A critical evaluation of various methods of using a laser scanner with an industrial robotic arm.

D De Becker, J Dobrzanski, YM Goh, L Justham
Wolfson School of Mechanical, Electrical and Manufacturing Engineering, Loughborough University, United Kingdom

Abstract

A major difficulty with integrating a laser line scanner and an industrial robotic arm is the closed nature of the robot controllers. Due to this closed nature, the usual way of integrating a laser line scanner and an industrial robotic arm is to use an external PC or PLC which receives the robot position and the laser line scan data over a communication protocol such as TCP or UDP. This external PC can then be used to reconstruct the geometry in the robot's co-ordinate system.
This paper uses this external PC set up to explore three implementations to match the robot position and scan data. These methods are point to point, interpolation and timestamp. All three methods can be implemented with various degrees of difficulty and with differing benefits in accuracy, repeatability and speed.
Each of these three methods requires various levels of effort, this paper critiques these three methods therefore making it easier for other users of laser line scanners to decide which method would best suit their application.

1 Introduction

Laser line scanners are widely used within robotic inspection applications varying from abnormality or defect detection [1], weld seam tracking [2], to reverse engineering applications [3]. When implementing a laser line scanning inspection method, the key is transforming the data from the laser scanner's coordinate system to the robot's co-ordinate system or a similar global co-ordinate system. To complete this transformation a laser line scan must be carefully matched with an accurate robot position. To decrease the latency between these two systems, the laser scanner should be directly controlled, and data processed on the robot controller. However, due to the closed nature of robot controllers, generally requiring expensive optional extras to get robot I/O, real time positioning or simply not allowing access to the robots' root systems, this is difficult to achieve. When this is coupled with the processing power required for point cloud manipulations or co-ordinate system transformation, this is usually not possible on the robot controller. This has meant that most implementations have used an external PC or PLC which communicates over a TCP or UDP protocol. This communication and the inherent latency cause difficulties in assigning a robot position to each laser line scan, which this paper is going to address.

Although a large amount of work has already been completed on integrating laser line scanners and industrial robotic arms, to the authors' knowledge, there is no easily accessible critique of the various implementations or techniques used. This paper discusses three ways in which a laser line scanner can be implemented with the use of a robotic arm. These methods are the point the point method, the interpolation method and finally the timestamp method. The point to point method is usually used for initial testing or calibrating due to its simple implementation and accurate robot positioning. The timestamp method is used in more industrial ready products due to its significantly faster speed over the point to point method. This leaves the interpolation method as a good middle ground for many applications due to its easier implementation over the timestamp method and quicker scanning speed to the point to point method. A drawback of this method is that the range of robot movements which can be completed, is limited.

2 Method description

In this section each method is described with its corresponding advantages and drawbacks as well a description of the hurdles which must be overcome to implement the method. The three methods use the robot pose relative to a base as given by the robot and then uses a hand eye homogeneous transformation matrix to go from the laser scanner's co-ordinate system to the robot's base co-ordinate system; with the initial *guess* created by the robot's 4-point tool calibration. The method of gaining this hand eye transformation matrix can be done multiple ways, with this paper using a fitted plane approach [1]. This is where a plane is scanned at various joint angles covering as much of the working volume as possible. These scans are then handed to a non-linear solver which attempts to reduce the error of a fitted plane to the scans by adjusting the 6 hand eye calibration values. These include three rotations (α, β, γ) and three translations (z, y, x). The rotation and translations are then put back into the robot controller or can be used offline as a further transformation after the scans have been completed.

2.1 Point to point method

The point to point method is the easiest to implement as it only requires the user to drive to a position in 3D space and pulse the laser scanner to take a single scan. When scanning large volumes this can take a large amount of time, for example with a robot moving at 50mm/s, accelerating at 1m/s^2, scanning at 1 mm intervals would result in a scanning speed of 15.81mm/s. This speed does not allow for a settling time which would have to be introduced to gain a better accuracy. This method is the simplest as it only requires simple robot control, with no real-time robot positioning and laser scanner control. The robot simply reports its position once it has reached the desired location in 3D space and the external PC can trigger the laser scanner to take a single scan. This scan data and the robot position are then fed into the transformation matrix and the point cloud is created. Due to communication latency not being a problem in the quality of the point clouds, these scans can be used to determine the hand eye calibration transformation matrix with a high level of certainty.

2.2 Interpolation method

The second method is the interpolation method, this is where the start and end position of the robot during the scan are recorded. Using the laser scanner's capture frequency, the scans can be aligned to a corresponding estimated robot position. The start and end position can either; 1) be saved on the robot directly and then transferred after the scanning is complete or 2) using timestamps and the estimated communication delay between the robot and PC; the corresponding position can be recorded in 'real-time'. Estimating this delay has been discussed further in section 2.3. Depending on the robot path, various interpolation methods may be used; if a linear move is used then a simple linear interpolation on each axis can be implemented. This is due to the nature of a linear move; the XYZ and corresponding rotations are moved in a linear motion independently. If this is not the case for the robot being used, or if the motion is a point to point move, then, assuming that the move only has a single axis of rotation, the Rodrigues method [4] can be used to find the axis, or vector, and the angle of rotation. The angle of rotation can then be divided by the number of scans allowing the estimated robot position to be calculated at the specific rotation angle around the vector. Both the Rodrigues and linear interpolation methods require the scanner frequency and the robot speed to be constant and any deviation in these constants will cause an error in the estimated position. Although either interpolation method can be used, the linear interpolation method has been tested in section 3 and 4. Due to the constant robot velocity requirement, this method disregards the acceleration and deceleration portion of the robot's movement therefore no scans were taken at the start or end of the robot's path. The final nuance of this method is the problem of Euler equivalency or symmetry [5]. This was found on both the Kuka KRC-2 and KRC-4 controllers that were used within these trials. This equivalency becomes particularly evident when the β (rotation around y) value is equal to 0 and thus α (rotation around x) and γ (rotation around z) can be any value if the sum of α and γ does not change. A further equivalency can also be seen for multiples of 2π for any of the angles or for the relationship ($\alpha+\pi$, $-\beta$, $\gamma+\pi$) [5], although, these both appear less commonly in robotic applications. To overcome the equivalency effect, the user must calculate the rotation's principal angles, this can be done by converting the Euler angles to a transformation matrix and then back to the Euler angles or by setting limits for α and γ to mod 2π and β to mod π.

Overall this method allows for a greater capture rate than the point to point method as well as being less susceptible to communication latency compared to the timestamp method. However, this method does introduce a significant reduction in robot path types as well as a reduction in the usable regions of the robot path.

2.3 Timestamp method

The final method considered in this work uses timestamps to match the corresponding robot position and the time which the scan was taken. Although this method is probably the simplest to understand, it is the most difficult to implement and the most prone to noise due to its reliance on real-time robot positioning. For this method each laser scan is matched to its two nearest robot positions, then a linear interpolation is completed between these two positions to approximate the position of the robot at the time of the laser scan.
The key in implementing this method is measuring the time delay between the external PC or PLC and the robot controller. The time delay must be categorized to allow for the user to implement this method, if there is a constant delay then this can be put into the reconstruction as an offset. However, if it is variable or inconsistent then this method becomes difficult to implement. To measure the delay, multiple different methods and approaches can be used. The solution used in this paper utilised an artefact with a sine wave embedded into it, as shown in Figure 1.

Figure 1: Artefact to measure the time delay between the robot and the external PC.

Using a laser line scanner, a point cloud can be generated of the artefact shown in Figure 1. Extracting the centre of the sine wave embedded into the artefact allows for the phase of the sine wave to be measured. If there is a communication delay, the user should observe an apparent increase in the phase of the sine wave with respect to robot speed, as shown in Figure 2(A). Plotting this phase against various robot speeds allows the user to observe if the time delay is constant. If it is constant, this would be shown as a straight line where the gradient is equal to the time delay of the robot. The test results of this method can be seen in Figure 2(B).
The strengths of this method are the versatility in allowing the robot to move in various paths and allowing the laser scanner to capture even when the robot is accelerating and decelerating. This is a significant advantage over the interpolation method, however, it adds the difficulty of having to both measure and quantify the communication as well as adding the potential error due to communication delays.

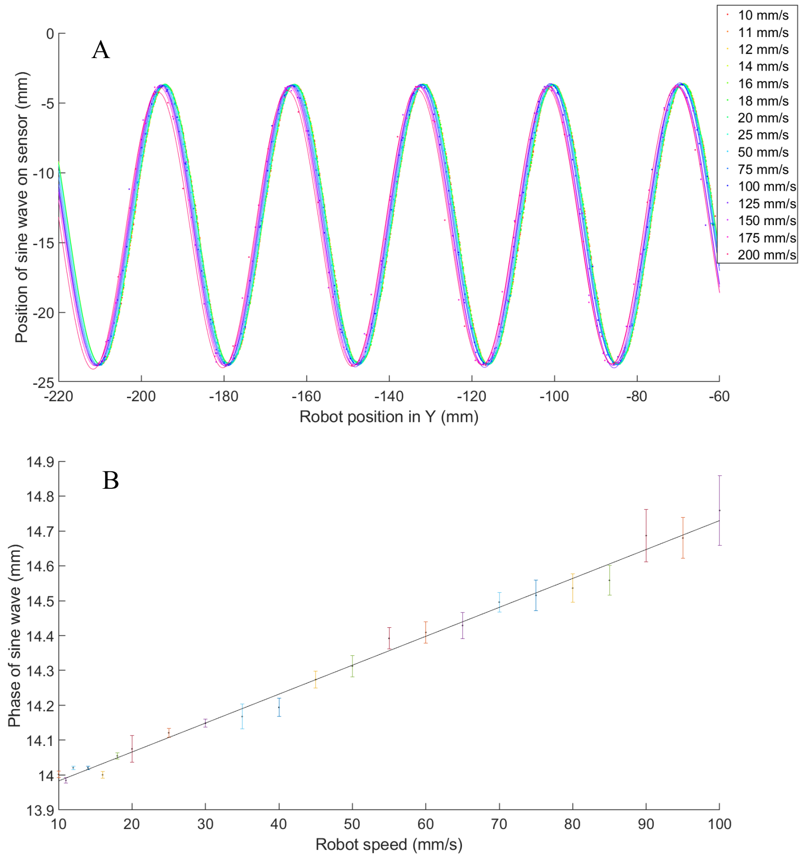

Figure 2: Graphs showing the extracted centre of the sine wave artefact (A) and experimental results to calculate the communication delay between the robot and the external PC (B).

3 Equipment and testing methodology

The equipment used in testing all three methods included a Kuka KR16 with a KRC2 controller as well as a Micro Epsilon 2900-100/BL laser line scanner. The control for both the robot and the laser line scanner was done on an external PC using LabVIEW. Finally, to compare meshes and point clouds, Cloud Compare [6] was used. The test rig can be seen in Figure 3.

The testing involved scanning a 3D printed artefact using multiple passes to scan all sides of the artefact. After printing the artefact, it was initially scanned with a David scanner [7] the results of which can be seen in Figure 4. The data presented in Figure 4 show a mean error of 0.0138mm and a standard deviation of 0.111mm. These errors are due to the shrinkage occurring as part of the 3D printing manufacturing process. Due to the geometry and orientation of the David scanner, the internal geometry could not be accurately captured which is represented by the large range of ±0.4mm as seen in Figure 4.

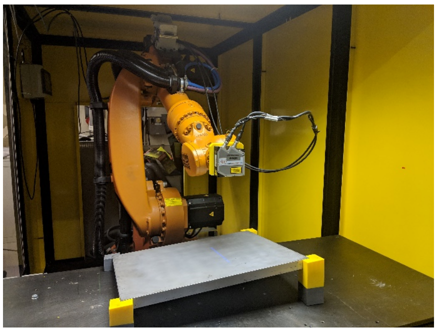

Figure 3: Image showing the test rig setup.

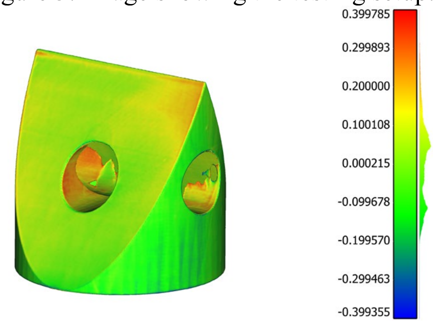

Figure 4: Comparison between CAD geometry and output from David scanner.

After the artefact was characterized it allowed each laser line scan and thus each scan methodology to be compared against the David scanner geometry. This meant aligning the scanned point clouds to the mesh generated by the David scanner which was completed using the ICP (Iterative closest point) method. Finally, the normal distance between the point cloud and the mesh generated by the David scanner were calculated. This was repeated 10 times and created the basis for the accuracy comparison between the various methods. To calculate the repeatability of each scanning method, each of the 9 subsequent scans were compared to the initial scan where the distance between each point and its closest neighbour was calculated. Throughout testing the key capture parameters remained constant, the values of which can be seen in Table 1. For all methods the hand eye transformation and robot path where also kept constant.

Table 1: Constant testing parameters

Parameter	**Value**
Scanner frequency	50Hz
Scanner wavelength	405nm
Robot Velocity	0.05m/s
Robot acceleration	1m/s^2
Communciation delay	8.297ms

4 Results and comparison of each method

This section has been split into the three integration methods with a heatmap shown for the first run of each test. The results from the subsequent runs have been shown in the table following the heatmaps. The heatmaps show a large range of values, however, this is mostly due to the surface which the artefact is sitting on being scanned, in addition to the artefact itself. Therefore, the results shown include the mean error and standard deviation only, as the range values were greatly affected by outliers generated by surfaces surrounding the artefact.

4.1 Point to point method

To test the point to point method each pass was divided into 25 intermediary positions equalling a total of 178 positions or scans. This led to an absolute average mean distance error of 0.154mm and a standard deviation of 0.800mm. These results are in the same order of magnitude of the mean error and standard deviation of the hand eye measurement which was equal to 0.1426mm and 0.2023mm respectively, as well as being within the expected limitations of an industrial robotic arm. Figure 5(A) shows a colourmap of the first run with Table 2 showing the results of the subsequent runs.

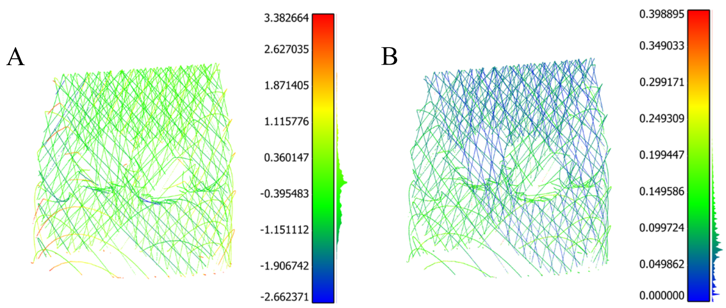

Figure 5: Accuracy (A) and repeatability (B) result for the first run of the point to point method.

Table 2: Accuracy and repeatability results of the point to point method.

Run Number	Accuracy		Repeatability	
	Mean error (mm)	**STD (mm)**	**Mean (mm)**	**STD (mm)**
1	-0.200	0.805	N/A	N/A
2	-0.131	0.794	0.094	0.068
3	-0.147	0.796	0.098	0.052
4	-0.156	0.802	0.104	0.052
5	-0.167	0.809	0.104	0.057
6	-0.123	0.793	0.114	0.062
7	-0.132	0.798	0.117	0.064
8	-0.177	0.812	0.121	0.070
9	-0.137	0.792	0.120	0.065
10	-0.164	0.803	0.123	0.070

The repeatability testing showed an average standard deviation of the distance between neighbouring points to be 0.0622mm. The results for the first run can be seen Figure 5(B). This shows a high level of repeatability when compared to the standard deviation of the accuracy testing. The system has a greater repeatability than accuracy which is consistent with other robotic systems [8].

4.2 Interpolation method

The results for the accuracy testing of the interpolation method can be seen in Figure 6 (A) and Table 3. The results show an average mean error distance of 0.0462mm with a standard deviation of 0.795mm. As with the point to point method, this is comparable to the mean and standard deviation of the hand eye which was equal to 0.1426mm and 0.2023mm respectively.

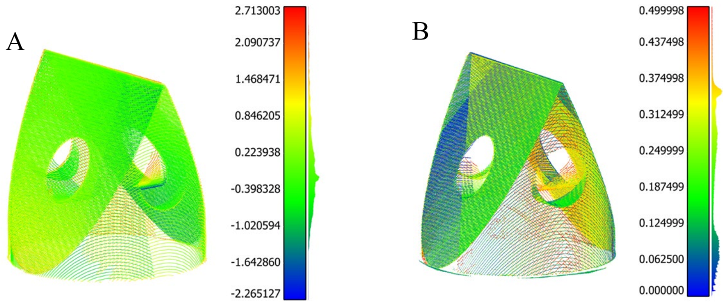

Figure 6: Accuracy (A) and repeatability (B) result for the first run of the interpolation method.

The repeatability testing showed a standard deviation of the distance between neighbouring points of 0.127mm. When this standard deviation is compared to the 0.795mm of the accuracy testing, it shows that the repeatability of the laser scanning system is much better than the overall accuracy of the system as shown in Figure 6 (B) with further details within Table 3.

Table 3: Accuracy and repeatability results the interpolation method.

Run Number	Accuracy		Repeatability	
	Mean error (mm)	STD (mm)	Mean (mm)	STD (mm)
1	0.007	0.748	N/A	N/A
2	0.047	0.825	0.212	0.122
3	-0.036	0.715	0.260	0.169
4	0.063	0.799	0.249	0.125
5	0.009	0.786	0.231	0.134
6	0.094	0.839	0.255	0.144
7	0.034	0.840	0.210	0.120
8	0.059	0.793	0.196	0.093
9	0.086	0.806	0.241	0.108
10	0.099	0.796	0.234	0.135

4.3 Timestamp method

The results for the accuracy testing of the timestamp method can be seen in Figure 7 (A) and Table 4. The results show an average mean distance of 0.067mm with a standard deviation of 0.798mm. As with the interpolation method, this is comparable to the mean and standard deviation of the hand eye approach.

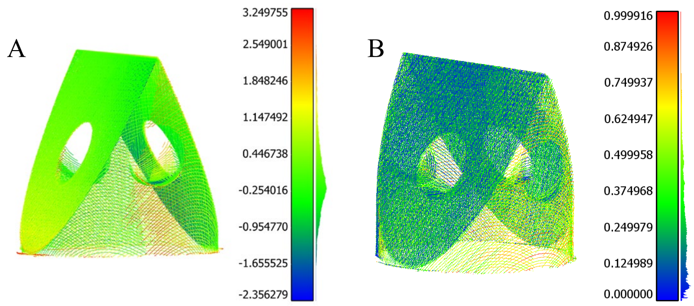

Figure 7: Accuracy (A) and repeatability (B) result for the first run of the timestamp method.

In a similar manner to the interpolation and the point to point methods, the repeatability testing showed a lower standard deviation of 0.1615mm compared to the accuracy testing, once again showing that the system has a greater repeatability than accuracy. This data is shown in Figure 7 (B) for the first run of the timestamp method and a full table of results can be seen in Table 4.

Table 4: Accuracy results for the timestamp method.

Run Number	**Accuracy**		**Repeatability**	
	Mean error (mm)	**STD (mm)**	**Mean (mm)**	**STD (mm)**
1	0.075	0.802	N/A	N/A
2	0.043	0.795	0.293	0.205
3	0.010	0.802	0.216	0.177
4	0.067	0.798	0.253	0.160
5	0.061	0.789	0.227	0.142
6	0.083	0.795	0.280	0.166
7	0.068	0.800	0.144	0.141
8	0.061	0.794	0.196	0.138
9	0.046	0.792	0.255	0.167
10	0.063	0.809	0.224	0.158

5 Conclusion

To conclude, all three methods show a very similar standard deviation when compared to the original geometry - with the accuracy of the system being comparable to the overall hand eye calibration accuracy. Therefore, the only way of increasing this accuracy further would be to generate a more accurate hand eye calibration transformation. It should be noted, however, that these measurements and findings are most likely to be limited by the overall accuracy of the robot. The testing also showed that the point to point method showed the greatest repeatability among all three methods, this is mainly due to the greater confidence in knowing the robot position at the time of taking the laser line scan over the other two methods. Overall all three methods showed a better repeatability than overall accuracy, which is consistent with almost all industrial robotic arms systems. Finally, the testing also showed a reduction in error of the interpolation method over the timestamp method which is most likely due to the timestamp method being highly susceptible to the calculation of the communication delay. Therefore, further work should be done on using different techniques to quantify this communication delay more accurately. Further work could also be completed on designing an artefact that isn't symmetrical thus making the aligning stage of the testing less susceptible to error. Furthermore, designing a scan path that reduces the scan area around the artefact would also be beneficial in reducing the number of erroneous points. The results and the work undertaken to complete these tests have shown that the interpolation method tends to be the best middle ground for accuracy and simplicity. This method is less susceptible to communication delays between the robot and the external PC compared to the timestamp method whilst allowing for a large amount of data to be captured. The main drawbacks of this method are that it does restrict the available moves that can be completed and is dependent on either capturing the varying laser scanner frequency or making sure that there is constant capture rate.

References

[1] S. Sharifzadeh, I. Biro, N. Lohse, and P. Kinnell, "Abnormality detection strategies for surface inspection using robot mounted laser scanners," *Mechatronics*, vol. 51, no. November 2016, pp. 59–74, 2018.

[2] F. B. Carlson, R. Johansson, and A. Robertsson, "Six DOF eye-to-hand calibration from 2D measurements using planar constraints," *IEEE Int. Conf. Intell. Robot. Syst.*, vol. 2015–Decem, pp. 3628–3632, 2015.

[3] P. Laube, M. O. Franz, and G. Umlauf, "Evaluation of features for SVM-based classification of geometric primitives in point clouds," *Proc. 15th IAPR Int. Conf. Mach. Vis. Appl. MVA 2017*, pp. 59–62, 2017.

[4] C. Tomasi, "Vector Representation of Rotations," Durham, North Carolina, 2013.

[5] Felix Pahl, "How can I find equivalent Euler angles?," *Mathematics Stack Exchange*, 2011. [Online]. Available: https://math.stackexchange.com/q/55552 (version: 2011-08-05). [Accessed: 04-Dec-2018].

[6] GPL software, "CloudCompare." 2018.

[7] Hewlett Packard, "David Scanner (SLS 2)." 2019.

[8] J. Chen and L. M. Chao, "Positioning Error Analysis for Robot Manipulators with All Rotary Joints," *IEEE J. Robot. Autom.*, vol. 3, no. 6, pp. 539–545, 1987.