IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 7, 2022, accepted February 22, 2022, date of publication February 28, 2022, date of current version March 11, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3155199

SCARA Self Posture Recognition Using a Monocular Camera

VÍTOR TINOCO[1,2], **MANUEL F. SILVA**[1,3], **FILIPE N. SANTOS**[1], **RAUL MORAIS**[1,2], **AND VÍTOR FILIPE**[2]

[1]INESC TEC—Institute for Systems and Computer Engineering, Technology and Science, 4200-465 Porto, Portugal
[2]Department of Engineering, UTAD—University of Trás-os-Montes and Alto Douro, 5000-801 Vila Real, Portugal
[3]ISEP/IPP—School of Engineering, Polytechnic Institute of Porto, 4200-072 Porto, Portugal

Corresponding author: Vítor Tinoco (vitor.tinoco@inesctec.pt)

This work was supported in part by the National Funds through the Portuguese Funding Agency, FCT—Fundação para a Ciência e a Tecnologia, under Project UIDB/50014/2020; and in part by the European Union's Horizon 2020—The EU Framework Program for Research and Innovation (2014–2020) through DEMETER to Identify the Users Requirements under Grant 857202.

ABSTRACT Robotic manipulators rely on feedback obtained from rotary encoders for control purposes. This article introduces a vision-based feedback system that can be used in an agricultural context, where the shapes and sizes of fruits are uncertain. We aim to mimic a human, using vision and touch as manipulator control feedback. This work explores the use of a fish-eye lens camera to track a SCARA manipulator with coloured markers on its joints for the position estimation with the goal to reduce costs and increase reliability. The Kalman Filter and the Particle Filter are compared and evaluated in terms of accuracy and tracking abilities of the marker's positions. The estimated image coordinates of the markers are converted to world coordinates using planar homography, as the SCARA manipulator has co-planar joints and the coloured markers share the same plane. Three laboratory experiments were conducted to evaluate the system's performance in joint angle estimation of a manipulator. The obtained results are promising, for future cost effective agricultural robotic arms developments. Besides, this work presents solutions and future directions to increase the joint position estimation accuracy.

INDEX TERMS Visual servoing, self posture, monocular camera, marker tracking, SCARA manipulator, planar homography.

I. INTRODUCTION

Robotic manipulators usually rely on data obtained from rotary encoders to determine their joint angles and to relay it to a closed-loop control system. However, rotary encoders have flaws [1], [2], due to effects of elastic joints, joint frictions, flexible links, gearbox backlashes, etc [1]. Furthermore, using an incremental rotary encoder requires an absolute position sensor, such as a hall sensor, to obtain a reference point, and requires the current position saved in memory before shutting down the system. Therefore, on startup, the system will assume the manipulator's joints are in the same angular position as they were on shutdown, which is not necessarily true.

Several applications of vision systems on robotic manipulators, namely for joint pose estimation and calibration, have been proposed in the literature. Balanji *et al.* [3] proposed a vision-based calibration framework for industrial manipulators using ArUco markers. The authors concluded that their system is reliable in real-world applications with millimetric and near 0° errors. Kuo *et al.* [1] proposed a single camera vision-based system for estimating a manipulator joint angles. The authors used visual markers and pointed at the manipulator with a fixed camera. They concluded that the vision system could not compete with the joint encoders; however, it works perfectly as a backup if the encoders fail. Li *et al.* [4] proposed a vision-based system using a monocular camera to estimate planar joint angles. The camera was positioned above the manipulator and pointing down, and it tracked visual markers to determine the joint positions. The authors concluded that this system offered high precision and more usability than other methods, such as laser tracking. Hajiloo *et al.* [5] developed an image-based visual servoing control system for a six degree-of-freedom manipulator with an eye-in-hand camera. The authors successfully attenuated the displacements between the initial and desired position,

The associate editor coordinating the review of this manuscript and approving it for publication was Guilin Yang.

often large in conventional visual servoing controllers. They concluded that their controller increases success by keeping the system within the desired limits. Zhang *et al.* [6] proposed an inversion-free image-based visual servoing system for manipulators with an eye-in-hand camera configuration using neural networks. Their system was theoretically effective at converging feature errors to near-zero values while within the manipulator's velocity and position limits; moreover, the authors propose implementing the proposed system on a physical manipulator. Wang *et al.* [7] developed an adaptive visual servoing system for soft manipulators. Their control system is based on piecewise-constant curvature kinematic and does not require the true values of the manipulator link lengths and the target positions. The authors verified the adaptability of the soft manipulator to the environment in free space, constrained environments, and environments with the influence of gravity. They concluded that the manipulator could be applied to such environments with the developed controller. Xu *et al.* [8] proposed a vision-based method for cable-driven robots to simultaneously measure the manipulator configuration and the target position. The authors used "global cameras" pointing at the manipulator and at the scene. The vision-based system was able to complete a docking maneuver with a 98 % success rate with position errors bellow 2 mm. Xu *et al.* [9] developed a prototype vision-based control system for an excavator manipulator using a monocular camera and visual markers, with the purpose of increasing safety and production. The experimental results showed position errors of 22 mm and orientation errors of 8.5°. The authors concluded that their experimental results proved the effectiveness of the approach for practical applications.

For agricultural purposes, such as harvesting and pruning, a high precision joint angle estimator is not as required as it would be for industrial purposes, where the shapes and sizes of objects are known and constant. Thus, an agricultural manipulator picks fruits or other products with unknown properties - and in some cases in bulk, such as grapes - that are not positioned in a specific place and do not have their properties stored in a database. Given this, or to attenuate the previously mentioned problems, a vision-based joint angle estimation method is proposed using a monocular camera on the manipulator base. This method, known as visual servoing [10], [11], offers a backup in case the encoders fail [1], providing absolute angular positioning. The sensor (camera) can be used by other systems simultaneously. The vision-based system can be considered a reference for incremental encoders to reduce the costs of buying an absolute position encoder, assuming the camera will also be used for other purposes in the system. Furthermore, with visual servoing, an agricultural manipulator can mimic a human being, as humans only use vision and touch to grab objects. This proposed system differs from the ones presented in the literature as the camera is to be placed behind the manipulator with a fish-eye lens to have a greater field of view. This proposed system offers an advantage in an agricultural scenario where the manipulator needs to be moved to different locations, and an onboard camera with an entire field of view of the manipulator at all times is crucial. Using a monocular camera will reduce the processing cost, as fewer frames will be processed, leaving the rest of the processing power for other applications. Moreover, the camera can detect fruits, like tomatoes, so the manipulator knows their position and, through inverse kinematics, knows which joint angles it should have to reach the fruit.

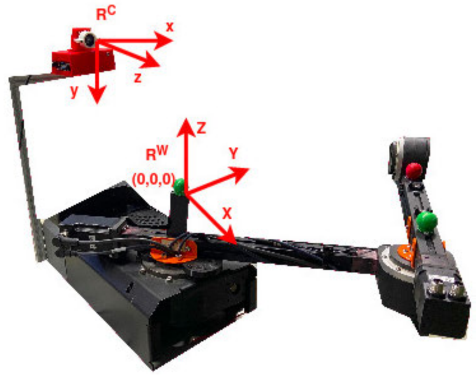

FIGURE 1. SCARA manipulator setup.

This document is organized in the following way: in Section II the experimental setup for this work is presented; in Section III the methodology for the angle estimation is shown; in Section IV the used marker tracking algorithms are presented; in Section V the experimental results and their analysis are shown; and, finally, in Section VI the conclusions to this work are drawn.

II. EXPERIMENTAL SETUP

The manipulator used in this work is a Selective Compliance Articulated Robot Arm (SCARA), presented in Figure 1 along with the camera (R^C) and world (R^W) coordinate references, modified from an existing igus[1] manipulator, with an incremental encoder (with 500 steps) and a hall sensor on each motor. It has co-planar joints, meaning a homography matrix can define the entire manipulator space. Furthermore, it has three links and three coloured co-planar spherical markers: two green markers on the first and second joint and one red marker on the end of the third link. The markers have a diameter of 25 mm and the first and second links have a length of 0.35 cm and 0.26 cm, respectively. Finally, a Raspberry Pi camera is positioned 26 cm behind the first marker and 19 cm above it, using a fish-eye lens to increase the camera field of view on the manipulator. The camera captures images with a resolution of 1920 × 1440 pixels. This will be used to validate if a feedback system, solely based on a monocular camera and markers, is reliable for pruning and harvesting operations.

III. JOINT ANGLE ESTIMATION

A. PLANAR HOMOGRAPHY

The homography matrix translates the relationship between the image plane and, in this case, the visual marker plane.

[1] https://www.igus.com/

This relation can be defined by Equation (1), where **x** is the point in the image plane, **P** is the homography matrix, and **X** is the point in the world frame.

$$\mathrm{x} = \mathrm{PX} \tag{1}$$

Expanding the previous equation, Equation 2 is obtained.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} r11 & r12 & r13 & t1 \\ r21 & r22 & r23 & t2 \\ r31 & r32 & r33 & t3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2}$$

The Z coordinate in the world frame is constant, as the markers are parallel to the Z axis. As such, the coordinate is assumed to be $Z = 0$, and thus, the third column of the translation matrix is multiplied by 0 and can be removed. Given this, equation (1) can be decomposed into Equation (3), where **H** is the homography matrix.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} r11 & r12 & t1 \\ r21 & r22 & t2 \\ r31 & r32 & t3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \mathrm{H} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{3}$$

The image coordinates presented previously can be further decomposed into pixel coordinates using the camera matrix, show in Equation (13), where u and v are the pixel coordinates in the x and y axis, respectively, and f and c are the camera intrinsic parameters.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r11 & r12 & t1 \\ r21 & r22 & t2 \\ r31 & r32 & t3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{4}$$

This work's objective requires the translation of pixel coordinates to world coordinates. This can be achieved with linear algebra by multiplying, on both sides of the previous equation, the inverse of the multiplication of the camera matrix, and the homography matrix, and thus Equation (5) is obtained.

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \left(\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r11 & r12 & t1 \\ r21 & r22 & t2 \\ r31 & r32 & t3 \end{bmatrix} \right)^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{5}$$

B. OBTAINING THE HOMOGRAPHY MATRIX

To obtain the homography matrix, the Raspberry Pi camera was first calibrated using a ChArUco board, shown in Figure 2, using the calibration and arUco classes in the OpenCV[2] libraries. The camera calibration process provided the camera matrix used to calibrate each image frame and calculate the planar homography by transforming the pixel coordinates into image coordinates. Furthermore, each frame was undistorted, with its radial and tangential distortions being attenuated.

As mentioned previously, the colored markers share the same plane (co-planar). Given this, an image of the chessboard parallel to the marker plane, shown in Figure 3, was taken. The OpenCV functions were able to determine the chessboard corners, both in the image frame and the pixel frame. Each corner is then defined by its pixel coordinates, (u, v), and by its image coordinates (x, y). To calculate the homography matrix elements, **h**, Equation (3) can be re-arranged into Equation (6).

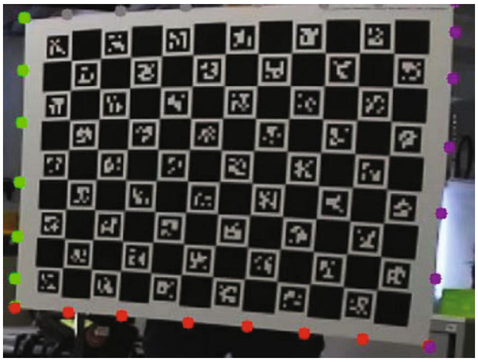

FIGURE 2. ChArUco board.

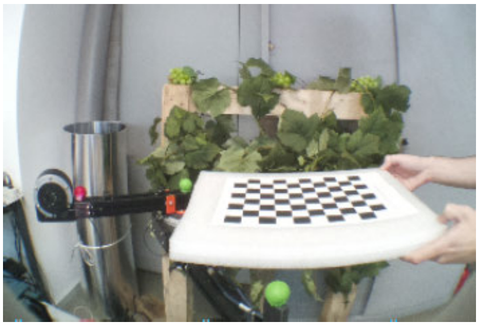

FIGURE 3. Chessboard parallel to markers.

$$\begin{aligned} &\mathrm{Ah} = 0, \\ &\mathrm{A} = \begin{pmatrix} -x & -y & -1 & 0 & 0 & 0 & ux & uy & u \\ 0 & 0 & 0 & -x & -y & -1 & vx & vy & v \end{pmatrix} \\ &\mathrm{h} = \begin{pmatrix} h1 & h2 & h3 & h4 & h5 & h6 & h7 & h8 & h9 \end{pmatrix}^T \end{aligned} \tag{6}$$

Given, at least, four known corresponding points, and defining $h9 = 1$ – since the last element of the **H** matrix is 1 –, the previous equation can be transformed into Equation (7), as shown at the bottom of the next page.

For this work, functions from the OpenCV libraries were used to determine the homography matrix from the obtained chessboard corner points.

C. ANGLE CALCULATION

Once the marker's pixel coordinates are known, their world coordinates can be obtained through planar homography, as they are co-planar. A diagram of the manipulator is presented in Figure 4. In this diagram, the first joint can be defined as the origin, and the two links can be defined as two

[2] https://opencv.org/

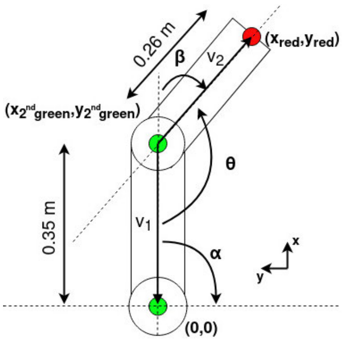

FIGURE 4. **Manipulator diagram.**

vectors v_1 and v_2. Both vectors have their starting point on the second green marker to simplify the following equations.

Before performing any angle calculation, the first joint must be set as the origin. For this, the world coordinates of the second green marker, and the world coordinates of the red marker, are subtracted by the world coordinates of the first green marker (base marker). After this translation, the first joint angle can be calculated using Equation (8), where $(x_{2^{nd}\text{green}}, y_{2^{nd}\text{green}})$ are the second green marker world coordinates.

$$\alpha = \frac{\pi}{2} - \text{atan2}(y_{2^{nd}\text{green}}, x_{2^{nd}\text{green}}) \tag{8}$$

To calculate the second joint angle, the angle θ between vectors v_1 and v_2, both vectors were calculated using Equation (9).

$$\begin{aligned} v_1 &= (0 - x_{2^{nd}\text{green}}, 0 - y_{2^{nd}\text{green}}) \\ v_2 &= (x_{\text{red}} - x_{2^{nd}\text{green}}, y_{\text{red}} - y_{2^{nd}\text{green}}) \end{aligned} \tag{9}$$

To calculate θ, Equation (10) was used.

$$\theta = \text{atan2}(v_{2y}, v_{2x}) - \text{atan2}(v_{1y}, v_{1x}) \tag{10}$$

Finally, the second joint angle, β, is obtained by Equation (11).

$$\beta = \pi - \theta \tag{11}$$

The end-effector's cartesian coordinates, x,y, can be obtained through forward kinematics. In this work, these coordinates are the red marker coordinates obtained through the vision-based feedback system. However, the forward kinematics are used to determine the cartesian ground truth by calculating the end-effectors position using the encoder data. Since the manipulator has two links and moves in a two-dimensional frame, the forward kinematic model can be translated by equation (12), where L_1 and L_2 are the first and second links, respectively.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} L_1\cos(\alpha) + L_2\cos(\alpha+\beta) \\ L_1\sin(\alpha) + L_2\cos(\alpha+\beta) \end{bmatrix} \tag{12}$$

IV. MARKER TRACKING

To introduce robust tracking of coloured markers' position, two estimators, a Kalman Filter and a Particle Filter, were compared. These filters were chosen since they do not require much computational power to track coloured markers. In different scenarios, where coloured markers were not used, a more advanced method, such as neural networks, should be used. The first joint is always static in the cartesian frame. Therefore, this specific marker's position does not need to be estimated with the used filters, and thus, only the green marker and the red marker positions need to be estimated.

A. KALMAN FILTER

The used Kalman Filter has a state $\mathbf{x} = [v, w, V_v, V_w]^T$, where v and w are the pixel coordinates, and V_v and V_w are the pixel velocities. The filter parameter settings consist of a state transition matrix ($\mathbf{A} = \mathbf{I}_4$), a measurement matrix ($\mathbf{H} = \mathbf{I}_2$), a process noise covariance matrix ($\mathbf{Q} = \mathbf{I}_4 \times 1\text{e}^{-1}$) and a measurement noise covariance matrix ($\mathbf{R} = \mathbf{I}_2 \times 1\text{e}^{-1}$). This filter is divided into two states: (*i*) Prediction and (*ii*) Innovation.

$$\text{A} = I_4; \quad H = I_2; \tag{13}$$

In the prediction state, the filter calculates the predicted state vector, $\hat{x}_k^-$, and the predicted error covariance, P_k^-. The predicted state vector is calculated in Equation (14), where **A** is the state transition matrix, **B** is the control input matrix, a_{k-1} is the control vector and w_{k-1} is the process noise.

$$\hat{x}_k^- = \text{A}x_{k-1} + \text{B}a_{k-1} + w_{k-1} \tag{14}$$

In this work, no control input was considered, and thus, Equation (14) can be rewritten into Equation (15).

$$\hat{x}_k^- = \text{A}x_{k-1} + w_{k-1} \tag{15}$$

$$\begin{bmatrix} -x_1 & -y_1 & -1 & 0 & 0 & 0 & u_1x_1 & u_1y_1 & u_1 \\ 0 & 0 & 0 & -x_1 & -y_1 & -1 & v_1x_1 & v_1y_1 & v_1 \\ -x_2 & -y_2 & -1 & 0 & 0 & 0 & u_2x_2 & u_2y_2 & u_2 \\ 0 & 0 & 0 & -x_2 & -y_2 & -1 & v_2x_2 & v_2y_2 & v_2 \\ & & & & . & & & & \\ & & & & . & & & & \\ & & & & . & & & & \\ -x_n & -y_n & -1 & 0 & 0 & 0 & u_nx_n & u_ny_n & u_n \\ 0 & 0 & 0 & -x_n & -y_n & -1 & v_nx_n & v_ny_n & v_n \end{bmatrix} \times \begin{bmatrix} h1 \\ h2 \\ h3 \\ h4 \\ h5 \\ h6 \\ h7 \\ h8 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{7}$$

After each pixel measurement, the filter jumps to the innovation state. In this state, the filter outputs an estimate based on the prediction and the measurement, using a gain that increases, or decreases, depending on the predicted error covariance. If the gain is high, the filter gives more weight to the measurement; if the gain is low, the filter gives more weight to the prediction. The estimated state, $\hat{x}_k^+$, is calculated using Equation (16), where **K** is the Kalman Gain, z_k is the measurement vector and **H** is the measurement matrix.

$$\hat{x}_k^+ = \hat{x}_k^- + \mathrm{K}(z_k - \mathrm{H}\hat{x}_k^-) \tag{16}$$

The Kalman Filter was developed using the OpenCV libraries, which contain a Kalman Filter class. For the measured pixels, each frame was divided into two separate frames. Each of these frames was filtered so that only the red and green colours were available. The program then searched for contours. If the contour was circular and had a minimum area of 900 square pixels, it was a valid contour. The centroid of the contours was calculated, and the resulting pixel coordinate was used as the measurement for the filter.

B. PARTICLE FILTER

In contrast to the Kalman Filter, the Particle Filter only considers a single frame per cycle and does not filter out the colours apart from green and red. This filter initializes by placing several points - or particles - around a predefined pixel with a uniform distribution ranging from -70 to $+70$. The filter reads the hue, saturation and value (HSV) values from the pixel positioned on the particle location (the green and red markers have their HSV range of values predefined). After reading these values, the filter calculates the weight of the particle using a normal curve equation for each of the HSV values, as shown in Equation (17), where H, S and V are the hue, saturation and value values of the pixel positioned at the particle, respectively; μ_H, μ_S and μ_V are the predefined HSV values for the green and red markers; and σ_H, σ_S and σ_V are the standard deviations of the HSV values, defined by the HSV range of the two markers.

$$w_p = \mathrm{e}^{-\frac{1}{2}(\frac{H-\mu_H}{\sigma_H})^2}\,\mathrm{e}^{-\frac{1}{2}(\frac{S-\mu_S}{\sigma_S})^2}\,\mathrm{e}^{-\frac{1}{2}(\frac{V-\mu_V}{\sigma_V})^2} \tag{17}$$

This equation will output a weight value between 0 and 1 per particle. For this work, if the weight is below 0.1, it is negligible. After this process, the particles with less weight are deleted and then resampled around the particles with a higher weight, and the process restarts. The fewer the particles with a high weight, the farther away from each other the particles resample, up to a limited distance. To determine the marker point, all the particles with weights superior to 0.1 have their position multiplied by their weight and are summed together, giving the marker pixel coordinates. The distribution of the particles depends on an uncertainty parameter of the filter. The lower this parameter, the less distributed the particles will be around the higher weighted particles. In this work, the frames were taken with a time difference of seconds in between each image. Given this, the manipulator did not move smoothly between frames; instead, there were jumps of several degrees between each frame. For this filter to work in this scenario, the uncertainty parameter was set high so that the particles are distributed over a large area around the higher weighted particles. Meaning if this marker jumps 10° between frames, the marker will still have particles on it. However, this affects the quality of the predicted points.

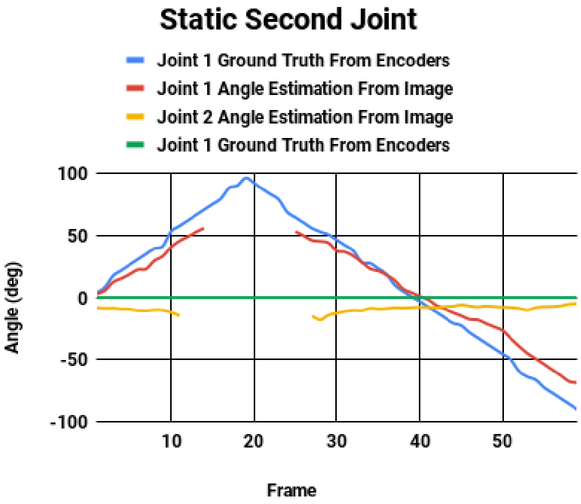

FIGURE 5. Experiment (*i*) - validation of angles from manually annotated points.

V. RESULTS

Three experiences were performed on the manipulator: (*i*) with the first joint rotating from 0 rad to $\frac{\pi}{2}$ rad and to $-\frac{\pi}{2}$ rad and with the second joint static on 0 rad, (*ii-iii*) with the second joint rotating from 0 rad to $\frac{\pi}{2}$ rad and to $-\frac{\pi}{2}$ rad, while the first joint is static on 0 rad, and $-\frac{\pi}{4}$ rad, respectively. Incremental encoders on both joints were used to determine a ground truth. The re-projection error translates how exact the found parameters are. The smaller the value, the more exact the measurements are.

After calibrating the camera, the re-projection error was obtained by calculating the difference between the ChArUco pattern points and their corresponding projected world points. This was done using the OpenCV functions.[3] The average re-projection error was found to be 0.15 pixels.

A. JOINT ANGLE ESTIMATION WITH KNOWN PIXEL COORDINATES

For the first set of experiments, all of the marker pixel coordinates were manually annotated and were used as inputs to the computer vision system. The results of experiments *i-iii* are presented in Figures 5, 6 and 7, respectively.

In Figure 5, there is a gap in the first and second joint estimated angles. This gap happens because the manipulator

[3] https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html

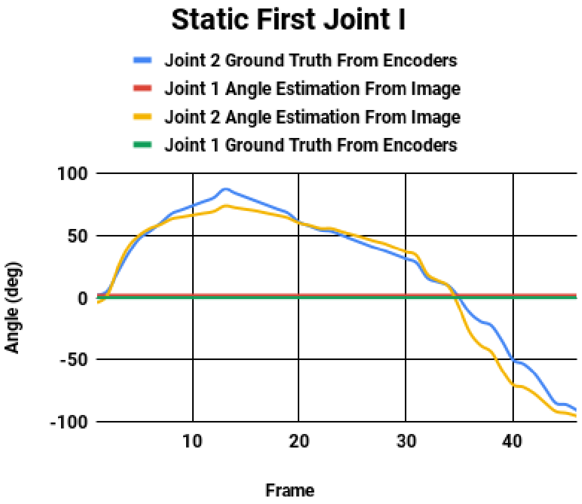

FIGURE 6. Experiment (*ii*) - validation of angles from manually annotated points.

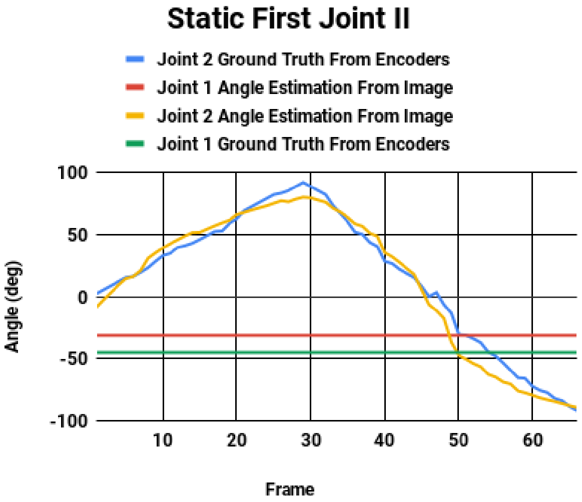

FIGURE 7. Experiment (*iii*) - validation of angles from manually annotated points.

goes out of the camera field of view. Nevertheless, as soon as the markers re-entered the camera field of view, it started detecting the angles again.

Although the calculated angles match the angles from the encoders, there is an error between the calculated data and the ground truth. The mean error and standard deviation for each experiment are presented in Table 1.

As shown in this table, the standard deviation of the first joint for experiments *ii-iii* is 0. This is because the joint remained static the whole time, and the same pixel coordinate was used for all frames. However, in experiment *i*, even though the second joint was static, there is a standard deviation on its error. This happened since, as the first joint rotated, the second joint moved with the first link, and the system had to recalculate its angle with different pixel coordinates.

TABLE 1. Joint angle mean error and standard deviation.

	Joint 1		Joint 2	
Experiment	Mean (^{o})	Std (^{o})	Mean (^{o})	Std (^{o})
i	1.97	11.64	-9.18	2.62
ii	1.99	-	-4.76	8.33
iii	13.64	-	-3.21	8.62

TABLE 2. End-effector position mean error and standard deviation.

Experiment	Mean (mm)	Std (mm)
i	20.72	135.52
ii	0.66	37.77
iii	130.38	39.08

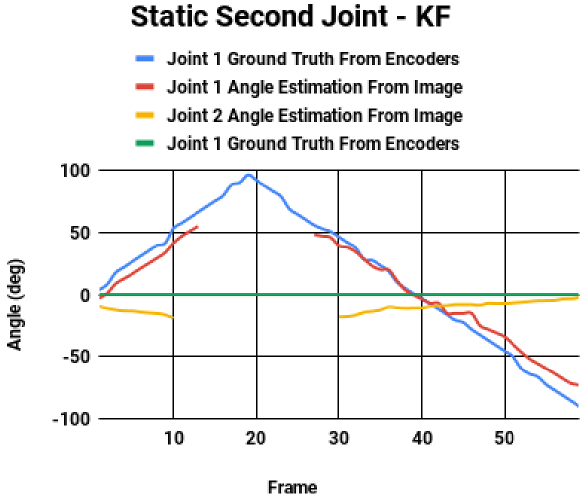

FIGURE 8. Experiment (*i*) - Kalman filter based solution.

Overall, the average error and standard deviation were under 10°.

Using the forward kinematic model, presented previously in (12), and the data presented in Table 1, the position error can be calculated. This error is presented in Table 2.

B. JOINT ANGLE ESTIMATION WITH MARKER TRACKING

The markers were tracked using a Kalman Filter and a Particle Filter on the second set of experiments. Although there are two green markers, the first green marker (base marker) was assumed to always be in the same pixel coordinate for these experiments. This is because the marker is positioned on a joint that does not move in the Cartesian frame; thus, the marker only rotates on the same place at any given time.

The performed experiments are the same as the previous ones.

1) KALMAN FILTER

The results of the angle estimation with automatic marker tracking using a Kalman Filter for the experiments (*i-iii*) are presented in Figures 8, 9 and 10, respectively.

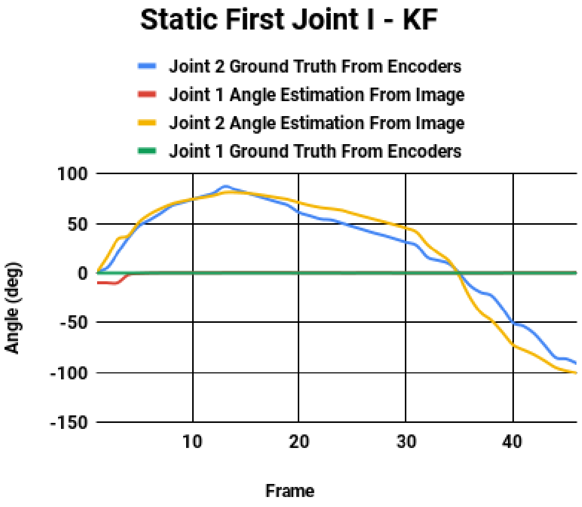

FIGURE 9. Experiment (*ii*) - Kalman filter based solution.

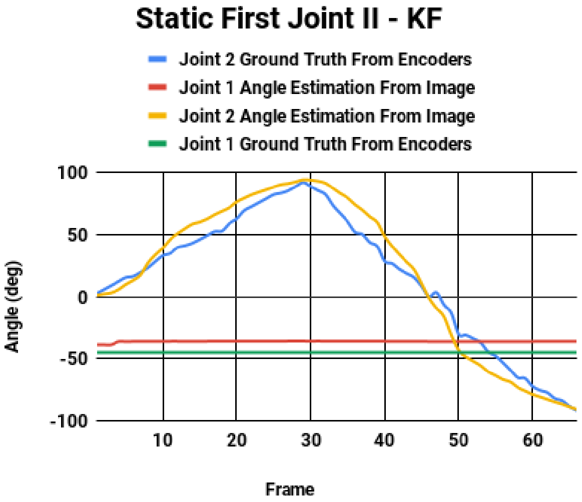

FIGURE 10. Experiment (*iii*) - Kalman filter based solution.

TABLE 3. Joint angle mean error and standard deviation - KF.

	Joint 1		Joint 2	
Experiment	Mean (°)	Std (°)	Mean (°)	Std (°)
i	0.49	8.99	-10.24	4.42
ii	-0.61	2.49	0.40	12.05
iii	8.75	0.59	3.27	11.43

The Kalman Filter tracking performed similarly to the manual marker tracking. Nevertheless, there were moments where the filter only converged after a few frames and not right at the start, such as in Figures 9 and 10. The first joint converged to a stable angle after a few frames in these figures. The mean error and standard deviation for each experiment are presented in Table 3.

TABLE 4. End-effector position mean error and standard deviation - KF.

Experiment	Mean (mm)	Std (mm)
i	41.21	115.55
ii	4.68	80.94
iii	107.83	58.03

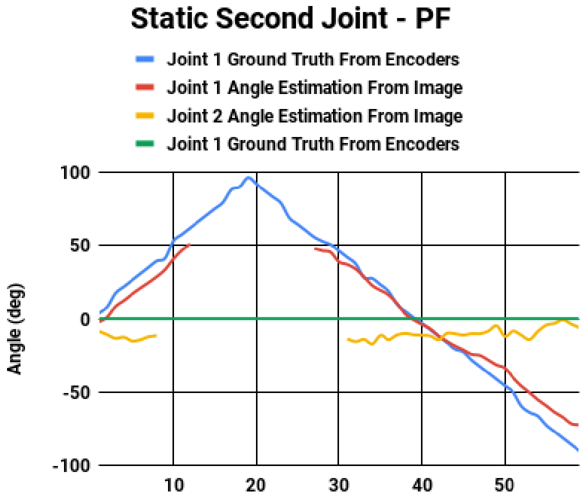

FIGURE 11. Experiment (*i*) - particle filter based solution.

TABLE 5. Joint angle mean error and standard deviation - PF.

	Joint 1		Joint 2	
Experiment	Mean (°)	Std (°)	Mean (°)	Std (°)
i	0.11	8.87	-10.57	3.73
ii	-0.31	0.24	-0.71	14.08
iii	9.11	1.89	0.27	13.15

Relatively to Table 1, this table shows that the mean error on both joints is reduced. Experiments *i* and *ii* have sub-degree mean errors on the first joint, but a mean error of 8.75° on experiment *iii*. On the second joint, experiment *i* shows a higher mean error of $-10.24°$. However, the other experiments show a much reduced and acceptable mean error.

The position error was calculated and is presented in Table 4.

In this table, it is observable that in experiment *iii* the error was 107.83 mm. This error is high; however, on the other experiments, the error was lower. On experiment *i* the mean error was 41.21 mm. This error value is more acceptable for agricultural purposes; however, it needs more refining to become suitable for pruning and harvesting.

2) PARTICLE FILTER

The results of the angle estimation with automatic marker tracking using a Particle Filter for the experiments (*i*-*iii*) are presented in Figures 11, 12 and 13, respectively. Like the Kalman Filter, this filter behaved similarly to the first set of experiments. There is still a constant error present on the second joint of experiment *i* and the first joint of experiment *iii*. The mean error and standard deviation for each experiment are presented in Table 5.

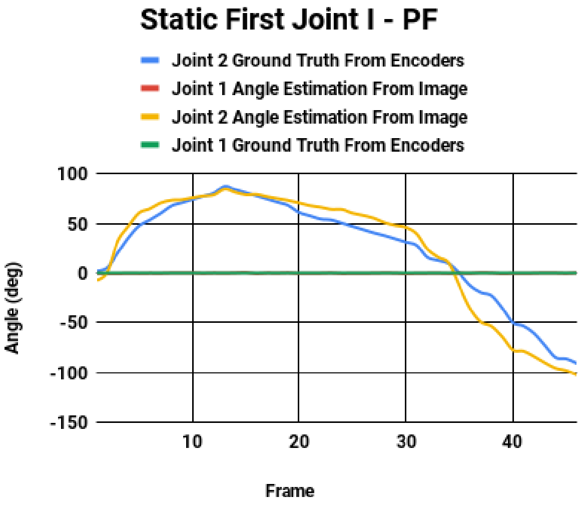

FIGURE 12. Experiment (*ii*) - particle filter based solution.

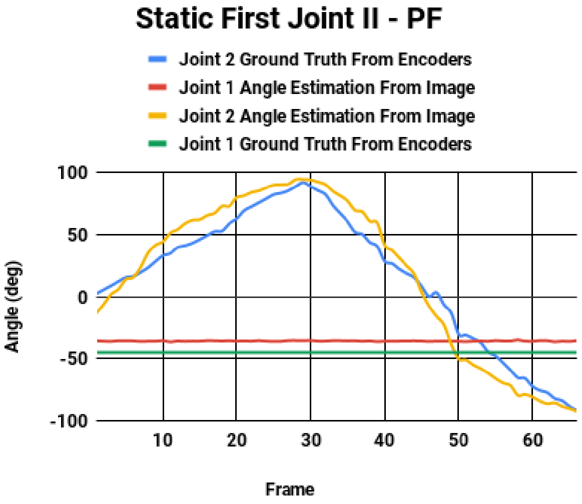

FIGURE 13. Experiment (*iii*) - particle filter based solution.

TABLE 6. End-effector position mean error and standard deviation - PF.

Experiment	Mean (mm)	Std (mm)
i	49.05	111.18
ii	6.52	66.27
iii	98.11	79.53

Comparatively to the Kalman Filter, the errors are very similar with low value on the first joint during experiments *i* and *ii* and a high mean error value on experiment *iii*, and a high mean error value on the second joint on experiment *i* and low mean error values on experiments *ii* and *iii*. The position error was calculated and is presented in Table 6.

As the joint angle errors were similar to the ones of the Kalman Filter, the position errors were also similar with a slight deviation.

VI. CONCLUSION

This work developed a vision-based system to estimate joint angles on a SCARA manipulator. The system, composed of a SCARA manipulator, a raspberry pi camera, a fish-eye lens and coloured markers (painted spheres), was able to estimate the joint angles with some errors. The markers were tracked using a Kalman Filter and a Particle Filter and were subsequently transformed into world coordinates on a plane using planar homography. Given their dimensions, the marker trackers could not precisely detect the centre of the marker. Furthermore, being painted markers, as the manipulator turned, the HSV values of each marker would change slightly, affecting performance, and the homography matrix requires further calibration to obtain better results. The results proved that a system like the proposed one can calculate the joint angles of a co-planar manipulator and thus reduce the number of sensors required. Moreover, the use of the camera allows to obtain a feedback system and thus to develop applications where adjustment based on visual perception represents an added value. Furthermore, the vision-based feedback system can contribute to further advances in robots vision, thus becoming more similar to human vision, since the feedback system from the eyes is used to move the arms and reach objects. The use of this type of feedback can instruct the robot to move the manipulator slightly to the left, to the right, up and down to reach an object, thus not requiring encoders. This innovative procedure can be used, for instance, to adjust the positioning of the arm of the robot to reach and pick fruits and vegetables, randomly placed in a selection conveyor. In this work, there were distinct scenarios where the errors were around 10 cm and could compromise the system in pruning and harvesting operations as the branch dimensions and the dimensions of some fruits are smaller than these errors.

A marker system with high-power coloured light-emitting diodes (LED) is proposed for future work. The use of high-power LED will attenuate the change in colour due to the lighting environment. Moreover, the LED blinking can be synchronized with the camera shutter, reducing power draw and dimming the LED to the human eye. A self-posture calibration system and a relative position error attenuation system are also proposed to reduce errors derived from an ill-calibrated transformation matrix. Using a known reference point, the relative position error attenuation significantly reduces the positioning error and can eliminate the need for a self-posture calibration system.

REFERENCES

[1] Y.-L. Kuo, B.-H. Liu, and C.-Y. Wu, "Pose determination of a robot manipulator based on monocular vision," *IEEE Access*, vol. 4, pp. 8454–8464, 2016.

[2] Z. Yang, S. Li, C. Chen, H. Mei, and Y. Liu, "Reliability prediction of rotary encoder based on multivariate accelerated degradation modeling," *Measurement*, vol. 152, Feb. 2020, Art. no. 107395.

[3] H. M. Balanji, A. E. Turgut, and L. T. Tunc, "A novel vision-based calibration framework for industrial robotic manipulators," *Robot. Computer-Integrated Manuf.*, vol. 73, Feb. 2022, Art. no. 102248.

[4] H. Li, X.-M. Zhang, L. Zeng, and Y.-J. Huang, "A monocular vision system for online pose measurement of a 3RRR planar parallel manipulator," *J. Intell. Robotic Syst.*, vol. 92, no. 1, pp. 3–17, Sep. 2018.

[5] A. Hajiloo, M. Keshmiri, W.-F. Xie, and T.-T. Wang, "Robust online model predictive control for a constrained image-based visual servoing," *IEEE Trans. Ind. Electron.*, vol. 63, no. 4, pp. 2242–2250, Apr. 2016.

[6] Y. Zhang and S. Li, "A neural controller for image-based visual servoing of manipulators with physical constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5419–5429, Nov. 2018.

[7] H. Wang, B. Yang, Y. Liu, W. Chen, X. Liang, and R. Pfeifer, "Visual servoing of soft robot manipulator in constrained environments with an adaptive controller," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 1, pp. 41–50, Feb. 2017.

[8] W. Xu, P. Yan, F. Wang, H. Yuan, and B. Liang, "Vision-based simultaneous measurement of manipulator configuration and target pose for an intelligent cable-driven robot," *Mech. Syst. Signal Process.*, vol. 165, Feb. 2022, Art. no. 108347.

[9] J. Xu and H.-S. Yoon, "Vision-based estimation of excavator manipulator pose for automated grading control," *Autom. Construction*, vol. 98, pp. 122–131, Feb. 2019.

[10] C. J. Taylor and J. P. Ostrowski, "Robust vision-based pose control," in *Proc. Millennium Conf. IEEE Int. Conf. Robot. Automat. Symposia (ICRA)*, Apr. 2000, pp. 2734–2740.

[11] D. Kragic and H. I. Christensen, "Survey on visual servoing for manipulation," *Comput. Vis. Act. Perception Lab., Fiskartorpsv*, vol. 15, p. 2002, Jan. 2002.

VÍTOR TINOCO received the B.Sc. and M.Sc. degrees in electrical and computer engineering from the School of Engineering, Polytechnic Institute of Porto, in 2018 and 2020, respectively. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of Trás-os-Montes e Alto Douro (UTAD). He is also a Researcher with the Institute for System and Computer Engineering, Technology and Science (INESC TEC).

MANUEL F. SILVA was born in April 1970. He received the B.Sc., M.Sc., and Ph.D. degrees in electrical and computer engineering from the Faculty of Engineering, University of Porto, in 1993, 1997, and 2005, respectively. He is currently a Coordinating Professor with the Department of Electrical Engineering, Instituto Superior de Engenharia do Porto (having supervised or co-supervised more than 60 M.Sc. theses and 80 graduation projects/internships) and also a Senior Researcher with the Centre for Robotics in Industry and Intelligent Systems, INESC TEC. He is the author of more than 100 publications in international journals and conferences (https://www.researchgate.net/profile/Manuel-Silva-8) and has been involved in several research and development projects. He has also been actively involved in the organization of several international conferences, integrating the Management Team of the CLAWAR Association (https://clawar.org/) and the General Assembly Board of the Portuguese Robotics Society (of which he has been the President of the Steering Committee). His research interests include modeling, simulation, robotics, bio-inspired robotics, and control and education in robotics and control.

FILIPE N. SANTOS received the degree in electrical and computer engineering from the Instituto Superior de Engenharia do Porto (ISEP), in 2003, the M.Sc. degree in electrical and computer engineering from the Instituto Superior Técnico (IST), Universidade Técnica de Lisboa, in 2007, and the Ph.D. degree in electrical and computer engineering from the Faculdade de Engenharia (FEUP), Universidade do Porto, Portugal, in 2014. Since 2003, he has been a Formal Robotics Researcher with a large experience on navigation systems for unnamed aerial and ground vehicles. He is currently a Senior Researcher and the Head of the Laboratory of Robotics and the IoT for Smart Precision Agriculture and Forestry, Centre for Robotics in Industry and Intelligent Systems (CRIIS), INESC TEC. He has more than 40 peer-reviewed articles in international conferences and international journals. Actually, his priority research fields are robotics for agriculture/forestry, human–robot interaction, perception, and semantic SLAM.

RAUL MORAIS received the B.Sc. degree in electrical engineering (branch of electronics, instrumentation and computer science) from UTAD, in 1993, the M.Sc. degree in industrial electronics from the University of Minho, in 1998, and the Ph.D. degree in electrical and computer engineering (specialty microelectronics) and the Habilitation degree in electrical and computer engineering from UTAD, in 2004 and 2009, respectively. He is currently an Associate Professor (Habilitation) with the Engineering Department—School of Sciences and Technology, University of Trás-os-Montes e Alto Douro (UTAD), Vila Real, Portugal (https://www.utad.pt). His area of expertise sensors and sensor interfaces in CMOS microelectronics, energy harvesting techniques to power small and unattended electronic devices, wireless sensor networks, and the Internet of Things in the context of agriculture/precision viticulture. His research interests include the field of biomedical implantable devices, especially in biotelemetry systems regarding vibration microgenerators to produce electric energy inside smart prosthesis. He is currently a Senior Researcher with the Research and Development Centre, Institute for Systems and Computer Engineering, Technology and Science (INESC TEC) (https://www.inesctec.pt/en) and a Collaborator with the Centre for the Research and Technology of Agro-Environmental and Biological Sciences (CITAB) (http://www.citab.utad.pt/). He is a member of 11 research projects in which four of them are international. He is also a Coordinator of the POCTEP/INTERREG VA Project "SIAPD—Sistema Integrado Transregional de Apoio ao Combate de Pragas e Doenças na Agricultura." His authorship is almost 100 Scopus-indexed papers (H-index of 21). His past & current supervision are 12 Ph.D., 44 M.Sc., and 40 scholarships.

VÍTOR FILIPE received the M.S. degree in informatics from the University of Minho, Portugal, in 1997, and the Ph.D. degree in electrical engineering from the University of Trás-os-Montes e Alto Douro (UTAD), Portugal, in 2003. He is currently an Associate Professor (Habilitation) in electrical engineering with the School of Science and Technology, UTAD. Since 2015, he has been a Senior Researcher with the Institute for System and Computer Engineering, Technology and Science (INESC TEC). He is the coauthor of about 100 papers published in scientific journals, book chapters, and proceedings of peer-reviewed international scientific conferences. His current research interests include computer vision and the application of artificial intelligence in images.