

# Improving Vision Based Pose Estimation Using LSTM Neural Networks

Diyar Khalis Bilal, Mustafa Unel, Lutfi Taner Tunc

Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul, Turkey

Integrated Manufacturing Technologies Research and Application Center, Sabanci University, Istanbul, Turkey

{diyarbilal, munel, ttunc}@sabanciuniv.edu

**Abstract**—This paper deals with the development of a machine vision based pose estimation system for industrial robots and improving accuracy of the estimated pose using Long Short Term Memory (LSTM) neural networks. To this end, a target object trackable with a monocular camera with  $\pm 90^\circ$  in all directions was designed and fitted with fiducial markers. The designed placement of these fiducial markers guarantees the detection of at least two non-planar markers thus preventing ambiguities in pose estimation. Moreover, a LSTM network is proposed in order to improve the accuracy obtained from the Levenberg-Marquardt (LM) based pose estimation algorithm during trajectory tracking of the robot's end effector. The proposed method utilizes a LSTM network to extract dynamic features from the pose estimated by the LM algorithm and then feeding it to a regression layer to estimate the correct pose. The effectiveness of the proposed method is validated by an experimental study performed using a KUKA KR240 R2900 ultra robot while following sixteen distinct trajectories based on ISO 9238. The obtained results show that the proposed method significantly improves the pose estimation accuracy and precision of the vision based system during trajectory tracking of industrial robots' end effector.

**Index Terms**—Machine Vision, Pose Estimation, Trajectory Tracking, Industrial Robots, Machine Learning, LSTM

## I. INTRODUCTION

Robots are expected to be the standard for use in machining processes in the coming years due to their high degree of automation, large working space and lower prices when compared with conventional CNC machines. However, due to their relatively lower accuracies and stiffness, they are not being used in high precision applications. Based on the aerospace process specifications, the required accuracy for robotic manufacturing is around  $\pm 0.20$  mm, but the accuracies obtained in reality are around 1 mm [1]. Therefore, the robot's relatively low accuracy is the main obstacle in their usage in high precision manufacturing.

In literature there exist many works for increasing the accuracy of industrial robots through utilization of secondary high accuracy encoders installed at each joint, static calibration or dynamic pose correction [2], [3]. However, installation of these encoders is very expensive and not always feasible, moreover the static calibration methods do not consider disturbances acting on the robots due to interactions with their environment. Hence, continuous tool path tracking and dynamic pose correction in real time become necessary for industrial robots to achieve the desired accuracies. This can be achieved by visual

servoing, which utilizes visual feedback and various control strategies to correct the pose of the robot's end effector in real time [4], [5]. Visual servoing assumes the availability of a highly precise visual sensor in its feedback loop and many works in literature utilize laser trackers or photogrammetry sensors for use [6]–[9]. However, all these works in literature utilizing visual servoing rely on the availability of a highly accurate external measurement system such as a laser or dynamic photogrammetry tracker, however these trackers are sometimes even more expensive than the industrial robot itself. Therefore, many works in literature have used an alternative and relatively cheaper approach which is the monocular camera based systems. In the work by Nissler et al. [10], the authors proposed the usage of planar AprilTag markers and attached it to the end effector of the robot. Then, through the usage of optimization techniques they were able to reduce the positioning errors to less than 10 mm. However, they do not consider the rank deficiency problem when using only planar targets and the obtained accuracies were not evaluated during trajectory tracking. Claes et al. [11] proposed a method based on structured light where a monocular camera was attached to the end effector of a robot and the structured light was projected on the workpiece. In their application they evaluated it only for robot positioning and achieved an accuracy of 3 mm, without performing any trajectory tracking. Besides these, Liu et al. [12] proposed two data fusion approaches based on Kalman filter (KF) and multi sensor optimal information algorithms (MOIFA) to fuse position data obtained from a four camera based photogrammetry system and orientation data acquired from a digital inclinometer. Their work was validated on a KP 5 Arc Kuka robot's end effector moving to seventy six points in a one meter cube space and staying there for seven seconds. However, they did not perform trajectory tracking and did not report orientation errors.

From these works in literature it is observed that, in general, either the kinematics/dynamics of the industrial robot must be known in the proposed eye in hand approaches, or in the case of KF type methods, the process and measurement noise along with a linear dynamic process model are assumed to be known. However, it is well known that industrial robotic systems and camera based pose estimation are nonlinear processes. In order to overcome these shortcomings, some work has been done using extended Kalman filter (EKF) [13], [14], and adaptive Kalman filter (AKF) [15] to estimate the pose of

industrial robots. However, EKF depends on the availability of an accurate dynamic process model, which is hard to obtain, and the proposed AKF by G. E. D’Errico [16] does not take into account the time varying process and measurement noise due to robots’ various trajectories and speeds during operation; thus effectively decreasing their usefulness. In such cases data driven modeling techniques have been found to be more effective since the acquired data already contains all kinds of uncertainties, sensor errors and sensor noise. Moreover, machine learning provides some of the most effective data based modeling techniques [17]–[20].

In this work, an eye to hand camera based pose estimation system is developed for industrial robots through which a target object trackable with a monocular camera with  $\pm 90^\circ$  in all directions is designed. The designed camera target (CT) is fitted with fiducial markers where their placement guarantees the detection of at least two non-planar markers from a single frame, thus preventing ambiguities in pose estimation. Moreover, a Long Short Term Memory (LSTM) type recurrent neural network (RNN) [21] is proposed for improving the pose estimated by the Levenberg Marquardt (LM) based algorithm [22]. The proposed method uses a LSTM network to extract dynamic features from the estimated pose and then feeding it to a fully connected regression layer to estimate the correct pose, where the ground truth for training the proposed network is obtained from a laser tracker. Using the proposed method, one can train all the camera based systems using a single laser tracker in a factory where several industrial robots are required to perform the same task, instead of purchasing a laser tracker for each robot.

The rest of the paper is structured as follows: The experimental setup is presented in Section II where the construction of the camera target, coordinate systems, transformations, and the unknowns are described. In Section III, a method for improving pose estimation using LSTM is presented. The effectiveness of the proposed approach is validated by an experimental study in Section IV, followed by a reasoned conclusion in Section V.

## II. EXPERIMENTAL SETUP

### A. Construction of the Camera Target for Pose Estimation

The experimental setup used in this work consisted of a KR240 R2900 ultra KUKA robot, a Leica AT960 laser tracker and a Basler acA2040-120um camera as shown in Figure 1. The pose of the KUKA KR240-2900 robot’s end effector is tracked in real time using the Leica AT960 laser tracker with an accuracy of  $\pm 10$  micrometers through the usage of T-MAC probe rigidly attached to the end effector. In order to estimate the pose of the end effector from the camera, a target object with markers was designed and rigidly fixed to the end effector of the robot. Design and distribution of markers on a target is crucial for the proper estimation of the camera’s pose from images. That is because pose estimation algorithms rely on knowing the exact location of markers in the image plane. Therefore, it is essential to design targets that can be tracked accurately in real time in a robust manner. To this end, this

work proposes the usage of fiducial markers. These markers can be generated from ArUco library which is used for the creation of markers that can be detected and decoded in real time. The patterns known as ArUco markers are small 2D barcodes often used in augmented reality and robotics [23].



Fig. 1: Experimental Setup.

The target object to be tracked by the camera was constructed using 3D printing to hold 40 ArUco markers on it. The camera target (CT) was designed to have 5 faces and each to hold 8 markers. Four of the markers are planar and the other four are placed at  $60^\circ$  with the horizontal axis to produce nonplanar markers. This is because extensive works in literature have shown that points extracted from a single plane may result in ambiguities in pose estimation algorithms. It has been proven that if points extracted from at least two distinct non-parallel planes are used for pose estimation, then they can provide a unique solution. The overall size of the target object is  $250 \times 234 \times 250 \text{ mm}$  with a weight of  $500 \text{ gr}$ . The size of each marker was chosen to be  $30 \text{ mm}^2$ . The target object was made to be modular and only its base is permanently mounted on the robot. The 40 markers were chosen from ArUco’s  $4 \times 4 \times 100$  library and they were fixed to the holes in the constructed target object. The 5 sides of the target object along with the used markers and their ID’s are shown in Figure 2. Through the usage of this camera target, the locations of all the markers can be obtained in the object frame from the CAD model and thus they can be used for pose estimation.

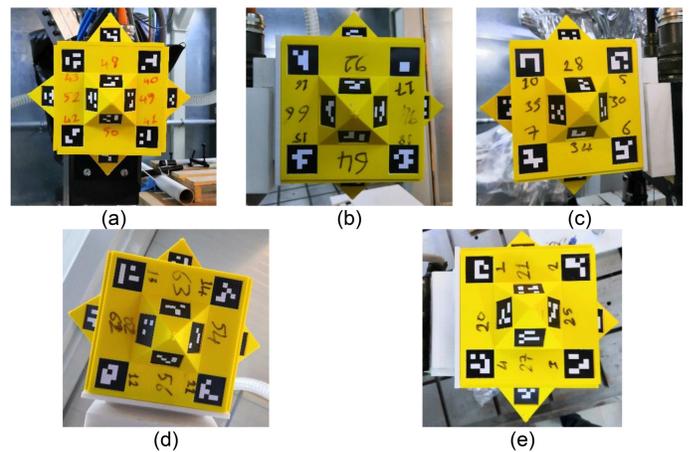


Fig. 2: (a) Front, (b) Left, (c) Right, (d) Bottom and (e) Top views of ArUco markers placed on the target to be tracked.

### B. Robot, Laser and Camera Frames Transformations

In order to evaluate the accuracy and precision of the vision based pose estimation, the coordinates of the laser tracker and the camera system were transformed into a fixed base ( $FB$ ) frame. The  $FB$  frame was defined by moving the robot's end effector to a point in world frame and then defining it as the origin. The overall transformation between the coordinates of the targets' and trackers' frames are given in Figure 3. Then a common target object's pose such as the laser target ( $LT$ ) was calculated in the  $FB$  frame from the Laser and Camera frames as given in (1)-(3). This way the pose of the common target in  $FB$  as obtained from the camera can be compared with the measurement's of the same target in  $FB$  as obtained from the laser tracker.

$${}^C T_{LT} = {}^C T_{CT} {}^{CT} T_{LT} \quad (1)$$

$$\{{}^{FB} T_{LT}\}^C = {}^{FB} T_C {}^C T_{LT} \quad (2)$$

$$\{{}^{FB} T_{LT}\}^L = {}^{FB} T_L {}^L T_{LT} \quad (3)$$

where  ${}^{CT} T_{LT}$  is the fixed rigid body transformation between the laser target and the camera target frames,  ${}^L T_{LT}$  and  ${}^C T_{CT}$  are provided by the measurements from the laser tracker and the camera based pose estimation, respectively.

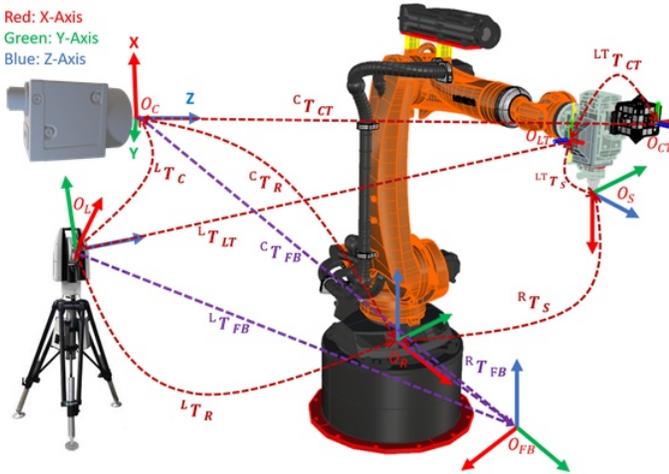


Fig. 3: Coordinate system transformations, where  ${}^A T_B$  is a transformation representing  $B$  in  $A$  and  $O_A$  denotes origin of frame  $A$ .  $C, L, R, CT, LT, S$  and  $FB$  denote Camera, Laser, Robot, Camera Target, Laser Target, Spindle and Fixed Base frames.

### III. IMPROVED POSE ESTIMATION USING LSTM

This work proposes to improve the pose estimation accuracy of vision based systems by utilizing supervised machine learning algorithms. This way existing camera based systems can be made to provide better accuracies when trained using the ground truth pose  $(X, Y, Z, \alpha, \beta, \gamma)$  such as the one provided by a laser tracker. In order to formulate this problem under a machine learning framework, the inputs and ground truth of the system needs to be determined properly. The ground truth in pose estimation problem can be obtained through the highly accurate laser tracker systems. As for inputs, the estimated pose  $(\hat{X}, \hat{Y}, \hat{Z}, \hat{\alpha}, \hat{\beta}, \hat{\gamma})$  provided by the vision system can be

obtained through standard pose estimation algorithms in literature such as the Levenberg Marquardt (LM) based algorithm [22]. As for the supervised learning algorithm, there exist several architectures through which dynamic systems can be modeled. One such algorithm is the LSTM which is proved to provide robust models for dynamic systems. LSTM networks were initially developed for sequence data such as word and sentence prediction, however in this work they will be modified to work with time varying data. Since the problem of pose estimation is a continuous and time dependent regression problem, this work proposes to use a LSTM to extract the dynamics from the data followed by a single feed forward neural network to provide the corrected pose. The proposed network's architecture is illustrated in Figure 4 where 6 inputs provided by the LM based algorithm are fed into the LSTM network. Therefore, the input contains 6 elements for each time frame and is denoted by  $X_t$ . They are then concatenated with the previous state vectors of a chosen size and passed through multiple gates. These gates are input ( $R_t$ ), output ( $O_t$ ), and forget ( $F_t$ ) gates each with individual activation functions of either sigmoid ( $\sigma$ ) or tanh, which are defined as:

$$R_t = \sigma(W_r[S_{t-1}, X_t]) \quad (4)$$

$$F_t = \sigma(W_f[S_{t-1}, X_t]) \quad (5)$$

$$O_t = \sigma(W_o[S_{t-1}, X_t]) \quad (6)$$

where  $W_r, W_f$  and  $W_o$  are the weight matrices associated with the input, forget, and output gates.

These gates are neural networks which can learn what information to keep or throw away during the training of the LSTM network. They determine the amount of information to add or throw away from the cell state, thus allowing only relevant data to pass through and reducing the effects of short-term memory problems seen in simple RNN networks. The cell/memory state ( $C_t$ ) acts as the memory of the network, thus information seen during the earlier time steps can make it through to the latest time steps. The cell/memory state is defined as:

$$C_t = F_t \odot C_{t-1} + R_t \odot \tilde{C}_t \quad (7)$$

where  $C_{t-1}$  is the previous cell/memory state,  $\odot$  denotes element wise product and  $\tilde{C}_t$  is another gate with the associated weight matrix  $W_c$  and is defined as:

$$\tilde{C}_t = \tanh(W_c[S_{t-1}, X_t]) \quad (8)$$

After data passes through these gates and the relevant information is extracted, the state vector ( $S_t$ ) can then be calculated from the output gate ( $O_t$ ) and the cell state ( $C_t$ ) as follows:

$$S_t = O_t \odot \tanh(C_t) \quad (9)$$

The number of states in the state vector is user defined and depends on the complexity of the problem at hand. After the states are obtained, they are passed through a single layer neural network with linear activation function so as to obtain the estimated pose of the target in current frame as follows:

$$\widehat{PS}_t = WS_t + B \quad (10)$$

where,  $W$  and  $B$  are the weight matrix and the bias vector associated with the fully connected layer.

Afterwards, the Euclidean norm of the error between the estimated pose  $\widehat{PS}_t^i$  and the ground truth pose provided by the laser tracker ( $PS_t^i$ ) is defined as the cost function as:

$$CF = \frac{1}{N} \sum_i^N \|PS_t^i - \widehat{PS}_t^i\|_2 \quad (11)$$

The number of states used for the LSTM as well as the number of units used in the fully connected layer are given in the experimental results section. The proposed method was coded in TensorFlow [24] software and optimized using Adam optimizer [25].

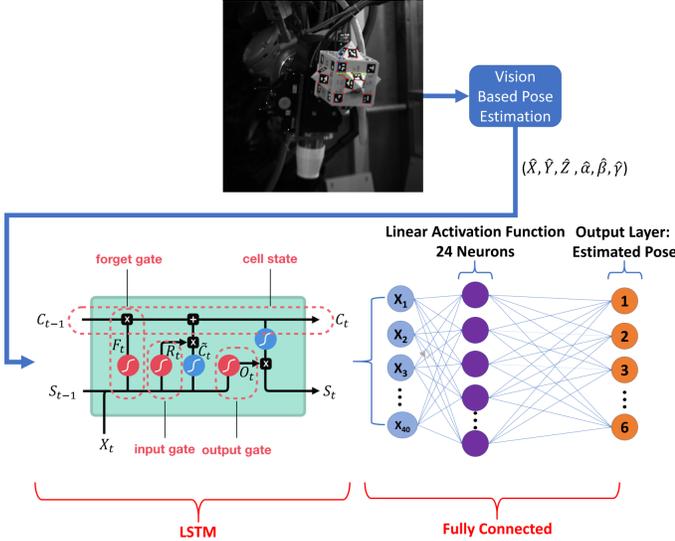


Fig. 4: The proposed LSTM neural network architecture for improving vision based pose estimation.

## IV. EXPERIMENTAL RESULTS

### A. Detection of the Camera Target

The realization of real time pose estimation is done in LabVIEW [26] software. The Basler ac2040-120um camera is connected to LabVIEW and the images can be acquired from 120 Hz with  $2048 \times 1536$  pixels upto 1220 Hz with  $160 \times 120$  pixels. The acquired images are then fed into the python node in which marker detection is performed using ArUco marker detection algorithm and the pose estimation is done through the Levenberg Marquardt optimization algorithm. In the experiments, the image size was set to  $640 \times 480$  pixels and were acquired at 375 Hz. The marker detection algorithm can work at 1000 Hz, the pose estimation at 1000 Hz and the proposed LSTM network operates at 1000 Hz for a single frame as well. Since all these algorithms are required to run in series, the total processing time is 0.003 seconds or about 333 Hz for a single frame<sup>1</sup>. The detected corners of the markers as well as the estimated pose of the object are shown in Figure 5. From these images it is clear that the designed

<sup>1</sup>Tested on a workstation with Intel Xeon E5-1650 CPU @ 3.5GHz and 16 GB RAM.

target object enables detection of multiple non-planar markers from any view, thus preventing rank deficiency in the pose estimation algorithm. Moreover, the camera is able to detect the markers with a viewing angle of  $\pm 90^\circ$  from all sides.

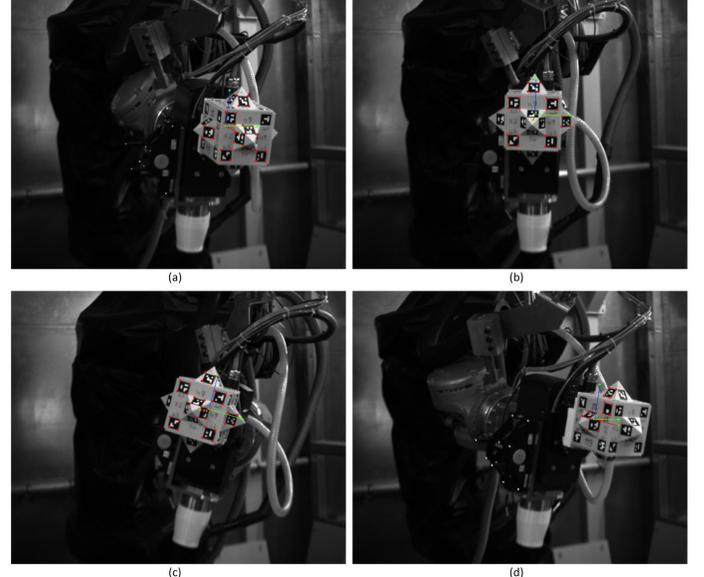


Fig. 5: (a) - (d) Samples showing marker detection (detected corners are in red) and estimated pose (red, green, blue coordinate axes) of the target object with respect to the camera frame.

### B. Pose Estimation Improved with LSTM

The accuracy and precision of the camera system was evaluated during trajectory tracking of a KUKA KR240 R2900 ultra's end effector based on ISO 9238. The ISO 9238 is typically used for evaluating the accuracy and repeatability of industrial robots through following a set trajectory for a number of times. Based on this standard, the robot's end effector is required to move to five specific points in its workspace and repeat it multiple times with or without performing any changes in the orientation of the robot's end effector. In order to evaluate the effectiveness of the constructed camera based pose estimation system and the proposed LSTM neural network in detail, in this work, 16 distinct trajectories were defined for the robot's end effector to follow while changing its orientation continuously around all of its three axes. Moreover, each trajectory contained five specific points at which the robot was stopped for 5 seconds as per the ISO 9238 guidelines. These 16 trajectories based on the ISO 9238 were relatively slow and took 105.9 minutes to complete.

Initially the LM based algorithm was implemented. Afterwards, the proposed machine learning architecture composed of a LSTM followed by a fully connected layer was used to improve the pose estimated by the LM based pose estimation algorithm as discussed in Section III. The proposed network was trained and validated three times in order to evaluate the robustness of the proposed method. For this purpose, initially the network was trained using 30% of the data and validated on the rest, then it was trained with 50% and validated on the rest and finally it was trained using 70% of the data and validated

on the remaining 30%. The training was performed using 2000 mini batches for 20000 iterations using 40 states for LSTM and 24 nodes at the fully connected layer. The absolute errors for position & orientation and their standard deviation for the ISO 9238 tracking are tabulated in Tables I to III. The errors given in these tables denoted as  $E_X$ ,  $E_Y$ ,  $E_Z$ ,  $E_{Roll}$ ,  $E_{Pitch}$ , and  $E_{Yaw}$  are the absolute errors between the estimated and ground truth pose provided by the laser tracker. We should note that ISO 9238 is a very challenging dataset for vision based pose estimation in that the distance between the camera and the target may increase a lot, in turn degrading the accuracy of the estimated pose. This is especially the case in this work due to the large working space covered ( $1140 \times 610 \times 945$  mm along the X, Y, and Z axes, respectively) by the robot in the conducted experiment, as seen from Figure 6. Moreover, due to the viewing angle restrictions the camera was placed 1 meter away from the closest point of the work space. Based on these, the distance between the camera and the target changed from 1 meters to 3 meters during the 16 trajectories followed by the robot, thus in turn made the position errors relatively high.

Figures 6 and 7 show the position and orientation trajectories of the laser target as tracked by the laser tracker in blue. The yellow trajectories from left to right are the ones estimated by the camera system using LM based pose estimation only and pose estimated by LM as input to the LSTM. These images are for training the proposed method with 70% of the data. Moreover, the results obtained for the validation set (30% of the data) for each individual axes are shown in Figure 8 for position tracking. As for orientation trajectories around X, Y and Z axes, hereby denoted as Roll, Pitch and Yaw, respectively, they are shown in Figure 9. As seen from the errors tabulated in Tables I to III, the proposed method is able to reduce the position tracking errors at least by 1.3, 1.2, and 2.23 times and upto 1.8, 1.6, and 2.94 times for X, Y, and Z axes, respectively when compared with the pure LM based algorithm using 30% and 70% of the data for training the models. This is in addition to reducing the standard deviation of the position errors by upto 1.41, 1.49, and 2.3 times for X, Y, and Z axes, respectively. Furthermore, the orientation tracking errors were reduced by at least 2.45, 2.04, and 2.67 times and upto 2.93, 2.42, and 3.14 times for Roll, Pitch and Yaw axes, respectively. Moreover, the standard deviation of orientation errors were reduced by upto 1.4 and 1.72 times for the Pitch and Yaw axes while providing similar results for the Roll axis.

From these results, it is seen that the proposed method is able to improve the position and orientation tracking accuracies even when 30% of the data is used for training the proposed network, thus proving its robustness.

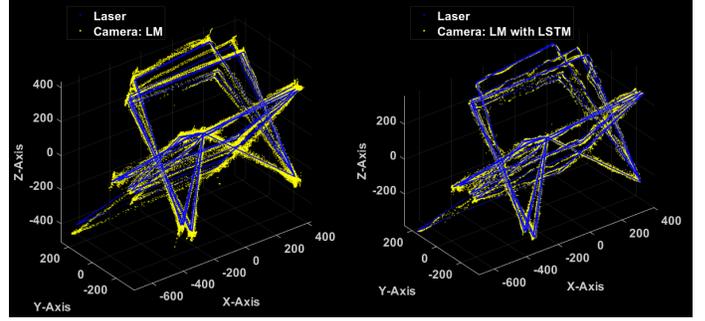


Fig. 6: Position tracking results based on ISO 9238.

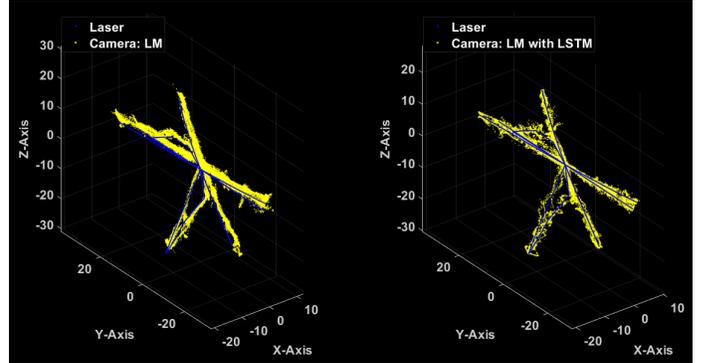


Fig. 7: Orientation tracking results based on ISO 9238.

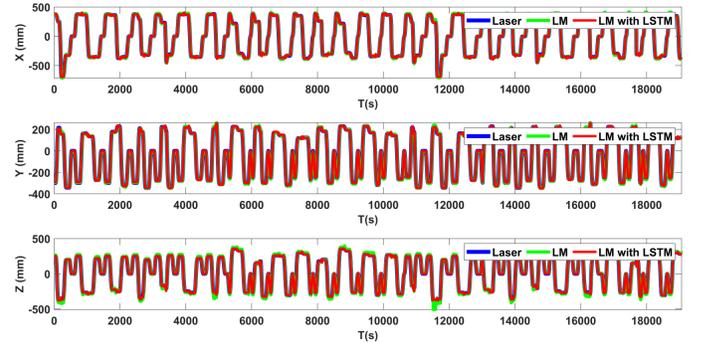


Fig. 8: Laser vs LM vs LM with LSTM for position tracking, validated on the 30% of the dataset.

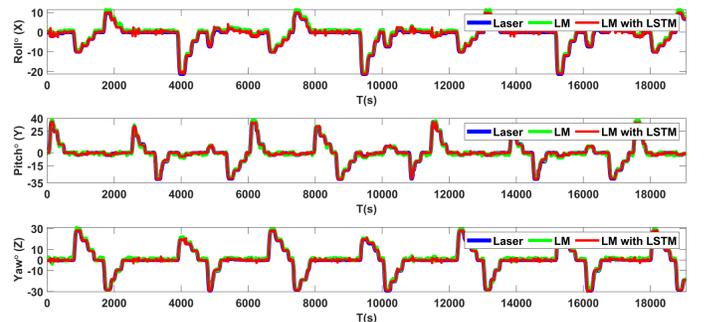


Fig. 9: Laser vs LM vs LM with LSTM for orientation tracking, validated on the 30% of the dataset.

TABLE I: Pose tracking errors during trajectory tracking based on ISO 9238, trained with 30% of the dataset and validated on the rest.

| Training Size                                     | 30% of the Dataset |             |               |                |                 |               |
|---|--------------------|-------------|---------------|----------------|-----------------|---------------|
| Error for the Validation Set (70% of the Dataset) | $E_X$ (mm)         | $E_Y$ (mm)  | $E_Z$ (mm)    | $E_{Roll}$ (°) | $E_{Pitch}$ (°) | $E_{Yaw}$ (°) |
| LM  | 9.84 (9.86)        | 7.30 (6.61) | 16.44 (14.07) | 0.93 (0.33)    | 1.02 (0.89)     | 1.15 (0.72)   |
| LM with LSTM                                      | 7.52 (8.83)        | 6.10 (5.77) | 7.34 (6.97)   | 0.38 (0.40)    | 0.50 (0.62)     | 0.43 (0.47)   |

The ( ) beside the errors contain their standard deviation.

TABLE II: Pose tracking errors during trajectory tracking based on ISO 9238, trained with 50% of the dataset and validated on the rest.

| Training Size                                     | 50% of the Dataset |             |               |                |                 |               |
|---|--------------------|-------------|---------------|----------------|-----------------|---------------|
| Error for the Validation Set (70% of the Dataset) | $E_X$ (mm)         | $E_Y$ (mm)  | $E_Z$ (mm)    | $E_{Roll}$ (°) | $E_{Pitch}$ (°) | $E_{Yaw}$ (°) |
| LM  | 9.85 (9.87)        | 7.35 (6.62) | 16.23 (13.60) | 0.92 (0.32)    | 1.01 (0.88)     | 1.14 (0.71)   |
| LM with LSTM                                      | 6.30 (7.36)        | 4.76 (4.97) | 6.10 (6.43)   | 0.36 (0.40)    | 0.40 (0.43)     | 0.39 (0.38)   |

The ( ) beside the errors contain their standard deviation.

TABLE III: Pose tracking errors during trajectory tracking based on ISO 9238, trained with 70% of the dataset and validated on the rest.

| Training Size                                     | 70% of the Dataset |             |               |                |                 |               |
|---|--------------------|-------------|---------------|----------------|-----------------|---------------|
| Error for the Validation Set (70% of the Dataset) | $E_X$ (mm)         | $E_Y$ (mm)  | $E_Z$ (mm)    | $E_{Roll}$ (°) | $E_{Pitch}$ (°) | $E_{Yaw}$ (°) |
| LM  | 10.11 (10.20)      | 7.39 (6.78) | 15.79 (13.69) | 0.91 (0.33)    | 1.04 (0.87)     | 1.10 (0.67)   |
| LM with LSTM                                      | 5.56 (7.21)        | 4.60 (4.54) | 5.37 (5.94)   | 0.31 (0.35)    | 0.43 (0.62)     | 0.35 (0.39)   |

The ( ) beside the errors contain their standard deviation.

## V. CONCLUSION

In this work a machine vision based pose estimation system was developed for industrial robots. Initially a camera target was designed and fitted with markers so as to guarantee the visibility of at least two non planar markers from any viewing angle within  $\pm 90^\circ$  in all directions of the camera's view. Moreover, a Long Short Term Memory (LSTM) network followed by a fully connected layer was proposed to increase the accuracy and precision of the vision based pose estimation using a laser tracker's data as the target values where the pose estimated by Levenberg-Marquardt (LM) was utilized at the input of the LSTM network.

The proposed method was validated by tracking an industrial robot's end effector for 16 distinct trajectories based on ISO 9238. The trajectories were followed by a KUKA KR240 R2900 ultra robot and the ground truth data was provided by the Leica AT960 laser tracker. As shown from the experimental results, the proposed method was able to reduce the position tracking errors by upto 1.8, 1.6, and 2.94 times for X, Y, and Z axes, respectively when compared with the pure LM based algorithm. This is in addition to reducing the standard deviation of the position errors by upto 1.41, 1.49, and 2.3 times for X, Y, and Z axes, respectively. As for the orientation tracking errors, the proposed method was able to reduce these errors by upto 2.93, 2.42, and 3.14 times for Roll, Pitch and Yaw axes, respectively. All the while reducing the standard deviation of these errors by upto 1.4 and 1.72 times for the Pitch and Yaw axes while providing similar results for the Roll axis. Therefore, the proposed method is able to significantly increase the accuracy and precision of the standard LM based pose estimation algorithm during trajectory tracking of industrial robots' end effector.

As a future work, the improved pose estimation results can be utilized in a position based visual servoing (PBVS) scheme to increase task accuracies of various robotic manufacturing processes such as machining.

## REFERENCES

- [1] Klimchik, A., Ambiehl, A., Garnier, S., Furet, B., and Pashkevich, A. Efficiency evaluation of robots in machining applications using industrial performance measure. *Robotics and Computer-Integrated Manufacturing*, 48, 12-29, 2017.
- [2] Wilson, W. J., Hulls, C. W., and Bell, G. S. Relative end-effector control using cartesian position based visual servoing. *IEEE Transactions on Robotics and Automation*, 12(5), 684-696, 1996.
- [3] Keshmiri, M., and Xie, W. F. Image-based visual servoing using an optimized trajectory planning technique. *IEEE/ASME Transactions on Mechatronics*, 22(1), 359-370, 2016.
- [4] Hashimoto, K. A review on vision-based control of robot manipulators. *Advanced robotics: the international journal of the Robotics Society of Japan*, 17(10), 969-991, 2003.
- [5] Chaumette, F. Potential problems of stability and convergence in image-based and position-based visual servoing. In *The confluence of vision and control* (pp. 66-78). Springer, London, 1998.
- [6] Shi, X., Zhang, F., Qu, X., and Liu, B. An online real-time path compensation system for industrial robots based on laser tracker. *International journal of advanced robotic systems*, 13(5), 2016.
- [7] Qu, W. W., Dong, H. Y., and Ke, Y. L. Pose accuracy compensation technology in robot-aided aircraft assembly drilling process. *Acta Aeronautica et Astronautica Sinica*, 32(10), 2011.
- [8] Shu, T., Gharaaty, S., Xie, W., Joubair, A., and Bonev, I. A. Dynamic path tracking of industrial robots with high accuracy using photogrammetry sensor. *IEEE/ASME Transactions on Mechatronics*, 23(3), 2018.
- [9] <https://comet-project.eu/results.asp>
- [10] Nissler, C., Stefan, B., Marton, Z. C., Beckmann, L., and Thomasy, U. Evaluation and improvement of global pose estimation with multiple apriltags for industrial manipulators. In *2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation*, 2016.
- [11] Claes, K., and Bruyninckx, H. Robot positioning using structured light patterns suitable for self calibration and 3D tracking. In *Proceedings of the 2007 International Conference on Advanced Robotics, Korea, 2007*.
- [12] Liu, B., Zhang, F., and Qu, X. A method for improving the pose accuracy of a robot manipulator based on multi-sensor combined measurement and data fusion. *Sensors*, 15(4), 7933-7952, 2015.
- [13] Wilson, W. J., Hulls, C. W., and Bell, G. S. Relative end-effector control using cartesian position based visual servoing. *IEEE Transactions on Robotics and Automation*, 12(5), 684-696, 1996.
- [14] Janabi-Sharifi, F., and Marey, M. A kalman-filter-based method for pose estimation in visual servoing. *IEEE transactions on Robotics*, 2010.
- [15] Ficocelli, M., and Janabi-Sharifi, F. Adaptive filtering for pose estimation in visual servoing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems Proceedings*, 2001.
- [16] D'Errico, G. E. A la Kalman filtering for metrology tool with application to coordinate measuring machines. *IEEE Transactions on Industrial Electronics*, 59(11), 4377-4382, 2011.
- [17] Alcan, G. Data driven nonlinear dynamic models for predicting heavy-duty diesel engine torque and combustion emissions, Ph.D. Thesis, 2019.
- [18] Mumcuoglu, M. E., Alcan, G., Unel, M., Cicek, O., Mutluergil, M., Yilmaz, M., and Koprubasi, K. Driving Behavior Classification Using Long Short Term Memory Networks. In *AEIT International Conference of Electrical and Electronic Technologies for Automotive*, 2019.
- [19] Alcan, G., Yilmaz, E., Unel, M., Aran, V., Yilmaz, M., Gurel, C., and Koprubasi, K. Estimating soot emission in diesel engines using gated recurrent unit networks. *IFAC-PapersOnLine*, 52(5), 544-549, 2019.
- [20] Aran, V., and Unel, M. Gaussian process regression feedforward controller for diesel engine airpath. *International Journal of Automotive Technology*, 19(4), 635-642, 2018.
- [21] Hochreiter, S., and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8), 1735-1780, 1997.
- [22] Darcis, M., Swinkels, W., Güzel, A. E., and Claesen, L. PoseLab: A Levenberg-Marquardt Based Prototyping Environment for Camera Pose Estimation. *11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, 2018.
- [23] Romero-Ramirez, F. J., Muñoz-Salinas, R., and Medina-Carnicer, R. Speeded up detection of squared fiducial markers. *Image and vision Computing*, 76, 38-47, 2018.
- [24] Kingma, D. P., and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [25] <https://www.tensorflow.org/>
- [26] <https://www.ni.com/en-tr/shop/labview.html>