2021 The 21st International Conference on Control, Automation and Systems (ICCAS 2021)
Ramada Plaza Hotel, Jeju, Korea, Oct. 12∼15, 2021

# Pose Estimation of 5-DOF Manipulator using On-Body Markers

Sree S. S. Katta[1], Adnan J.[2], S. Chaudhary[3], S. Dutta Roy[4], Chetan Arora[5], S. K. Saha[6], Magid E.[7]

[1,3,4] Department of Electrical Engineering, (sree.bokka@gmail.com, sumantra@ee.iitd.ac.in),
[2,6]Department of Mechanical Engineering, (adnan.nitb@gmail.com, saha@mech.iitd.ac.in),
[5]Department of Computer Science and Engineering,(chetan@cse.iitd.ac.in),
Indian Institute of Technology Delhi, New Delhi, 110016, India.
[7]Head of Intelligent Robotics Department, (magid@it.kfu.ru),
Institute of Information Technology and Intelligent Systems (ITIS),
Kazan Federal University, Kazan, 420008, Russia.

**Abstract:** Manipulators are helpful in performing various hazardous tasks like sanitization with chemicals in germs infected areas, spraying pesticides in fields, pick and place of heavy and hazardous materials where direct human intervention is difficult. For manipulators to perform its assigned task accurately, prior estimation of its pose needs to be pinpointed. End-effector grasping and arm manipulation require estimation of 3D object poses. Recently, a number of procedures and databases for vision-based estimation of object pose have been advised. However, it is not clear about the performance of the developed algorithms for visual pose estimation of robot manipulation. In this paper we present the pose estimation of a 5-dof PhantomX Reactor Arm using On-Body/Aruco Markers. Forward and inverse kinematics were used to estimate the pose from the position coordinates calculated using computer vision techniques. This paper implements an approach aimed at estimating the pose of a camera, affixed to a robotic manipulator, against a target object. We adopt a single camera single-shot technique that minimizes the reprojection error over all the rigid poses. The simulation and experimental results using industrial monocular camera for different sizes of On-body markers were presented.

**Keywords:** Camera calibration, Aruco Marker, DH-parameters, Homogeneous Transformation Matrix, Kinematics.

## 1. INTRODUCTION

Robot Arm (or manipulators) have a wide range of applications from warehouse automation to manufacturing units. Traditionally, manipulators are calibrated, usually in isolation from other systems, after manufacturing and assembly is complete. To achieve high operational accuracy, methods such as laser tracking or a motion capture system are used to calibrate these robot arms. However, these methods are expensive, stationary and require a constrained environment. Although these might be feasible for large-scale manufacturers to calibrate the manipulators once assembled, these could deviate over time. These deviations can be attributed to wear-and-tear due to extensive usage or other error sources that might accumulate over time. It is beneficial for researchers and smaller organizations to have an alternate method to calibrate their own robot arms which is relatively more accessible and inexpensive.

Pose estimation based on visual sensor data is a key feature [1] in many robotic applications like localization [2], robot navigation, SLAM [3] and others [4]. This paper details a method to calibrate the manipulator using an on-board monocular camera and an on-body marker (ArUco) map[5]. This method provides an alternative for an extensive setup such as a laser tracking system. Also, since the setup is small and portable, we can calibrate the robot arms anywhere (even in the field) otherwise transporting ideally to a constrained environment as is the case with motion tracking systems is difficult [5]. We also demonstrate that it is possible to calibrate manipulators using the same setup or decouple them, depending upon the specific needs of the user and relative to performance of the system. For robotic arm calibration the kinematic parameters of the manipulator in DH parameter representation are used. In this paper, as an example 5-DoF manipulator is used.

This methodology for pose estimation of a 5-DOF manipulator using on-Body markers is explained in two parts:

1. Camera calibration
2. Kinematic identification of the manipulator

This paper is organized as follows: section 2 describes method of camera calibration used, section 3 explains about the on-body(Aruco) markers and their identification using a monocular camera, section 4 defines the kinematics of the 5-dof manipulator using existing methods and the procedure of experimentation and finally section 5 discusses the results and conclusions.

## 2. CAMERA CALIBRATION

The camera used for the experimentation in this work is the Basler ace aCA2440-20gc[6], with lens. The Basler aCA2440-20gc GigE camera with the Sony IMX264 CMOS sensor delivers 23 frames per second at 5.0 MP Resolution [6]. The advantage of using a High Dynamic Range (HDR) camera is the ability to separately adjust the gain of each pixel of the CMOS chip. Basler Lens C23-1224-5M-P [7] f12mm is mounted on the top of the basler camera.

### 2.1 Settings

The minimum working distance required for the basic camera is 200mm. The lens is set at 0.7mm for experiment and the aperture is set at f8 (for better depth of

field).

Using the pylon software, the image size captured by the basler camera with lens is selected to 2430 X 1984 pixels, gain is set at 10dB for a decent contrast and brightness, images are captured at 6.7 Frames Per Second for slow pace of image capturing and Exposure time is also adjusted for avoiding tracking error problems in the detection of corners. With these fixed settings minimum working distance of 600mm is calculated according to the chosen lens from the basler website [8] for a better field of view.

### 2.2 Calibration

Camera Calibration is necessary to map the virtual objects into the real world coordinates, this is generally referred to as mapping of camera coordinates to pixel coordinates. To convert the camera coordinates to pixel coordinates, camera (or intrinsic) parameters are required. The camera parameters include :

scale factor **s** (often equal to 1).

Focal length F (in world coordinates) is the distance between the pinhole of the camera and the image plane.

Principal point $(\boldsymbol{c_x}, \boldsymbol{c_y})$ which is assumed to be at the centre of the image ideally.

Assuming the pinhole camera model, to map a 2D image point $\boldsymbol{P_i}$ into a 3D point $\boldsymbol{P_w}$ the equation (1) needs to be solved.

$$\boldsymbol{s * P_i = C[R|t]P_w} \tag{1}$$

$\boldsymbol{P_i}$ in the 2D image plane is defined by $[\boldsymbol{u} \quad \boldsymbol{v} \quad \boldsymbol{1}]^{\boldsymbol{T}}$
**C** is the Camera (intrinsic) Matrix defined as

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$f_x$ and $f_y$ are the effective focal lengths in x and y direction in pixels
**[R|t]** is a 3 X 4 extrinsic matrix which is

$$\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \tag{3}$$

$(r, t)$ are called the extrinsic parameters where R is rotation matrix and $\mathbf{t} = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^T$ is translation vector. and $\mathbf{P_w}$ is the world coordinate $[X_w \quad Y_w \quad Z_w \quad 1]^T$.

**Distortion coefficients:** Distortion coefficients are required to minimize the image distortions which are caused by factors like optical design of lens, position of camera etc.

$$(k_1 \quad k_2 \quad p_1 \quad p_2 \quad k_3) \tag{4}$$

where $k_1, k_2, k_3$ are the first, second and third order radial distortion coefficients, $p_1, p_2$ are the first and second order tangential distortion coefficients and $r$ is radial displacement of the undistorted point from the principal point.

In this paper, the existing methods to reduce two types of distortions are considered.

**Tangential distortion** generally occurs when the lens and the image plane are not aligned properly. Knowing the distortion coefficients, it can be corrected by equations (5)-(6).

$$x_{corrected} = x + [2p_1xy + p_2(r^2 + 2x^2))] \tag{5}$$

$$y_{corrected} = y + [p_1(r^2 + 2y^2)) + 2p_2xy] \tag{6}$$

**Radial distortion** is point-symmetric at the optical center of the lens and causes an inward or outward shift of image points from their initial perspective projection. By substituting the values of the distortion coefficients in equations (7)-(8), the image can be undistorted.

$$x_{corrected} = x(1 + k_1r^2 + k_2r^4 + k_3r^6) \tag{7}$$

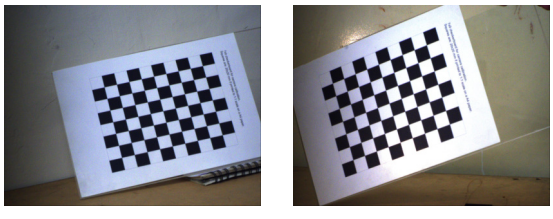
$$y_{corrected} = y(1 + k_1r^2 + k_2r^4 + k_3r^6) \tag{8}$$

Therefore the distortion coefficients are necessarily required to be accurate for decent undistortion/restoration of the disturbed images.

## 3. EXPERIMENTS

### 3.1 Camera calibration using checkerboard images

In this a 7 X 9 checkerboard with each square of 20mm X 20mm size was used for calibrating the camera, obtaining the camera parameters and distortion coefficients [9]. The checkerboard was glued to a thick board and is placed at a distance between 1 to 1.5m from the camera. Python OpenCV was used to determine the parameters for the camera matrix and distortion coefficients. As shown in the Figure 1, 70 images were captured at different perspective angles for a better initial calibration and the calculated camera matrix and distortion coefficients were stored [9]. This is a one-time procedure implemented as the initial setup of the camera for a given camera lens and corresponding settings. Initially the checker-



(a) (b)

Fig. 1.: Checker board images with different perspectives.

board corners are detected from the image. Using the corner coordinates, camera matrix and distortion coefficients are computed. The camera matrix obtained is

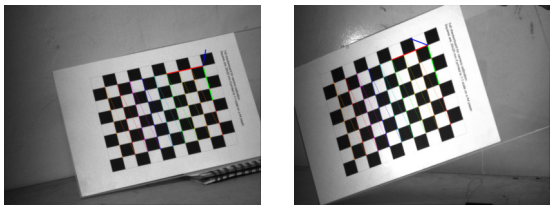
$$\begin{bmatrix} 7.567e+03 & 0.0 & 1.239e+03 \\ 0.0 & 7.590e+03 & 1.023e+03 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \tag{9}$$

and the distortion coefiicients are

$$\begin{matrix} -2.63e-01 & -4.48e+00 & -2.89e-03 \\ -6.08e-04 & 5.13e+01 \end{matrix} \tag{10}$$

The resultant average perview error or the residual error is found to be 0.38 pixels.

Using the computed camera matrix and distortion coefficients, the images captured by basler camera as in figure 1, are corrected or undistorted approximately to the original image as shown in Figure 2. Once the calibration is completed, it is important to verify the accuracy of these parameters. It could be done in two ways, as follows:



(a) (b)

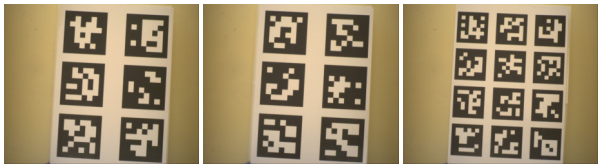
Fig. 2.: Undistorted images of Figure 1a and 1b respectively.

1. Obtain the camera pose with respect to a fiducial (ARUCO) marker and compare it with the ground truth.

2. In many real-life cases, it might be difficult to get accurate distances between the camera origin and the marker. So, considering this use case, two ArUco markers are placed at a fixed distance from each other and the camera pose with respect to both are found. These two camera poses are then used to compute the pose of one marker with respect to each other and hence compared with ground truth. This method should give accurate pose closer to ground truth.

### 3.2 Camera Pose Estimation using On-body (Aruco) Markers

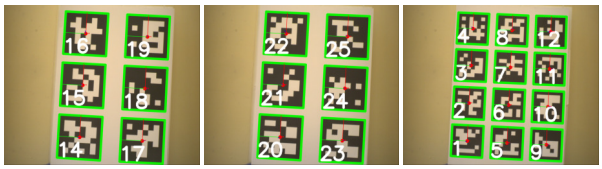
Aruco Marker Map (AMM) consists of Aruco Markers with encoded binary information. Each Aruco marker contains different pixel information stored in different IDs. Python library "aruco.DICT_6X6_250" is used in this work to generate the Aruco images. Each Aruco marker is enclosed by black boxes as borders. Figure 3a is generated using the IDS: 14 to 19, Figure 3b using the IDS: 20 to 25 and Figure 3c using the IDS: 1 to 12. In this experiment the distance between AMM and camera was in the range of 100 to 120cm. The AMM was printed on an A4 thick sheet and was glued to a thick and flat acrylic board. Using python Aruco library "aruco.detectmarkers", computed camera matrix and the distortion coefficients from the initial setup, the corners and ids of each marker are identified from the image. With "aruco.drawDetectedMarkers" python opencv library the identified corners and Ids are displayed as shown in Figure 4.



(a) (b) (c)

Fig. 3.: AMMs of different patterns: no two markers are identical.(a) 8cm X 8cm square size AMM of 2 X 3 grid, (b) 8cm X 8cm square size AMM of 2 X 3 grid and (c) 6cm X 6cm square size AMM of 3 X 3 grid.

In each marker, the "estimatePoseSingleMarkers" Aruco library retrieves the information of corners, rotation matrix **R** and translational vector **t**. Thus from the corner coordinates, the origin of each Marker is calculated. Further, using the equation 1 world coordinates of the image are transformed into camera coordinates. The translation vector **t** can be interpreted as the position of the world origin of each marker in camera coordinates, and the columns of the rotation matrix **R** represent the directions of the world-axes in camera coordinates. If the camera position coordinates are denoted as




(a) (b) (c)

Fig. 4.: Images of detected IDs and corners along with axis orientation embedded in the center of Aruco markers in Figure 3 respectively by the Aruco library.

$$\boldsymbol{P_c} = \begin{bmatrix} \boldsymbol{X_c} & \boldsymbol{Y_c} & \boldsymbol{Z_c} \end{bmatrix}^{\boldsymbol{T}} \tag{11}$$

Origin of the Marker in world axis as

$$\boldsymbol{P_o} = \begin{bmatrix} \boldsymbol{X_i} & \boldsymbol{Y_i} & \boldsymbol{Z_i} \end{bmatrix}^{\boldsymbol{T}} \tag{12}$$

and the orientation (roll, pictch and yaw angles) matrix of the camera as

$$\boldsymbol{R_c} = \begin{bmatrix} \boldsymbol{\alpha_c} & \boldsymbol{\beta_c} & \boldsymbol{\gamma_c} \end{bmatrix}^{\boldsymbol{T}} \tag{13}$$

Therefore the following expressions can be used to find $\boldsymbol{P_c}$ and $\boldsymbol{R_c}$

$$\boldsymbol{P_c} = -\boldsymbol{R^T t} \quad \& \quad \boldsymbol{R_c} = \boldsymbol{R^T} \tag{14}$$

For each marker in the Aruco Marker Map of Figure 4a positions and orientations are as tabulated in table 1. Similarly for each AMM, the image and camera data w.r.t. the world frame is collected and stored in excel sheet using python. The true values of the distances between markers of map are approximately identical both in horizontal and vertical directions. Accurate distances are measured physically using a vernier scale. The results are tabulated in table 2. Similar computations are made for Figures 4b and 4c and the overall results are tabulated in table 3 with measured average horizontal distance, measured average vertical distance, True values, errors and percentage of errors and distance from AMM to camera.

From table 3, it is evident that the errors in bigger size AMM were better than that of the smaller size AMM. Assumptions made in this experiment were that the real distance between camera and the AMMs were between 1 to 1.2m, the markers printed on the map are of true size (error of 0.5mm to 0.7mm) and lens having a circle of least confusion [10].

Table 1.: Image and camera points; orientation angles of camera w.r.to each marker in the first image of Figure 4

| ID | 14 | 15 | 16 | 17 | 18 | 19 |
|---|---|---|---|---|---|---|
| $X_i$ | −4.47 | −3.92 | −3.39 | 7.05 | 7.57 | 8.15 |
| $Y_i$ | 9.16 | −0.42 | −10.04 | 9.77 | 0.21 | −9.44 |
| $Z_i$ | 112.01 | 112.78 | 113.79 | 111.48 | 112.28 | 113.82 |
| $X_c$ | 1.57 | −5.35 | −15.34 | −2.09 | −2.10 | −7.58 |
| $Y_c$ | −10.95 | −11.40 | −5.03 | 14.46 | 15.42 | −2.61 |
| $Z_c$ | 111.92 | 111.45 | 113.15 | 111.17 | 111.45 | 111.22 |
| $\alpha_c$ | 0.07 | 0.05 | 0.05 | 0.09 | 0.01 | −0.02 |
| $\beta_c$ | −0.06 | −0.06 | −0.01 | 0.07 | 0.07 | −0.09 |
| $\gamma_c$ | 1.51 | 1.51 | 1.51 | 1.52 | 1.52 | 1.52 |

Table 2.: Distance between the origin of Markers in the Figure 4 a. both in horizontal and vertical axis.

| Horizontal distance in cms | | | |
|---|---|---|---|
| id-id | 18-19 | 15-18 | 14-17 |
| distance | 11.56 | 11.52 | 11.54 |
| Average = 11.35 | | True Value = 11.35 | |
| Error = 0.19 | | %Error = 1.64 | |
| Vertical distance in cms | | | |
| 16-15 | 15-14 | 19-18 | 14-17 |
| 9.64 | 9.59 | 9.67 | 9.58 |
| Average = 9.621 | | True Value = 9.45 | |
| Error = 0.17 | | %Error = 1.81 | |

## 4. KINEMATIC IDENTIFICATION OF THE MANIPULATOR

Manipulators can perform an array of complex tasks such as pick-and-place objects or assemble products in a warehouse. Many of these tasks are becoming automated, which require the robot to know the end-effector transformation with respect to its base. The origin is at the base of the manipulator, which is stationary or fixed as shown in figure 5. The end-effector transformation with respect to the base, can be computed using the popular DH parameter representation. For each $i^{th}$ joint, the DH parameters are the joint offset ($b_i$), link length ($a_i$), joint angle ($\theta_i$) and twist angle($\alpha_i$). The joint angle is variable (fixed with respect to the home position). The DH parameters of a PhantomX Reactor robotic arm (Figure 5) are computed, and the kinematics of manipulator are modelled [12]. The robot has 5 DOFs, and all the five joints are revolute joints.

### 4.1 Forward Kinematics

Using the Homogeneous Transformation Matrix(HTM) formulation in [12], HTM of the 5-DOF reactor arm is obtained. The resultant forward kinematics for position is given by

$$T = T_1 * T_2 * T_3 * T_4 * T_5 \quad (15)$$

Table 3.: Results from the AMMs in figure 4 are tabulated here. All the measured values, true values and errors are in cms.

(a)

| Readings | Horizontal | Vertical | Z-distance |
|---|---|---|---|
| Measured | 11.5357 | 9.625 | 112.649 |
| True value | 11.35 | 9.45 | 112.343 |
| Error | 0.18569 | 0.17089 | 0.35078 |
| %Error | 1.636 | 1.808 | 0.312 |

(b)

| Readings | Horizontal | Vertical | Z-distance |
|---|---|---|---|
| Measured | 11.5224 | 9.5997 | 112.639 |
| True value | 11.34 | 9.58 | 112.414 |
| Error | 0.1824 | 0.0197 | 0.226 |
| %Error | 1.61 | 0.21 | 0.20 |

(c)

| Readings | Horizontal | Vertical | Z-distance |
|---|---|---|---|
| distance | 7.501 | 7.164 | 116.046 |
| True value | 7.18 | 6.86 | 115.144 |
| Error | 0.321 | 0.304 | 0.902 |
| %Error | 4.476 | 4.425 | 0.78 |

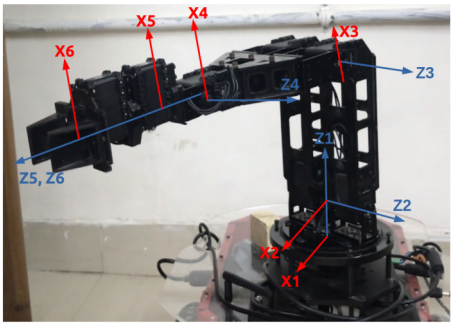



Fig. 5.: PhantomX Reactor Arm at home position.

which yields

$$P_x = 3.34s_1 + 4.71s_{234}c_1 + 14.52c_1c_2 + 14.37c_1c_{23} \quad (16)$$

$$P_y = 4.71s_1s_{234} + 14.52s_1c_2 + 14.37s_1c_{23} - 3.34c_1 \quad (17)$$

$$P_z = 14.52s_2 + 14.37s_{23} - 4.71c_{234} + 3.82 \quad (18)$$

This point $\mathbf{P} = [P_x \quad P_y \quad P_z]^T$ are the coordinates of the manipulator's end effector w.r.t. the world frame. If the camera is placed above the gripper, the camera pose will reflect the end-effector position for the manipulator (since the transformation between the end-effector and camera origin is rigid). Hence any constrained movement of the manipulator will be reflected by the change of camera pose. With respect to a fixed world frame, we can trace the manipulator's movement by tracking the camera poses in the same reference frame. Thus $\mathbf{P}$ and $\mathbf{P_c}$ should be identical. From this, the error between $\mathbf{P}$ and the camera coordinates $\mathbf{P_c}$ are calculated and the experiment is verified.

After this verification of the 3D-points from the image,

Table 4.: DH parameters of PhantomX Reactor Arm lengths measured in cm and angles measured in degrees.

| Link | $b_i$ | $\alpha_i$ | $a_i$ | $\theta_i$ |
|---|---|---|---|---|
| 1 | $b_1$=3.82 | 90 | 0 | $\theta_1$ |
| 2 | $b_2$=3.34 | 0 | $a_2$=14.52 | $\theta_2$ |
| 3 | 0 | 0 | $a_3$=14.37 | $\theta_3$ |
| 4 | 0 | 90 | 0 | $\theta_4$ |
| 5 | $b_5$=4.71 | 0 | 0 | $\theta_5$ |

$b_i$ is Joint offset, $\alpha_i$ is Twist angle, $a_i$ is Link length and $\theta_i$ is the Joint angle.

the pose of manipulator and mobile robot can be modelled using inverse kinematics.

### 4.2 Inverse Position Analysis

The manipulator pose is identified, if all the set of joint variables can be calculated using inverse kinematics corresponding to a given end-effector location and orientation. This 5 dof reactor arm has a closed form of solution and no arbitrary orientation. Algebraic approach is used for deriving closed form solutions.

The coordinate transformation given by Eq. 15 can be rearranged as

$$\begin{bmatrix} \mathbf{P}_b \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{Q} & \mathbf{O}_c \\ \mathbf{O}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{P}_c \\ 1 \end{bmatrix} \tag{19}$$

where $\mathbf{O} = [0, 0, 0]^T$ is a three dimensional vector (column) of zeros.

$$\mathbf{Q} = \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix} \tag{20}$$

where

$$\begin{aligned} q_{11} &= s_1 s_5 + c_1 c_5 c_{234} \\ q_{12} &= s_1 c_5 - s_5 c_1 c_{234} \\ q_{13} &= s_{234} c_1 \\ q_{21} &= s_1 c_5 c_{234} - s_5 c_1 \\ q_{22} &= -s_1 s_5 c_{234} - c_1 c_5 \\ q_{23} &= s_1 s_{234} \\ q_{31} &= s_{234} c_5 \\ q_{32} &= -s_5 s_{234} \\ q_{33} &= -c_{234} \end{aligned}$$

Using algebraic approach, joint angles ($\theta_1$, $\theta_2$, $\theta_3$, $\theta_4$ and $\theta_5$) can be computed using the above equations (15) to (20).

### Simulations

The pose of PhantomX Reactor Arm is visualized in both python and RoboAnalyzer software [13] using both inverse and forward kinematics.

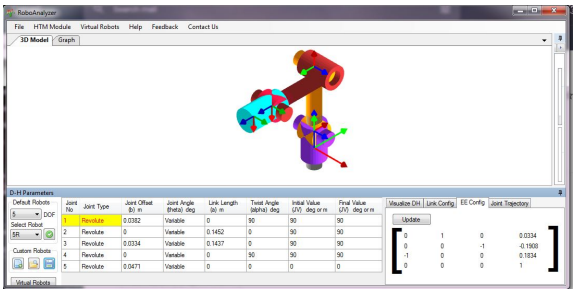


Fig. 6.: Forward kinematics in RoboAnalyzer software

To verify the inverse kinematic equations a known pose (Home) of the end effector (EE) is assumed. Using RoboAnalyzer and forward kinematics, the EE location and orientation is computed from the evaluated HTM as shown in Figure 6. Using this data, the inverse kinematic equations are solved for the pose. Solving the above equations results in multiple solutions. Based on the joint and link constraints and other factors a suitable solution should be chosen. In this example we check the error in distance between the actual and computed locations. From the four sets of solutions obtained two sets of different poses match the desired location (verified through python simulations).

Actual points $P_x$, $P_y$ and $P_z$ for the pose in Figure.6 is:
$\begin{bmatrix} 0.0334 & -0.1908 & 0.1834 \end{bmatrix}$

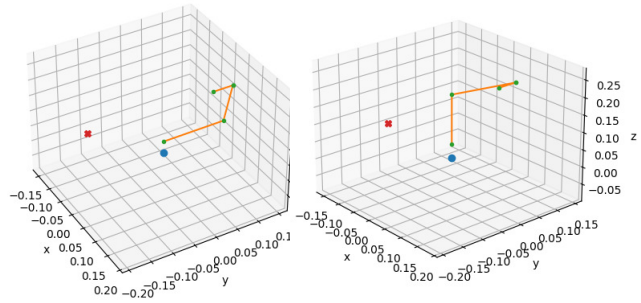
Upon solving the inverse kinematics in Python (Ver 3.8) four sets of solutions and respective EE points are obtained. The 3d plot shows the calculated pose and the distance between the actual EE point and the calculated EE point.

Table 5.: DH parameters of PhantomX Reactor Arm lengths measured in cm and angles measured in degrees.

| solution | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $\theta_1$ | 90.00 | 90.00 | -70.14 | -70.14 |
| $\theta_2$ | 0.59 | 90.00 | 0.59 | -180.59 |
| $\theta_3$ | 90.00 | -90.00 | 90.00 | -90.00 |
| $\theta_4$ | -180.59 | -90.00 | -0.59 | 90.00 |
| $\theta_5$ | 0.00 | 0.00 | -180.00 | -180.00 |
| x | 0.0334 | 0.0334 | 0.0334 | 0.0334 |
| y | 0.0966 | 0.0966 | -0.1908 | -0.1908 |
| z | 0.1834 | 0.1834 | 0.1834 | 0.1834 |
| error between actual and calculated | 0.2874 | 0.2874 | 2.082e-17 | 1.388e-17 |

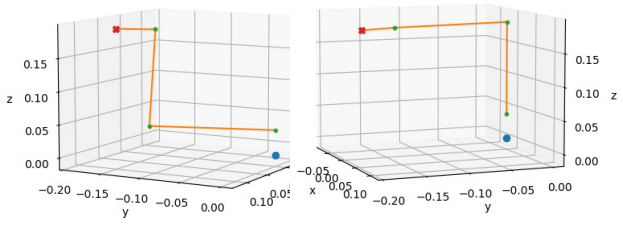
We obtained two solutions with different poses reaching the desired EE position as shown in Figure 8a and 8b. But set 4 is considered as the desired pose when compared to set 3 as the links are closer to the base platform.

From this simulation exercise, it is evident that one of



(a) (b)

Fig. 7.: The pose of solution 1 in(a) and solution 2 in (b) and actual EE points marked in RED. Origin is marked in



(a) (b)

Fig. 8.: The pose of solution 3 in (a) and solution 4 in (b) and actual EE points marked in RED. Origin is marked in BLUE and each joint is marked in GREEN.

the solutions from the inverse kinematics gives the desired pose.

## 5. CONCLUSIONS

In this paper we have developed and implemented the calibration algorithms for pose estimation of PhantomX Reactor arm using On-body markers. It is observed that the Aruco markers of medium size give better results than much bigger and smaller sizes where we found tracking error problems. The mobile manipulator has been setup and pose estimation using inverse kinematics for the manipulator has been verified. Field experiments using mobile manipulator could not be conducted due to covid pandemic, however, numerical calculations and simulations using RoboAnalyzer software and Python programming were done for the manipulator.

## ACKNOWLEDGEMENTS

We acknowledge the Programme in Autonomous Robotics (PAR) Lab, IIT Delhi and I-Hub Foundation for Cobotics (IHFC), Technology Innovation Hub (TIH) at IITD for supporting us with the necessary equipment and experimental space. We thank Dr.Venkat Bokka, Research Scientist, IHFC for his support during experimental setup. This work is supported by Indo-Russia Joint Research Grant from Department of Science and Technology (DST No: INT/RUS/RFBR/P-352 dated $4^{th}$ Oct, 2018).

## REFERENCES

[1] A. Khazetdinov, A. Zakiev, T. Tsoy, M. Svinin and E. Magid, "Embedded ArUco: a novel approach for high precision UAV landing," 2021 International Siberian Conference on Control and Communications (SIBCON), 2021, pp. 1-6, doi: 10.1109/SIBCON50419.2021.9438855.

[2] A. Babinec, L. Jurišica, P. Hubinsky and F. Duchon, "Visual localization of mobile robot using artificial markers", Procedia Engineering, vol. 96, pp. 1-9, 2014.

[3] E. Mingachev, R. Lavrenov, T. Tsoy, F. Matsuno, M. Svinin, J. Suthakorn, et al., "Comparison of ros-based monocular visual slam methods: Dso ldso orb-slam2 and dynaslam", International Conference on Interactive Collaborative Robotics, pp. 222-233, 2020.

[4] S. S. Tordal and G. Hovland, "Relative vessel motion tracking using sensor fusion aruco markers and mru sensors", MODELING IDENTIFICATION AND CONTROL, vol. 38, no. 2, pp. 79-93, 2017.

[5] A. Zakiev, T. Tsoy, K. Shabalina, E. Magid and S. K. Saha, "Virtual Experiments on ArUco and AprilTag Systems Comparison for Fiducial Marker Rotation Resistance under Noisy Sensory Data," 2020 International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1-6, *doi: 10.1109/IJCNN48605.2020.9207701.*

[6] *https://www.baslerweb.com/en/products/cameras /area-scan-cameras/ace/aca2440-20gc/* accessed on 3-5-2021.

[7] *https://docs.baslerweb.com/c23-1224-5m-p* acessed on 3-5-2021.

[8] *https://www.baslerweb.com/en/products/tools/lens-selector/* accessed on 4-5-2021.

[9] Z. Zhang, "A flexible new technique for camera calibration," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1330-1334, Nov. 2000, doi: 10.1109/34.888718.

[10] *https://en.wikipedia.org/wiki/Circle_of_confusion #CITEREFRay2002* accessed on 4-6-2021.

[11] R. A. Boby, S. K. Saha(2016),Single Image based Camera Calibration and Pose Estimation of the End-effector of a Robot,IEEE International Conference on Robotics and Automation (ICRA) Stockholm, Sweden, May 16-21.

[12] Saha, S.K. (Author), 2014, Introduction to Robotics, 2e, Tata McGraw-Hill, New Delhi [ISBN 9789332902817 (599 pages)].

[13] *http://www.roboanalyzer.com/virtual-experiments.html* accessed on 15-6-2021