t) = K_{Pi}e_i(t) + K_{Ii}\int e_i(\tau)d\tau - K_{Di}\frac{dq_i(t)}{dt} \tag{14}$$

Note that the derivative action is applied only to the output signal $q_i(t)$. $e_i(t)$ is the position error of the ith joint that is defined as $e_i(t) = q_{di} - q_i$, where q_{di} is the desired position of the ith joint. Moreover, k_{Pi}, k_{Ii}, and k_{Di} are, respectively, the proportional, integral, and derivative gains of the ith position controller. The Routh–Hurwitz stability criterion [34] allows determining the following range of gains k_{Pi}, k_{Ii}, and k_{Di} that guarantee a stable closed-loop system.

$$K_{Di} > \frac{-a_i}{b_i}; \quad K_{Ii} \geq 0; \quad K_{Pi} > \frac{K_{Ii}}{a_i + K_{Di}b_i} \tag{15}$$

Since the nominal values a_i and b_i of each motor are not available, the estimates $\hat{a}_i$ and $\hat{b}_i$ produced by the RLSM are replaced in Equation (15).

In order for the integral term of the PID controller not to cause a slow transient position response due to the voltage saturation of the actuators, this term is implemented using the anti-windup compensation scheme in Figure 7, where K_{ai} is the anti-windup gain. In this figure, $u_{\min}$ and $u_{\max}$ denote the minimum and maximum voltage of the actuators, respectively.

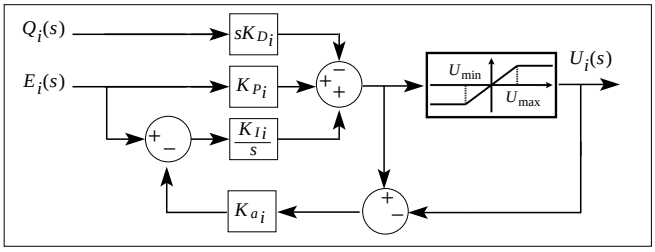


Figure 7. Block diagram of the PID controller with anti-windup compensation.

The PID controller with anti-windup requires the velocities of the robot actuators, which are not available. However, these signals are estimated by means of a state observer described below.

State Observer

The model of an actuator in Equation (10) can be written as the following state space equation:

$$\begin{aligned} \dot{\mathbf{x}} &= \boldsymbol{A}\mathbf{x} + \boldsymbol{B}u \\ y &= \boldsymbol{C}\mathbf{x} \end{aligned} \tag{16}$$

where

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} q_i \\ \dot{q}_i \end{bmatrix} \qquad \boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ 0 & -a_i \end{bmatrix}, \qquad \boldsymbol{B} = \begin{bmatrix} 0 \\ b_i \end{bmatrix}, \qquad \boldsymbol{C} = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{17}$$

For estimating the speed $\dot{q}_i$ of the ith motor, a Luenberger observer is programmed, whose mathematical model is given by [35]:

$$\dot{\hat{\boldsymbol{x}}} = \boldsymbol{A}\hat{\boldsymbol{x}} + \boldsymbol{B}u + \boldsymbol{K}_o(y - \boldsymbol{C}\hat{x}) = (\boldsymbol{A} - \boldsymbol{K}_o\boldsymbol{C})\,\hat{\boldsymbol{x}} + \boldsymbol{B}u + \boldsymbol{K}_o y \tag{18}$$

where $\hat{\boldsymbol{x}} = [\hat{x}_1,\ \hat{x}_2]^T$ is an estimate of $\boldsymbol{x}$, $\boldsymbol{K_o} = [K_1,\ K_2]^T$ is the observer gain, and matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are constituted with the parameter estimates $\hat{a}_i$ and $\hat{b}_i$, respectively. Note that $\hat{x}_2 = \hat{\dot{q}}_i$ is the estimated velocity employed by the PID controller.

Table 3 presents the gains of the PID controllers with anti-windup, as well as the gains of the state observer corresponding to each motor. On the other hand, Figure 8 shows the coupling of the state observer with the PID controller, whose gains K_{Pi}, K_{Ii}, K_{Di}, and K_{ai} are tuned so that the joint response q_i under a step input is sufficiently fast and damped. Moreover, the integral action of the actuator controllers corresponding to the shoulder and elbow joints allow counteracting the gravity forces of the links connected to these joints. Likewise, the observer gains are selected to produce an observer dynamics with both poles equal to -6.

Table 3. Gains of the PID controllers and state observers.

Actuated Joint	K_{Pi}	K_{Ii}	K_{Di}	K_{ai}	K_{1i}	K_{2i}
Body	64	0	16	0	13.8	140.65
Shoulder	56	40	20	700	10.9	142.09
Elbow	61	40	18.5	700	18.5	141.41
Wrist	64	0	24	0	7	143.02

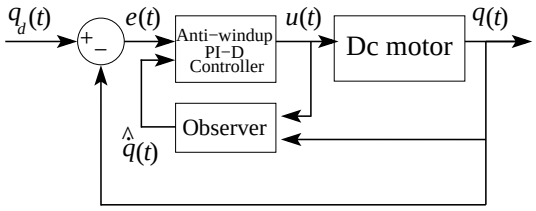


Figure 8. PID controller coupled to the state observer.

The next section describes the trajectory of the end-effector to reach, take, and release an object in the robot workspace. This planned trajectory generates the reference inputs q_{di} to the PID controllers of the actuators, which assure that the robot executes the desired motion.

7. Trajectory Planning

Figure 9 shows a flowchart representing the trajectory planning in the Cartesian workspace of the manipulator. The path planning algorithm consist of a sequence of points along the path, which are denoted as A–D. Point A is the robot initial position, Point B is a position above the object, Point C is the object position, andPoint D is the position where the manipulator deposits the object, as illustrated in Figure 10. Through the sequence of points A-B-C-B-A-D, the robot collects an object and deposits it in a container, avoiding collisions with other objects. To execute the trajectory planning, it is necessary to resort to the robot inverse kinematics in order to convert the Cartesian Points A–D into joint input references q_{di} provided to the PID controllers of the actuators. Due to the high gear reduction ratio of the dc motors, the movement from one point to another point is smooth.

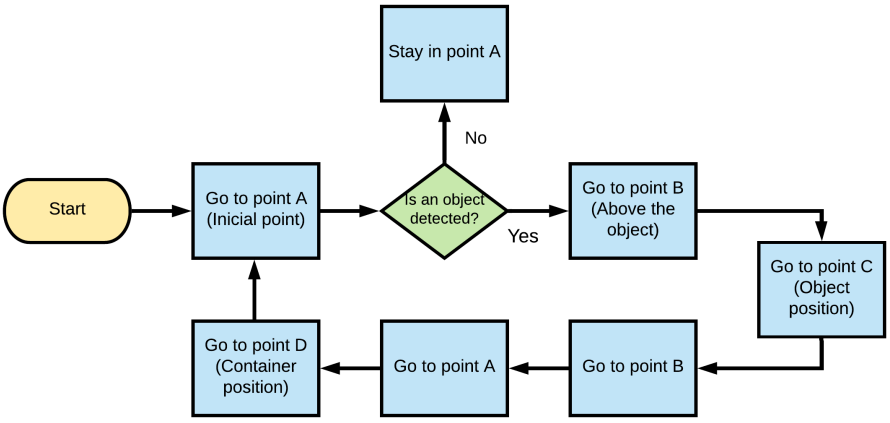


Figure 9. Trajectory planning to move the end-effector.

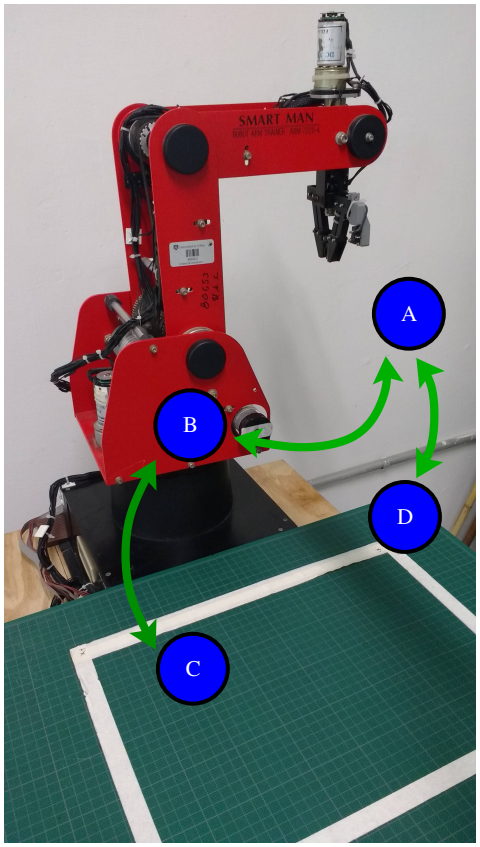


Figure 10. Sequence of Points A–D to move the robot.

To detect the position and shape of the objects gripped by the robot, it has artificial vision, as mentioned below.

8. Artificial Vision

The robot has artificial vision through a webcam located at a height of 700 mm above the robot base. The Image Acquisition Toolbox of Simulink is used to provide the artificial vision to the robot, and students can use this powerful tool for image processing, image segmentation [36], image enhancement, visual perception [37], recognition of 3D objects [36], human-like visual-attention-based artificial vision [38], visual SLAM [39], feature extraction [40], and noise reduction, just to mention a few. Camera images are acquired using the block From Video Device (FVD) of this toolbox, which

extracts their RGB values in a matrix that can be processed by the armory of matrix operators and functions of MATLAB [41]. Images are acquired at five frames per second (FPS). The image of a red circle on the robot workspace is shown in Figure 11.

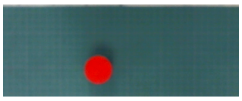
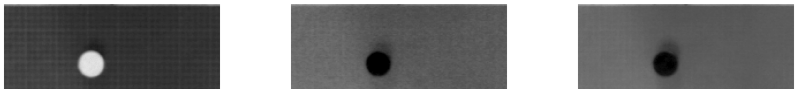


Figure 11. Acquired image of a red circle.

8.1. Color Detection

The FVD block is configured to only visualize objects within the purple rectangle of Figure 5. For this purpose, a resolution of 419 × 147 pixels is used, where each pixel is equal to 0.955 mm. RGB values obtained from an image are processed to produce a grayscale image for each RGB plane. The gray scale of the red circle in Figure 11 is shown in Figure 12 for each RGB plane.



(**a**) Red plane. (**b**) Green plane. (**c**) Blue plane.

Figure 12. Grayscale of the RGB planes corresponding to the circle in Figure 11.

The grayscale image of each RGB plane is filtered in order to smooth the edges of the objects using the block Median Filter of Simulink. Subsequently, a thresholding is applied to the filtered images so that the manipulator can recognize red and yellow objects. The thresholding produces a binary image, where a value of 1 means white, whereas a value of 0 means black. The thresholding employed to detect red and yellow colors is represented by means of the following expression

$$f_{th}(p^r, p^g, p^b) = \begin{cases} 1 & \text{if } p^r_{\text{sup}} \leq p^r \leq p^r_{\text{inf}} \quad \text{and} \quad p^g_{\text{sup}} \leq p^g \leq p^g_{\text{inf}} \quad \text{and} \quad \text{if } p^b_{\text{sup}} \leq p^b \leq p^b_{\text{inf}} \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

where $f_{th}(p^r, p^g, p^b)$ is the output of the thresholding and p^j, $j = r, g, b$, represents a pixel in the planes red, green, and blue, respectively. Their upper and lower limits are written, respectively, as p^j_{sup} and p^j_{inf}, whose values in Table 4 permit detecting red and yellow colors.

Table 4. Thresholding values to detect red and yellow colors.

Detected Color	Limits for the Grayscale Image of the Red Plane	Limits for the Grayscale Image of the Green Plane	Limits for the Grayscale Image of the Blue Plane
Red	$p^r_{\text{sup}} = 255$ $p^r_{\text{inf}} = 200$	$p^g_{\text{sup}} = 35$ $p^g_{\text{inf}} = 0$	$p^b_{\text{sup}} = 35$ $p^b_{\text{inf}} = 0$
Yellow	$p^r_{\text{sup}} = 255$ $p^r_{\text{inf}} = 200$	$p^g_{\text{sup}} = 255$ $p^g_{\text{inf}} = 200$	$p^b_{\text{sup}} = 170$ $p^b_{\text{inf}} = 0$

Finally, a morphological operation, executed with the Erosion and Dilation Simulink blocks, permits smoothing the edges of the resulting binary image, thus producing Figure 13.

Figure 13. Binary image after the morphological operation.

8.2. Shape Detection

To determine the object shape, its area A, perimeter P, compaction C, and centroid C_e are calculated. The previous operations, except for compaction, are performed by the block Blob Analysis of Simulink. The object compaction C is defined as [42–44]:

$$C = \frac{P^2}{A} \tag{20}$$

The manipulator collects circle and square objects whose compaction is given by:

$$C_{\text{square}} = \frac{(4L)^2}{L^2} = 16, \qquad C_{\text{circle}} = \frac{(2\pi\mathcal{R})^2}{\pi\mathcal{R}^2} = 4\pi \tag{21}$$

where L is the length of the sides of the square and $\mathcal{R}$ is the radius of the circle.

Figure 14 briefly describes the artificial vision process and its interaction with the robot motion control.

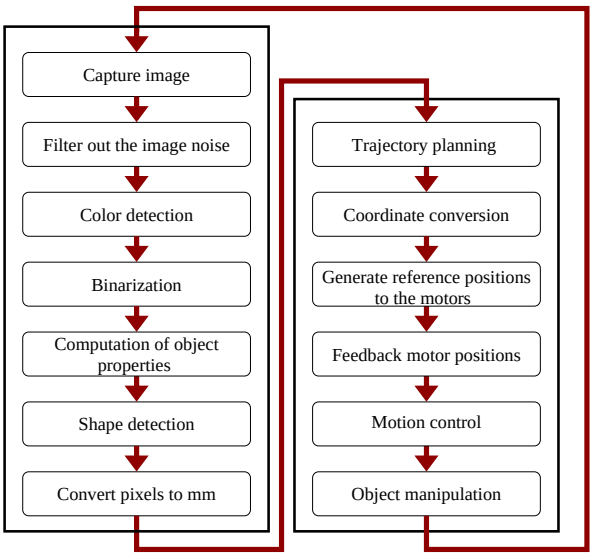


Figure 14. Logic sequence to control the manipulator using artificial vision.

8.3. Correction of the Object Position

The top surface of the objects, which is seen by the camera, has a height with respect to the xy plane where the objects are placed, as shown in Figure 15. For this reason, the position of an object, denoted as x_{obj}, does not coincide with the one provided by the camera, defined as x_{cam}. The following equation is used to compute x_{obj}:

$$x_{\text{obj}} = x_{\text{cam}} - \frac{h_{\text{obj}}}{h_{\text{cam}}} x_{\text{cam}} \tag{22}$$

where h_{obj} and h_{cam} are, respectively, the heights of the object and camera with respect to the robot base. It is important to mention that a similar correction of the object position is carried out on the y axis.

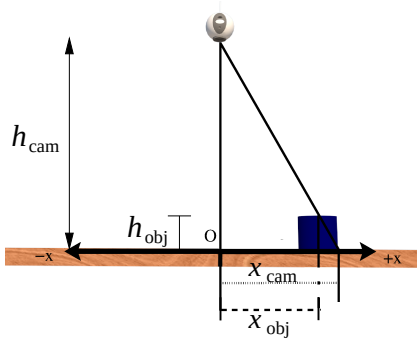


Figure 15. Correction of the object position due to its height.

9. Graphical User Interface

A graphic user interface (GUI) is designed to facilitate the interaction between the user and the recycled manipulator. The GUIDE tool, included in MATLAB to develop high-level graphical and simply layout, was used to design the GUI that is shown in Figure 16. It has buttons to place the manipulator in its initial position, to stop it for emergency, and to select the color and shape of the objects taken by the robot. Moreover, the GUI permits visualizing the positions of the objects in the robot workspace as well as capturing camera images.

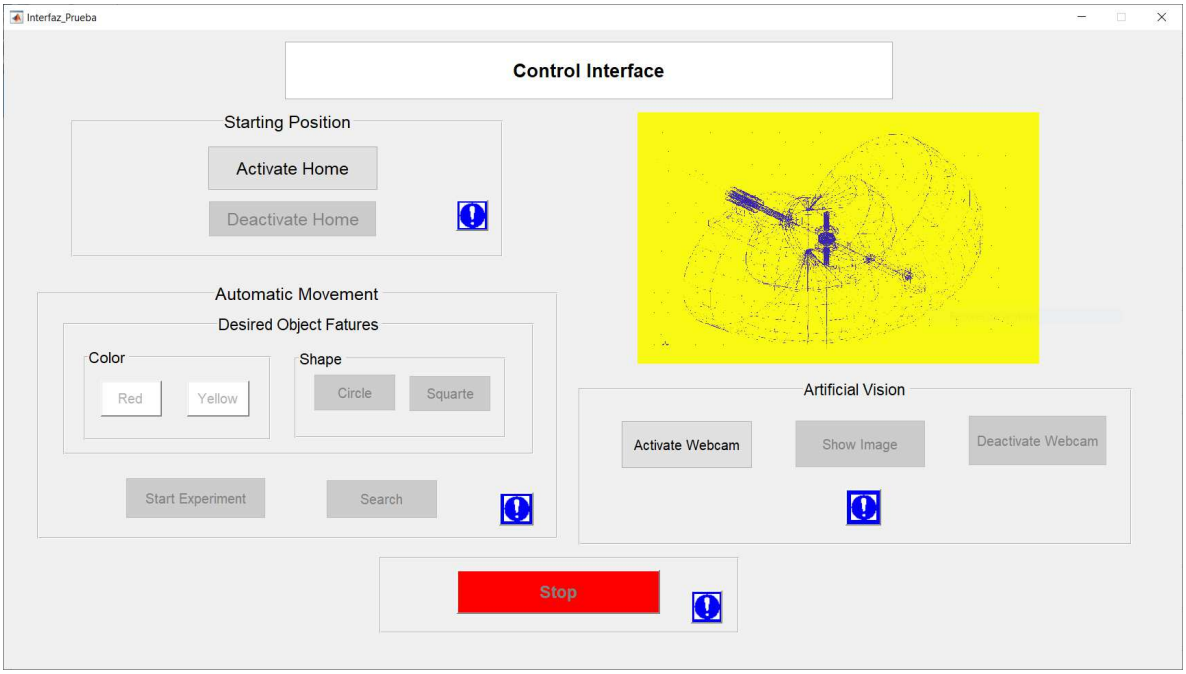


Figure 16. GUI designed to control and visualize the manipulator.

10. Experimental Results

Experimental results obtained with the proposed controller and artificial vision of the robot manipulator are presented in this section. All experiments used a sampling period T_s of 0.06 s. The manipulator's aim was to collect four cylinders randomly arranged, on that bases of which geometric figures with a shape of circle or square that were yellow or red were attached, as shown in Figure 17. These cylinders weighed 46 g; however, the actuators of the manipulator have enough torque to move payloads up to 1 kg [45]. The MATLAB and Simulink files of the robot control programming were uploaded to the open-source website Github [46], including the instructions to run the proposed controller. Moreover, this website contains videos of the experiments that are shown in this section.

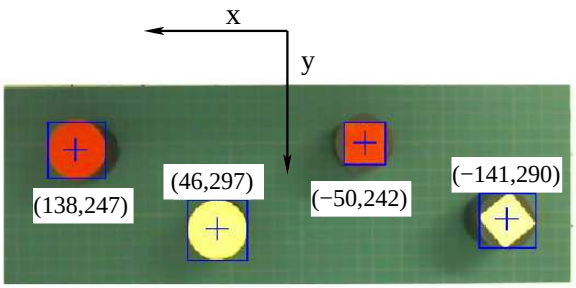


Figure 17. Objects to be collected by the robot.

The first experiment consisted of gripping a yellow circle, whose coordinates were $x = 46$ mm, $y = 297$ mm, and $z = 145$ mm, which were obtained with the help of the artificial vision of the robot. The trajectories of the end-effector in the x, y, and z axes, as well as its angle ϕ, are shown in Figure 18. In this figure, x_d, y_d, z_d, and ϕ_d represent the desired trajectories of the end-effector, where ϕ_d is fixed to $-90°$. The yellow circle was gripped by the robot at approximately 14 s, and it was released in a container at around 24 s. This fact is corroborated with the help of Figure 19 that shows the voltage of the force sensor placed inside the robot gripper. A voltage above 0.4 V of this sensor indicates that the object is gripped by the end-effector.

The joint trajectories q_1, q_2, q_3, and q_4, corresponding to this experiment, and their desired references q_{d1}, q_{d2}, q_{d3}, and q_{d4}, are shown in Figure 20. Note that, based on this figure, the responses of q_1, q_2, q_3, and q_4 are overdamped. Moreover, for each desired value of q_{d1}, the joint q_1 has a settle time of about 1 s. On the other hand, joints q_2 and q_3 have a settle time less than 6 s. It is worth mentioning that the gains of PID controllers for the actuators of joints q_2 and q_3 were selected so that their responses are fast enough with the least possible tracking error, despite the gravitational forces acting over them, which are considered as disturbances. Finally, note that q_4 remains close to its reference $q_{d4} = 0°$.

Figure 21 shows the velocities $\hat{\dot{q}}_1$, $\hat{\dot{q}}_2$, $\hat{\dot{q}}_3$, and $\hat{\dot{q}}_4$ estimated by the designed state observers. It can be observed that $\hat{\dot{q}}_1$, $\hat{\dot{q}}_2$, $\hat{\dot{q}}_3$, and $\hat{\dot{q}}_4$ reach velocities up 41.75, 23.47, 22.5, and 2.3 degrees per second, respectively. These signals are used by the proposed PID controllers of the actuators, whose control signals u_1, u_2, u_3, and u_4 can be seen in Figure 22. Note that, to reduce the tracking errors, the controllers produce signals u_1, u_2, and u_3 that reach their maximum and minimum values of 24 V and -24 V at some instants of time.

Define the position errors in the Cartesian space as follows:

$$\begin{aligned} \tilde{x} &= x_d - x \\ \tilde{y} &= y_d - y \\ \tilde{x} &= z_d - z \\ \tilde{\phi} &= \phi_d - \phi \end{aligned} \tag{23}$$

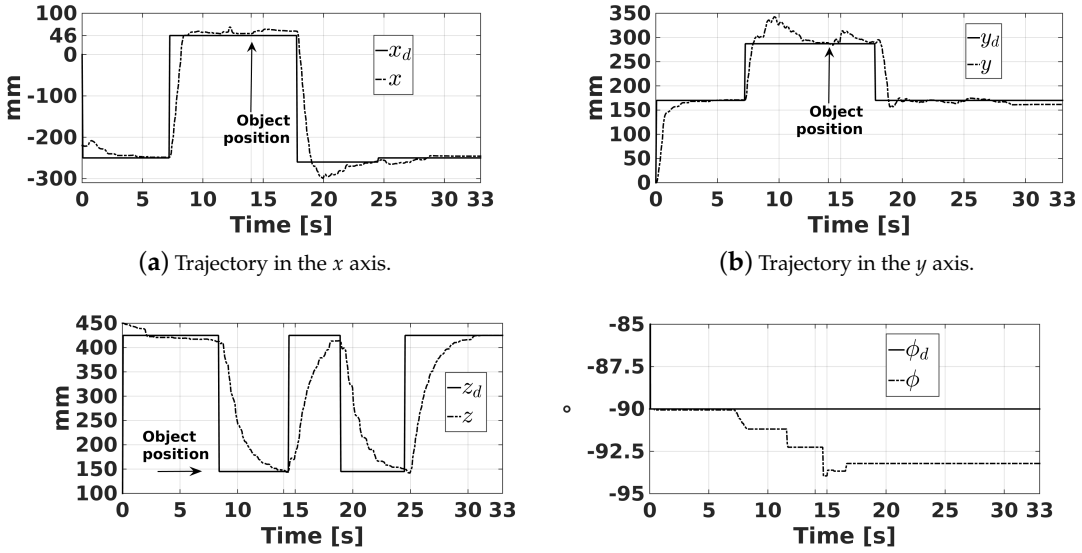
Similarly, the position error of the ith joint is defined as:

$$\tilde{q}_i = q_{di} - q_i, \qquad i = 1, 2, 3, 4 \tag{24}$$

Table 5 shows the position error at the instant when the yellow circle is gripped by the robot. Moreover, this table presents the position errors obtained in the remaining three experiments that consisted in collecting a yellow rectangle, a red circle, and a red rectangle. In this table, it is possible to observe that the maximum error in the Cartesian coordinates is less than 6 mm. Table 5 also shows that the maximum error in the joint space is 2.5°.

Table 6 compares the position accuracy of the proposed recycled platform with respect to two platforms based also on the ED-7220C robot. The first platform, described in [45], contains the manufacturer controller of this robot produced by the ED-Corporation company. The second platform,

called AUTAREP, uses the controller designed in [12]. A shown in this table, the manufacturer and the AUTAREP platforms have better position accuracy than the proposed recycled platform. We attribute this fact to the mechanical wear of the robot, since it has almost twenty years of service and the backlash in its gears has increased. However, the position accuracy of our proposal is appropriate since the robot gripper opens up to 60 mm, and it collects cylinders with a height of 55 mm and diameter of 40 mm. Despite this accuracy, the proposed platform has fulfilled its aim of allowing undergraduate students to experimentally validate the theory seen in robotics and automatic control courses using the low-cost controller of open architecture. Moreover, the proposed platform has a great utility for educational purposes since their high-level programming based on MATLAB-Simulink permits students to design their own controllers in a simple way and to use several toolboxes to acquire, process, and generate signals for verifying the controller performance. Note also that this software could be used to operate the manipulator as a remote platform without attending to the laboratory, thus allowing its use by students with physical disabilities.



(**a**) Trajectory in the x axis.

(**b**) Trajectory in the y axis.

(**c**) Trajectory in the z axis.

(**d**) Angle ϕ of the end-effector.

Figure 18. Trajectory of the end-effector.

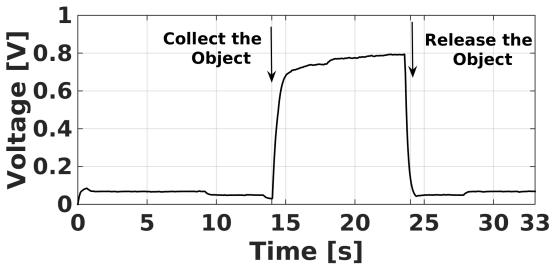
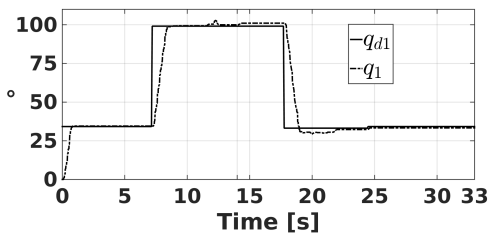
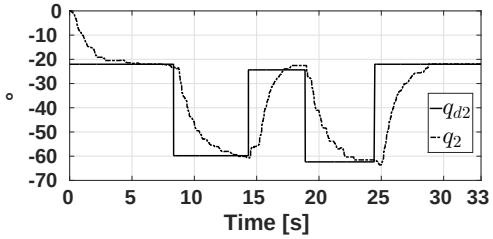


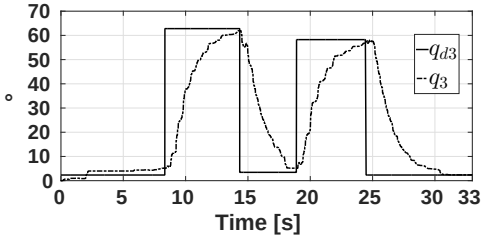
Figure 19. Voltage of the force sensor inside the gripper.



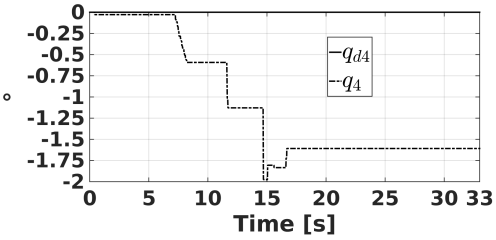
(**a**) Trajectory of q_1.



(**b**) Trajectory of q_2.

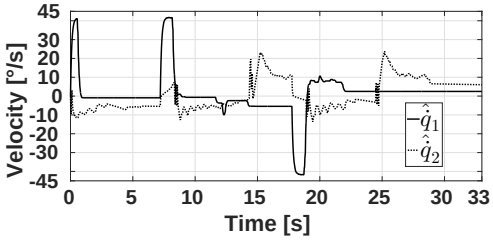


(**c**) Trajectory of q_3.

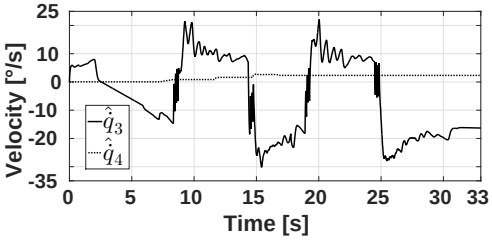


(**d**) Trajectory of q_4.

Figure 20. Trajectories of the joint positions.

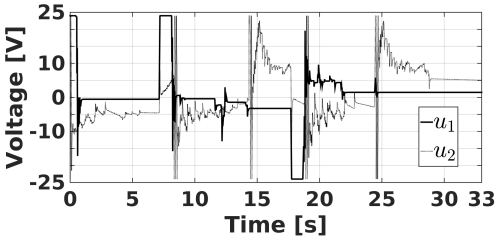


(**a**) Estimates $\hat{\dot{q}}_1$ and $\hat{\dot{q}}_2$.

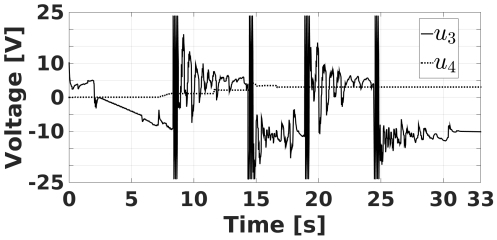


(**b**) Estimates $\hat{\dot{q}}_3$ and $\hat{\dot{q}}_4$

Figure 21. Estimates of the velocity joints provided by the state observers.



(**a**) Control signals u_1 and u_2.



(**b**) Control signals u_3 and u_4.

Figure 22. Control signals of the actuators.

Table 5. Errors at the instant when the object is taken by the robot.

Object	Cartesian Space Position Errors	Joint Space Position Errors
Yellow circle	$\tilde{x} = -5.15$ mm $\tilde{y} = 0.22$ mm $\tilde{z} = -1.72$ mm $\tilde{\phi} = 2.26^\circ$	$\tilde{q}_1 = -1.01^\circ$ $\tilde{q}_2 = 0.40^\circ$ $\tilde{q}_3 = 1.62^\circ$ $\tilde{q}_4 = 1.15^\circ$
Yellow square	$\tilde{x} = 0.09$ mm $\tilde{y} = -5.63$ mm $\tilde{z} = -1.56$ mm $\tilde{\phi} = -0.17^\circ$	$\tilde{q}_1 = -0.43^\circ$ $\tilde{q}_2 = 0.32^\circ$ $\tilde{q}_3 = 1.30^\circ$ $\tilde{q}_4 = -0.11^\circ$
Red circle	$\tilde{x} = 0.37$ mm $\tilde{y} = 0.45$ mm $\tilde{z} = -2.17$ mm $\tilde{\phi} = 2.05^\circ$	$\tilde{q}_1 = 0.04^\circ$ $\tilde{q}_2 = -0.04^\circ$ $\tilde{q}_3 = 1.30^\circ$ $\tilde{q}_4 = 1.04^\circ$
Red square	$\tilde{x} = -1.87$ mm $\tilde{y} = -5.30$ mm $\tilde{z} = -2.81$ mm $\tilde{\phi} = 1.11^\circ$	$\tilde{q}_1 = -0.68^\circ$ $\tilde{q}_2 = -0.14^\circ$ $\tilde{q}_3 = 2.12^\circ$ $\tilde{q}_4 = 0.58^\circ$

Table 6. Robot accuracy of experimental platforms based on the ED-7220C robot manipulator.

Platform	Accuracy
Proposed recycled robot	6 mm
Manufacturer of the ED-7220C robot	0.5 mm
AUTAREP	2 m

11. Conclusions

This article describes a methodology to recycle a 4-DOF educational robot with gripper, which is used for Mechatronics Engineering courses at the University of Colima. Its kinematics, dynamics, and artificial vision are presented along with a proposed low-cost and decentralized controller for the robot joint's actuators. Furthermore, the process of identifying the parameters of the manipulator actuators is also presented, as well as the use of them for designing PID controllers and implementing state observers that estimate the speed of the actuators. Experimental tests in the manipulator were executed through a proposed graphical user interface that allows selecting the shape and color of the objects gripped by the manipulator. Experiments with the proposed controller were successful in avoiding collisions between the robot and the objects collected by it. It was verified that the maximum positioning errors in the Cartesian and joint coordinates of the robot are 6 mm and 2.5°, respectively. As a future work, we will add another DOF to the manipulator to produce the roll motion in its wrist so that it can pick up objects with an irregular geometry. We will also implement adaptable and robust control schemes in the platform, and we will also upgrade the mechanical components of the manipulator to obtain more precise movements. In addition, we plan to recognize 3D objects with the artificial vision of the robot and to use its GUI for remote practices.

Author Contributions: Conceptualization, R.F.-C. and A.C.S.; methodology, S.K.G. and R.F.-C.; software, S.K.G.; validation, A.G.F.-C., J.F.F.-R. and A.C.S.; formal analysis, J.F.F.-R.; investigation, J.F.F.-R.; resources, R.F.-C. and S.K.G; data curation, A.G.F.-C.; writing—original draft preparation, A.C.S.; writing—review and editing, S.K.G.; visualization, J.F.F.-R.; supervision, A.C.S. and A.G.F.-C.; project administration, A.G.F.-C.; funding acquisition, R.F.-C. All authors have read and agreed to the published version of the manuscript.

Funding: All authors acknowledge Programa para el Desarrollo Profesional Docente (PRODEP-SEP) for supporting this research.

Acknowledgments: Authors thank the anonymous reviewers for their valuable comments and suggestions that improve the quality of this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kopacek, B.; Kopacek, P. End of life management of industrial robots. *Elektrotech. Informationstech.* **2013**, *130*, 67–71. [CrossRef]
2. Bomfim, M.; Fagner Coelho, A.; Lima, E.; Gontijo, R. A low cost methodology applied to remanufacturing of robotic manipulators. *Braz. Autom. Congr.* **2014**, *20*, 1506–1513.
3. Sanfilippo, F.; Osen, O.L.; Alaliyat, S. Recycling A Discarded Robotic Arm For Automation Engineering Education. In Proceedings of the 28th European Conference on Modelling and Simulation (ECMS), Brescia, Italy, 27–30 May 2014; pp. 81–86.
4. Soriano, A.; Marin, L.; Valles, M.; Valera, A.; Albertos, P. Low Cost Platform for Automatic Control Education Based on Open Hardware. *IFAC Proc. Vol.* **2014**, *47*, 9044–9050. [CrossRef]
5. Yen, S.H.; Tang, P.C.; Lin, Y.C.; Lin, C.Y. Development of a Virtual Force Sensor for a Low-Cost Collaborative Robot and Applications to Safety Control. *Sensors* **2019**, *19*, 2603. [CrossRef] [PubMed]
6. Qassem, M.A.; Abuhadrous, I.; Elaydi, H. Modeling and Simulation of 5 DOF educational robot arm. In Proceedings of the 2010 2nd International Conference on Advanced Computer Control, Shenyang, China, 27–29 March 2010; Volume 5, pp. 569–574.
7. Rai, N.; Rai, B.; Rai, P. Computer vision approach for controlling educational robotic arm based on object properties. In Proceedings of the 2014 2nd International Conference on Emerging Technology Trends in Electronics, Communication and Networking, Surat, India, 26–27 December 2014; pp. 1–9.
8. Cocota, J.A.N.; Fujita, H.S.; da Silva, I.J. A low-cost robot manipulator for education. In Proceedings of the 2012 Technologies Applied to Electronics Teaching (TAEE), Vigo, Spain, 13–15 June 2012; pp. 164–169.
9. Rivas, D.; Alvarez, M.; Velasco, P.; Mamarandi, J.; Carrillo-Medina, J.L.; Bautista, V.; Galarza, O.; Reyes, P.; Erazo, M.; Pérez, M.; et al. BRACON: Control system for a robotic arm with 6 degrees of freedom for education systems. In Proceedings of the 2015 6th International Conference on Automation, Robotics and Applications (ICARA), Queenstown, New Zealand, 17–19 February 2015; pp. 358–363.
10. Kim, H.S.; Song, J.B. Multi-DOF counterbalance mechanism for a service robot arm. *IEEE/ASME Trans. Mechatron.* **2014**, *19*, 1756–1763. [CrossRef]
11. Manzoor, S.; Islam, R.U.; Khalid, A.; Samad, A.; Iqbal, J. An open-source multi-DOF articulated robotic educational platform for autonomous object manipulation. *Rob. Comput. Integr. Manuf.* **2014**, *30*, 351–362. [CrossRef]
12. Iqbal, U.; Samad, A.; Nissa, Z.; Iqbal, J. Embedded control system for AUTAREP-A novel autonomous articulated robotic educational platform. *Tehnicki Vjesnik Tech. Gazette* **2014**, *21*, 1255–1261.
13. Ajwad, S.A.; Iqbal, U.; Iqbal, J. Hardware realization and PID control of multi-degree of freedom articulated robotic arm. *Mehran Univ. Res. J. Eng. Technol.* **2015**, *34*, 1–12.
14. Iqbal, J.; Ullah, M.I.; Khan, A.A.; Irfan, M. Towards sophisticated control of robotic manipulators: An experimental study on a pseudo-industrial arm. *Strojniški Vestnik J. Mech. Eng.* **2015**, *61*, 465–470. [CrossRef]
15. Ajwad, S.; Adeel, M.; Ullah, M.I.; Iqbal, J. Optimal v/s Robust control: A study and comparison for articulated manipulator. *J. Balkan Tribol. Assoc.* **2016**, *22*, 2460–2466.
16. Baek, J.; Jin, M.; Han, S. A new adaptive sliding-mode control scheme for application to robot manipulators. *IEEE Trans. Ind. Electron.* **2016**, *63*, 3628–3637. [CrossRef]
17. Wang, Y.; Gu, L.; Xu, Y.; Cao, X. Practical tracking control of robot manipulators with continuous fractional-order nonsingular terminal sliding mode. *IEEE Trans. Ind. Electron.* **2016**, *63*, 6194–6204. [CrossRef]
18. Yang, C.; Jiang, Y.; He, W.; Na, J.; Li, Z.; Xu, B. Adaptive parameter estimation and control design for robot manipulators with finite-time convergence. *IEEE Trans. Ind. Electron.* **2018**, *65*, 8112–8123. [CrossRef]
19. Guo, K.; Pan, Y.; Yu, H. Composite learning robot control with friction compensation: A neural network-based approach. *IEEE Trans. Ind. Electron.* **2018**, *66*, 7841–7851. [CrossRef]
20. Chen, D.; Li, S.; Lin, F.; Wu, Q. New Super-Twisting Zeroing Neural-Dynamics Model for Tracking Control of Parallel Robots: A Finite-Time and Robust Solution. *IEEE Trans. Cybern.* **2019**. [CrossRef] [PubMed]

21. Li, W. Predefined-Time Convergent Neural Solution to Cyclical Motion Planning of Redundant Robots Under Physical Constraints. *IEEE Trans. Ind. Electron.* **2019**, 1. [CrossRef]
22. Van, M.; Ge, S.S.; Ren, H. Finite time fault tolerant control for robot manipulators using time delay estimation and continuous nonsingular fast terminal sliding mode control. *IEEE Trans. Cybern.* **2016**, *47*, 1681–1693. [CrossRef]
23. Chen, D.; Zhang, Y.; Li, S. Tracking control of robot manipulators with unknown models: A Jacobian-matrix-adaption method. *IEEE Trans. Ind. Inf.* **2017**, *14*, 3044–3053. [CrossRef]
24. Jin, L.; Li, S.; Luo, X.; Li, Y.; Qin, B. Neural Dynamics for Cooperative Control of Redundant Robot Manipulators. *IEEE Trans. Ind. Inf.* **2018**, *14*, 3812–3821. [CrossRef]
25. Van, M.; Mavrovouniotis, M.; Ge, S.S. An Adaptive Backstepping Nonsingular Fast Terminal Sliding Mode Control for Robust Fault Tolerant Control of Robot Manipulators. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 1448–1458. [CrossRef]
26. Concha, A.; Figueroa-Rodriguez, J.F.; Fuentes-Covarrubias, A.G.; Fuentes-Covarrubias, R. Plataforma experimental de bajo costo para el control desacoplado de un robot manipulador de 5 GDL. *Revista de Tecnologías en Procesos Industriales* **2018**, *2*, 1–11.
27. Giampiero, C. Legacy MATLAB and Simulink Support for Arduino—File Exchange—MATLAB Central, 2016. Available online: https://www.mathworks.com/matlabcentral/fileexchange/32374-legacy-matlab-and-simulink-support-for-arduino (accessed on 28 February 2020).
28. Wilson, C.E.; Sadler, J.P.; Michels, W.J. *Kinematics and Dynamics of Machinery*; Pearson Education: Upper Saddle River, NJ, USA, 2003.
29. Spong, M.W.; Vidyasagar, M. *Robot Dynamics and Control*; John Wiley & Sons: Hoboken, NJ, USA, 2008.
30. Yen, S.H.; Tang, P.C.; Lin, Y.C.; Lin, C.Y. A Sensorless and Low-Gain Brushless DC Motor Controller Using a Simplified Dynamic Force Compensator for Robot Arm Application. *Sensors* **2019**, *19*, 3171. [CrossRef] [PubMed]
31. Ioannou, P.; Fidan, B. *Adaptive Control Tutorial*; SIAM: Philadelphia, PA, USA, 2006.
32. Jia, J.; Zhang, M.; Zang, X.; Zhang, H.; Zhao, J. Dynamic Parameter Identification for a Manipulator with Joint Torque Sensors Based on an Improved Experimental Design. *Sensors* **2019**, *19*, 2248. [CrossRef] [PubMed]
33. Ogata, K. *Modern Control Engineering*, 5th ed.; Pearson: Upper Saddle River, NJ, USA, 2010.
34. Dorf, R.C.; Bishop, R.H. *Modern control systems*; Pearson: Upper Saddle River, NJ, USA, 2011.
35. Nise, N.S. *Control Systems Engineering*; John Wiley & Sons: Hoboken, NJ, USA, 2007.
36. Bousquet-Jette, C.; Achiche, S.; Beaini, D.; Cio, Y.L.K.; Leblond-Ménard, C.; Raison, M. Fast scene analysis using vision and artificial intelligence for object prehension by an assistive robot. *Eng. Appl. Artif. Intell.* **2017**, *63*, 33–44. [CrossRef]
37. Madani, K.; Kachurka, V.; Sabourin, C.; Golovko, V. A soft-computing-based approach to artificial visual attention using human eye-fixation paradigm: Toward a human-like skill in robot vision. *Soft Comput.* **2019**, *23*, 2369–2389. [CrossRef]
38. Madani, K.; Kachurka, V.; Sabourin, C.; Amarger, V.; Golovko, V.; Rossi, L. A human-like visual-attention-based artificial vision system for wildland firefighting assistance. *Appl. Intelli.* **2018**, *48*, 2157–2179. [CrossRef]
39. Yang, G.; Chen, Z.; Li, Y.; Su, Z. Rapid relocation method for mobile robot based on improved ORB-SLAM2 algorithm. *Remote Sens.* **2019**, *11*, 149. [CrossRef]
40. Yang, G.; Yang, J.; Sheng, W.; Junior, F.E.F.; Li, S. Convolutional neural network-based embarrassing situation detection under camera for social robot in smart homes. *Sensors* **2018**, *18*, 1530. [CrossRef]
41. Corke, P. *Robotics, Vision and Control: Fundamental Algorithms in MATLAB®*; Springer: Berlin/Heidelberg, Germany, 2011.
42. Li, X.F.; Shen, R.J.; Liu, P.L.; Tang, Z.C.; He, Y.K. Molecular characters and morphological genetics of CAL gene in Chinese cabbage. *Cell Res.* **2000**, *10*, 29. [CrossRef]
43. Bribiesca, E. Measuring 2-D shape compactness using the contact perimeter. *Comput. Math. Appl.* **1997**, *33*, 1–9. [CrossRef]
44. Montero, R.S.; Bribiesca, E. State of the art of compactness and circularity measures. *Int. Math. Forum* **2009**, *4*, 1305–1335.

45. Corporation, E. ED-7220C Robot Manipulator. 2011. Available online: http://www.adinstruments.es/WebRoot/StoreLES/Shops/62688782/4C61/2F15/726A/B301/6188/C0A8/28BB/86B9/ED_7220C.pdf (accessed 28 January 2020).
46. Concha, A.; Figueroa-Rodríguez, J.F. Robot Software. 2020. Available online: https://github.com/skgadi/UCol-Educational-Robot (accessed on 28 February 2020).