



HAL
open science

Improving Automated Insertion Task in Robotics by Reducing Registration Error

Geraldine S. Cheok, Marek Franaszek, Karl Van Wyk, Jeremy A. Marvel

► **To cite this version:**

Geraldine S. Cheok, Marek Franaszek, Karl Van Wyk, Jeremy A. Marvel. Improving Automated Insertion Task in Robotics by Reducing Registration Error. 9th International Precision Assembly Seminar (IPAS), Dec 2020, Held virtually, Unknown Region. pp.108-121, 10.1007/978-3-030-72632-4_8. hal-03520400

HAL Id: hal-03520400

<https://inria.hal.science/hal-03520400>

Submitted on 11 Jan 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Improving Automated Insertion Task in Robotics by Reducing Registration Error

Geraldine S. Cheok¹, Marek Franaszek¹, Karl Van Wyk^{1,2}, and Jeremy A. Marvel¹

¹ National Institute of Standards and Technology, Gaithersburg, MD 20899, USA
{cheok, marek, jeremy.marvel}@nist.gov

² NVIDIA, 4545 Roosevelt Way, Seattle, WA 98105, USA
kvanwyk@nvidia.com

Abstract. A peg-in-hole operation is representative of common tasks performed in assembly lines by robots. It requires registration of the coordinate frame where a part resides to the robot frame where it is acted upon. Poor registration causes misalignment of the peg and the hole which will result in a failed insertion of the peg. In this study, the dependence of the outcome of the insertion task on the quality of registration is investigated. It was shown in earlier studies that Restoring Rigid-Body Condition (RRBC) leads to decreased target registration error. This study quantifies the improvement in the Pass/Fail rate when the target registration error is reduced for a given peg-hole tolerance. A robot arm operated in position control mode was used to perform the insertion without performing any search algorithms or force/torque sensing. Results show that use of the RRBC method yielded substantial reduction in failed insertions, especially for tighter tolerances.

Keywords: peg-in-hole, rigid-body registration, target registration error.

1 Introduction

Accurate knowledge of an object's location is essential in manufacturing-based robotic applications. This is particularly important for the peg-in-hole task, which is an extensively studied robotic problem and the basis for many common tasks in assembly lines [1]. Research to improve this task has been conducted, and most of the proposed methods assume a peg can be initially tilted relative to the hole centerline and a search algorithm is used to guide the peg into the hole [2]. Many of the algorithms rely on feedback from force/torque (F/T) sensors [3, 4]. The premise of these methods is to improve performance given the errors (systematic and random) of the hole and peg locations [5]. The method described in this paper improves task performance by reducing the error in the registered hole location by applying corrections to the points transformed from world to robot frame. The method reduces registration error by restoring the rigid-body condition (RRBC) for point-based rigid-body registration [6]. This approach is similar to the Volumetric Error Compensation (VEC) technique, but it is easier to use as no advanced, analytical modelling of the error function is required [7-9]. It is reasonable to expect that improvement in the world-to-robot registration will have a positive

impact on the peg-in-hole task; thus, increasing the likelihood for a successful insertion. However, the level of expected improvement has not been quantified. This paper quantifies the improvement in the peg-in-hole task that can be achieved by reducing the registration error. Recalibration of the perception system and/or remastering the robot arm (tasks which may require skills not readily available in small manufacturing companies) would reduce the registration error, but the method described in this paper is an easier task. Unlike some other approaches [10], the RRBC method does not require installation of additional hardware such as F/T sensors as it merely corrects the endpoints of the robot's path and no sensing is involved.

It was shown in [6] that RRBC method can substantially improve the quality of registration as gauged by the reduction of the Target Registration Error (TRE) – targets are points of interest to which registration is applied. To assess how the reduction of TRE improves the performance of a peg-in-hole task in a robotic application, three experiments were conducted. In these experiments, a robot arm, operated in position control mode, was commanded to move to a specified location and to insert a peg into a hole; that is, no modification of the preplanned path from a search algorithm or active force/torque feedback was used. This constitutes a very challenging test as the success of such an insertion depends almost entirely on accurate initial positioning. The coordinates of the hole were obtained by registering a world coordinate frame to the robot coordinate frame using a regular rigid-body registration. Then, automated insertions were performed using uncorrected hole locations and locations corrected by the RRBC method. A Pass/Fail metric was used to determine if the implementation of the RRBC method would increase the number of successful insertions. The RRBC method may be beneficial for insertion tasks when feedback from force/torque sensors is not available or sufficiently reliable or for other types of tasks such as automated drilling [11] or welding [12] where accurate positioning is crucial.

This paper summarizes the experiments, results, and conclusions from the three peg-in-hole experiments described in [13].

2 Related Work

Strategies used for peg insertion generally fall in two main categories: those which use active feedback from force/torque sensors to modify the preplanned path and those which complete the insertion task without such feedback.

In the first category, the methods rely on an active compliance strategy [14]. An example of such an implementation was shown in [15] where a vision system was used for part location together with a neural network processing data from a force/torque sensor in the wrist joint. Jokesch et al. in [16] used a generic algorithm to automatically connect electric vehicles with charging stations using a robot in impedance control mode. This task requires alignment of seven pins and an asymmetric peg and hole. Abdullah et al. [17] used data models created from F/T sensor data to search for the

hole center using an algorithm based on human intuitive behavior. In [18], Bös et al. addressed limitations of admittance control schemes caused by the dependence of contact forces and torques on speed. Two iterative learning controllers were used to increase assembly speed and to reduce contact forces. Jasim et al. [19] used Gaussian Mixtures Model and F/T feedback in Contact-State modeling to identify accurate hole position. Xu et al. [20] divided the contact area around the hole into sixteen regions and used KD-Tree method to detect the right region for insertion using force feedback. While the use of force/torque sensors greatly improves insertion performance, this approach slows down the process, increases the cost of hardware, and may not be feasible when complex [21] or large scale parts [22] are mated.

In the second category, a passive or hybrid compliance strategy is applied, and a search algorithm is based on the profile of insertion depth [23-27]. For this type of strategy, rudimentary force feedback may be used only to ensure that a peg is in a constant contact with the surface of a part into which it is inserted. Strategies in both categories actively modify the preplanned trajectory as the search progresses and therefore, accuracy of the initial peg position is less critical for a successful insertion. However, an inaccurate starting point for the search algorithm will adversely affect the time spent in searching [28]. This positional accuracy depends on the quality of registration between the two coordinate frames.

Point-based rigid-body registration is a frequently used procedure to determine the transformation, rotation \mathbf{R} and translation $\boldsymbol{\tau}$, between two coordinate frames, for example: world or perception system frame and the robot frame. This procedure requires measurement of 3D common points in both frames. The procedure assumes that the rigid-body condition is satisfied, i.e. the distance between any two points in one frame is exactly the same as the distance between corresponding points in the second frame. However, due to noise and possible bias, this condition may not be preserved and corrective methods, like mentioned earlier RRBC, must be used.

3 Description of the RRBC Method

In this report, the first coordinate frame, from which points are transformed, is called the working frame (X). The second frame, to which points are transformed, is called the destination frame (Y). If the transformation $\{\mathbf{R}, \boldsymbol{\tau}\}$ from X to Y is known, it can be applied to the points of interest, $\mathbf{T}_X(k)$, $k = 1, \dots, K$, which are measured only in X but must be accessed in Y . We call these points target points. Due to registration error, targets transformed from the working frame are not mapped exactly in the destination frame. To decrease the error of a transformed k -th target, a correction vector $\boldsymbol{\varepsilon}(\mathbf{T}_X(k))$ is added to the target measured in X

$$\hat{\mathbf{T}}_X(k) = \mathbf{T}_X(k) + \boldsymbol{\varepsilon}(\mathbf{T}_X(k)) \quad (1)$$

where $\widehat{\mathbf{T}}_X(k)$ is the corrected target location in X . In RRBC, the correction $\boldsymbol{\varepsilon}(\mathbf{T}_X(k))$ is obtained by linear interpolation of corrections $\boldsymbol{\varepsilon}_n$ determined earlier at points \mathbf{X}_n which are close to target $\mathbf{T}_X(k)$ and is given by

$$\boldsymbol{\varepsilon}(\mathbf{T}_X(k)) = \sum_{n=1}^N w_n(\mathbf{T}_X(k), \mathbf{X}_n) \boldsymbol{\varepsilon}_n(\mathbf{X}_n) \quad (2)$$

where weights w_n are based on the proximity of \mathbf{X}_n to $\mathbf{T}_X(k)$, and $\sum w_n = 1$.

Corrections $\boldsymbol{\varepsilon}_n$ can be calculated from points $\mathbf{X}_n, n = 1, \dots, N$, measured in the working frame and the same points \mathbf{Y}_n measured in the destination frame

$$\boldsymbol{\varepsilon}_n = \mathbf{R}_{inv} \mathbf{Y}_n + \boldsymbol{\tau}_{inv} - \mathbf{X}_n \quad (3)$$

where $\{\mathbf{R}_{inv}, \boldsymbol{\tau}_{inv}\}$ is inverse of the registration transformation $\{\mathbf{R}, \boldsymbol{\tau}\}$, i.e., $\mathbf{R}_{inv} = \mathbf{R}'$ where \mathbf{R}' is the transposed rotation matrix and $\boldsymbol{\tau}_{inv} = -\mathbf{R}_{inv} \boldsymbol{\tau}$. Points measured in both frames are called fiducials, and the registration transformation $\{\mathbf{R}, \boldsymbol{\tau}\}$ was determined from at least three non-colinear fiducials using least-squares fitting algorithm [29].

If targets $\mathbf{T}_Y(k), k = 1, \dots, K$ are also measured in the destination frame (as in this research study), the Target Registration Error (TRE) can be calculated as

$$TRE(\mathbf{T}_X(k)) = \|\mathbf{R}\mathbf{T}_X(k) + \boldsymbol{\tau} - \mathbf{T}_Y(k)\| \quad (4)$$

A reliable metric of registration is based on a representative set of K targets measured in both frames [30] and can be calculated as the Root Mean Square of TRE (RMS_T)

$$RMS_T = \sqrt{\frac{1}{K} \sum_{k=1}^K TRE^2(k)} \quad (5)$$

4 Description of Experiments

A six degree of freedom (6DOF) collaborative robot arm (UR5) operating under high stiffness, in position control mode, was used to insert a peg into a series of holes. The speed of the robot was set to about 0.35 m/s (i.e., 35% of its maximum tool speed). Three experiments, Exp. 1 to 3, were conducted; each with slight variations. The coordinates of the holes were determined by registering a world frame to a robot frame using conventional, rigid-body registration and three fiducials for Exp. 1 and 2 and five fiducials for Exp. 3 to obtain $\{\mathbf{R}, \boldsymbol{\tau}\}$. After transforming the target locations from world frame to robot frame using $\{\mathbf{R}, \boldsymbol{\tau}\}$, uncorrected hole locations and locations corrected by the RRBC method were sent to the robot controller. Insertion of a peg in each target hole was attempted, and a simple Pass/Fail metric was used to determine the effectiveness of the RRBC method. A Pass meant that the robot in position control mode was able to fully insert the peg into the hole (as determined by monitoring the robot's z axis coordinate) with a measured force less than 17 N; otherwise, it was considered a Fail. It was also considered to have failed if the peg was fully inserted but jammed while extracting the peg.

Each of three experiments was repeated at least seven times. Before each experiment, the locations of all holes were re-measured with the robot in compliance mode and the transformation matrix $\{R, \tau\}$ and corrections ϵ_n were calculated. Then, for each repeat, the insertion test was performed for two transformed, uncorrected and corrected, target locations. A third insertion test was performed using the raw measured positions of the targets. These insertions reflect the time-consuming process of lead-through programming and were nominally expected to always be successful as these were the actual measured positions in the robot coordinate frame. Any failed insertion in this category is attributed to a non-zero noise in robot which affects its repeatability. The performance using these insertions served as a baseline.

The purpose of these experiments was to determine the effect of the RRBC corrected vs. uncorrected position on the peg insertion rate. The main reason for the large number of insertions (7 runs with 88 holes per run) was to check that any improvement is not due to random effects such as noise and drift (see non-zero baseline curve in Fig. 3). The influence of systematic factors such as robot speed or stiffness was not investigated in this study.

4.1 Equipment

In the experiments, a 12.675 mm (0.499 in) diameter peg was used and was rigidly attached to the robot arm (see Fig. 1c). In practice, the pegs would likely not be rigidly mounted to the robot and would likely be held in a gripper. This removes a source of variability encountered in practice which can be either beneficial or detrimental to the insertion task depending on the gripper.

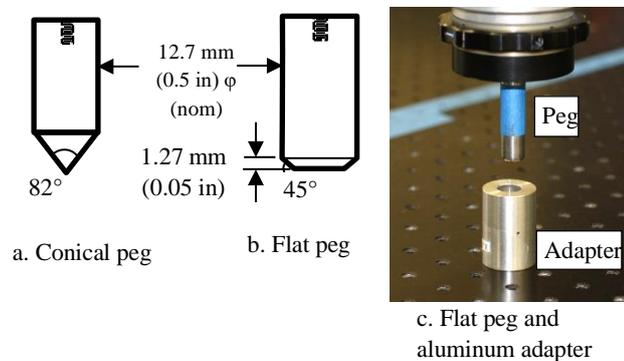


Fig. 1. Conical and flat pegs (a and b, respectively) and aluminum adapter (c).

For Exp. 1, the end of the peg was conical, and for Exp. 2 and 3, the end of the peg was flat with a slight chamfer (see Fig. 1a and b). From observations in Exp. 1, the peg was successfully inserted even though there was a visible initial misalignment of the peg and the hole; however, the “chamfer” of the conical peg allowed the peg to be inserted without exceeding the preset force threshold. After Exp. 1, it was decided that

a flat peg may make the distinction between Pass and Fail more definitive, and flat pegs are more typical in practice.

The diameter of the hole allowed for a 0.051 mm (0.002 in) clearance with respect to the peg. The hole was drilled into a 28.575 mm (1-1/8 in) diameter cylindrical piece of aluminum (Fig. 1c) and will be called an “adapter,” henceforth. Threaded holes at the bottom of the adapter enabled the attachment of the adapter to the optical plate.

The experiments were conducted using an optical breadboard plate mounted on a rigid aluminum frame. The plate had threaded holes spaced on a grid at 25 mm (0.98 in) intervals. The nominal locations of the holes were taken as ground truth locations. The work volume was essentially 2D with 17 rows by 21 columns of holes (Fig. 2).

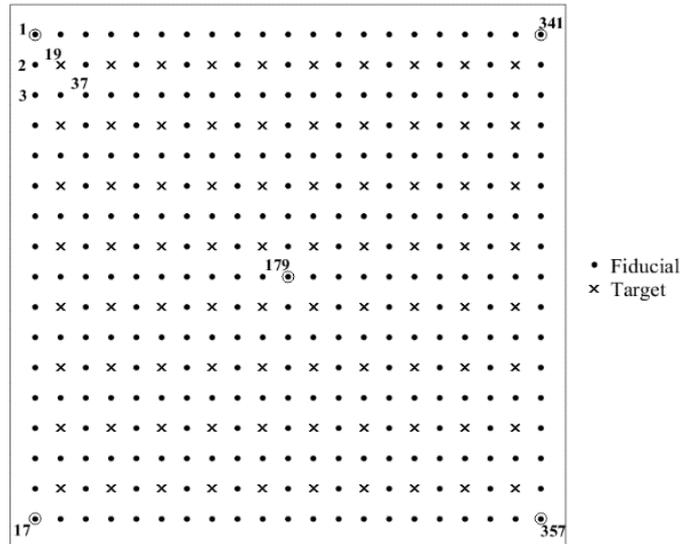


Fig. 2. Layout of the 17x21 grid of holes. Circles are fiducials used for registration. Dots represent Fiducials and crosses represent Targets in the figure.

4.2 General Procedure and Concept of Experiments

Given the manufacturing tolerances of the pre-drilled holes, the ground truth locations of the holes were set to the nominal locations of the holes in the world coordinate frame. The locations of all holes, in robot coordinate frame, were obtained by manually inserting the peg into the adapter with the robot in gravity compensation mode, and recording the values of $(x, y, z, \vartheta_x, \vartheta_y, \vartheta_z)$, where (x, y, z) is a Cartesian position, and $(\vartheta_x, \vartheta_y, \vartheta_z)$ are Euler angles representing the peg orientation. Each position was measured once. These manual measurements were obtained prior to the start of each experiment, and these measurements in robot frame along with the ground truth locations in world frame

are required to calculate corrections $\boldsymbol{\varepsilon}_n$ in Eq. (3) for the RRBC method.

Since the experiments described in this paper are essentially 2D, only corrections to the x- and y-coordinates were calculated, and the z-coordinate was unchanged from the raw measured value. No corrections were made to the peg orientations.

As seen in Fig. 2, targets were located in every other column and every other row, resulting in a total of 80 targets. Being a research study, the locations of the targets $\{\mathbf{T}_Y\}$ were measured beforehand in the robot coordinate frame using manual measurements – in practice, this may not be feasible and target locations are usually unknown. Target locations were treated as unknown positions in the calculations in this study to reflect practice, and the surrounding, manually-obtained fiducial locations were used in the calculations of corrections $\boldsymbol{\varepsilon}_n$. The measured target locations $\{\mathbf{T}_Y\}$ were only used to determine the registration error *TRE* in Eq. (4). The correction to a target location $\boldsymbol{\varepsilon}(\mathbf{T}_X)$ was estimated using $N = 8$ corrections $\{\boldsymbol{\varepsilon}\}_N$ of the fiducials closest to the target. The weights w_n in Eq. (2) were based on distances between the target and N nearest fiducials. In our experiments, the same set of weights was used for all targets, but in practical applications, the location of target \mathbf{T}_X relative to N nearest fiducials would vary.

The robot controller received $(x, y, z, \vartheta_x, \vartheta_y, \vartheta_z)$ as destination poses for all 80 target locations, where (x, y) were either the raw, transformed uncorrected, or transformed corrected Cartesian positions, and $(\vartheta_x, \vartheta_y, \vartheta_z)$ were either “Original” angles (raw measured values for a given position) or “Fixed” angles. The Fixed angles, $\bar{\vartheta}_x$ and $\bar{\vartheta}_y$, were simply the average of all 357 raw angles for ϑ_x and ϑ_y , respectively, where ϑ_x and ϑ_y determined the perpendicularity of the peg to the table. ϑ_z defined the rotation about the centerline of the peg which was irrelevant in our setup as we used symmetric, cylindrical peg.

A summary of the parameters used in the three experiments is given in Table 1. As noted in Table 1, each experiment consisted of either 7 or 12 runs. A run involved insertion of the peg into 80 hole locations.

Table 1. Summary of Parameters in Experiments.

Exp. #	# of runs	Peg type	D_p , Peg diameter mm (in)	D_h , Hole diameter mm (in)	$\Delta = D_h - D_p$ mm	Angles $(\vartheta_x, \vartheta_y)$	# of fiducials for regular registration
1	7	conical	12.675 (0.499)	12.751-12.776 (0.502-0.503)	(0.076-0.101)	original	3
	7	flat	12.675 (0.499)	12.751-12.776 (0.502-0.503)	(0.076-0.101)	original	3
2	7	flat	12.675 (0.499)	12.751-12.776 (0.502-0.503)	(0.076-0.101)	fixed	3
	7	flat	12.675 (0.499)	12.700-12.725 (0.500-0.501)	(0.025-0.05)	fixed	5
3	7	flat	12.675 (0.499)	12.700-12.725 (0.500-0.501)	(0.025-0.05)	fixed	5
	5	flat	12.675 (0.499)	12.700-12.725 (0.500-0.501)	(0.025-0.05)	fixed	5

5 Results and Discussion

5.1 Failed Insertions: Uncorrected vs. Corrected Target Locations

For each run, the number of times the peg failed to be inserted was determined, and the ratio of this number to the total number of insertions (80 targets) was calculated and plotted. The plots of the percent Fails for the three experiments are shown in Fig. 3. The data from these plots are summarized in Table 2. The columns labeled Unc /Cor in Table 2 quantify the improvement in performance of insertion task due to the RRBC corrections to the target locations. In Fig. 3 and Table 2, the baseline values are the results of attempted insertions using the raw (recorded in the robot's coordinate frame) positions. As mentioned in Section 4, there should be no failures for the baseline insertions, and the observed failures (about 2 % to 6 %) are likely due to measurement noise as the raw measurements were based on single measurements and to robot drift.

Fig. 3 shows that the RRBC increases the successful insertion rate as compared to the uncorrected targets locations. The average improvement was a 2.4 times reduction in the failure rate for the insertions with the Original angles (Exp. 1 and 2) and was approximately 11 and 4 times reduction in the failure rate for the Fixed angles, Exp. 2 and 3, respectively. This level of improvement should be analyzed in conjunction with the tolerances reported in Table 1, columns Peg diameter and Hole diameter.

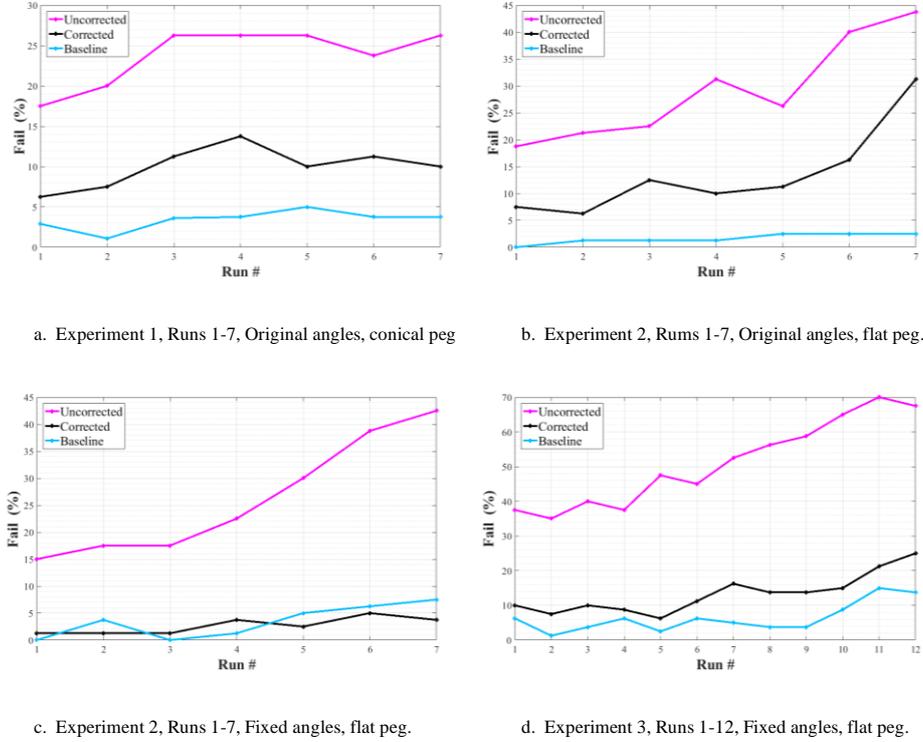


Fig. 3. Percentage of failed insertions for Experiments 1 -3.

Table 2. Percentage of Failed Target Insertions.

	Failed Insertions							
	Original Angles				Fixed Angles			
	Base-line (%)	Unc. (%)	Cor. (%)	Unc./Cor.	Base-line (%)	Unc. (%)	Cor. (%)	Unc./Cor.
EXPERIMENT 1								
Avg.	3.4	23.8	10.0	2.4				
Std. Dev.	1.2	3.6	2.5	0.3				
EXPERIMENT 2								
Avg.	1.6	29.1	13.6	2.4	3.4	26.3	2.7	11.0
Std. Dev.	0.9	9.6	8.5	0.7	3.0	11.0	1.5	3.0
EXPERIMENT 3								
Avg.					6.4	51.0	13.2	4.2
Std. Dev.					4.2	12.5	5.6	1.2

Another trend observed in Fig. 3 is the increase in the number of failures with time, observable even with the baseline trials. Some possible reasons for this increase were thermal effects as the day wore on and turning the robot on/off each day. Therefore, a third experiment was conducted. The first seven runs in Exp. 3 (Runs 1-7) were

conducted continuously in one shift without turning the robot off. However, as seen in Fig. 3d, Runs 1-7, the number of failures still increased with time despite the robot not being power-cycled. This increase is therefore likely due to drift in the robot hardware, as it also affects the baseline results, which do not depend on the quality of registration. At this time, the only explanation that can be offered for this increased number of failures is that the robot, sensors, and mechanics, drift with the number of performed insertions. The observed drift should not be interpreted as a process of asymptotic approach to steady state but rather as a systematic departure from initial conditions due to the performance of a repeated task. However, even as the repeated use of robot increased the number of failures, the use of the RRBC was always beneficial.

After the initial seven runs in Exp. 3, five additional runs were performed over the next three days. As seen in Fig. 3d, the number of failures did not seem to be leveling off, and after 12 runs the percent of failures was about 70 %. It needs to be stressed that the slow drift of robot parameters makes the calculated corrections ϵ_n in Eq. (3) progressively less accurate and increases the failure rate. Given the tight tolerances of the insertions, this drift need not be large to impact trial performance, and insertions with looser tolerances may not experience such failure growth trends.

5.2 Relationship between Failures and TRE

In Fig. 4, the normalized failure values for each target are plotted against the differences $\Delta_{TRE}(\mathbf{T}_X) = TRE(\mathbf{T}_X) - TRE(\hat{\mathbf{T}}_X)$ where $TRE(\mathbf{T}_X)$ and $TRE(\hat{\mathbf{T}}_X)$ are the TRE s for the uncorrected and corrected targets locations, respectively. The normalized failure value, $\Delta_{Fail,nom}(\mathbf{T}_X)$, is the difference between the number of failed insertions for uncorrected target location \mathbf{T}_X and the number of failed insertions for corrected target $\hat{\mathbf{T}}_X$ divided by the number of runs for that experiment. If $\Delta_{Fail,nom} > 0$ then more failed insertions are observed at uncorrected target location \mathbf{T}_X and if $\Delta_{Fail,nom} < 0$ then more failed insertions were observed for target locations corrected by RRBC. The plots are divided into four quadrants:

1. Q-I, upper left: RRBC increases TRE , reduces failed insertions
2. Q-II, upper right: RRBC reduces TRE , reduces failed insertions
3. Q-III, lower left: RRBC increases TRE , increases failed insertions
4. Q-IV, lower right: RRBC reduces TRE , increases failed insertions

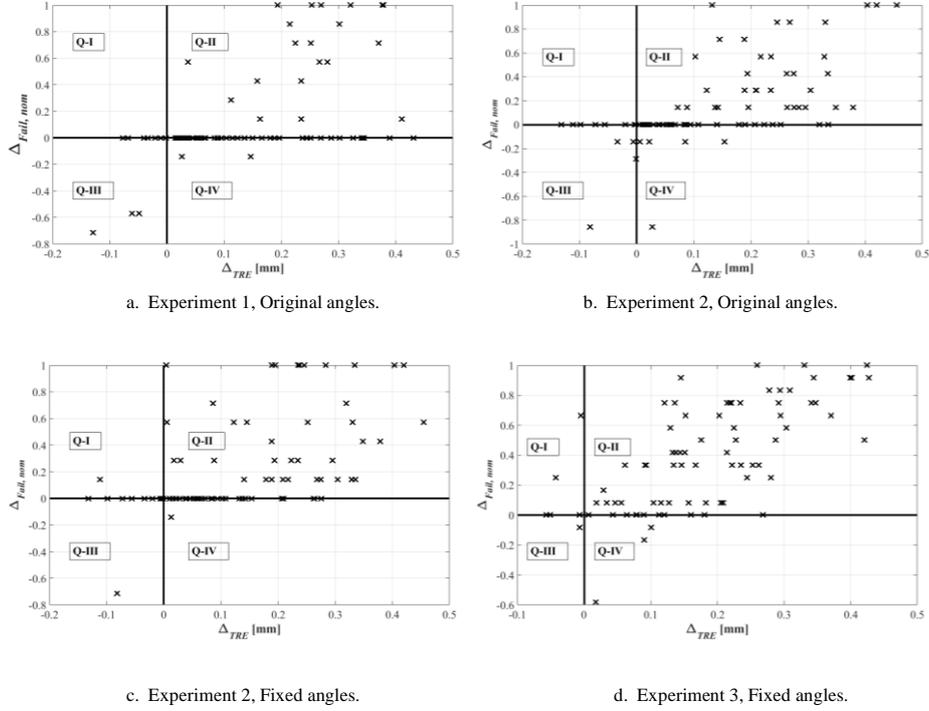


Fig. 4. Failures and TRE.

As seen in Fig. 4, the RRBC method either did not adversely affect the number of failures, or it reduced number of failures (Quad II). Also, the RRBC method reduced TRE for a majority of the targets (targets with $\Delta_{TRE} > 0$).

For Quads I, III, and IV, two possible explanations are offered for why the RRBC method increases TRE and/or increases failures. One reason is that the correction for the target is estimated based on the data of neighboring points. As discussed in [6], the data contains both noise and systematic bias. Although the noise is smaller than the bias, it affects the calculated corrections. The other reason is that linear interpolation is used to determine the corrections, and the correction may not be linear.

RMS_T (which is related to TRE) was calculated using Eq. (5) and is shown in Fig. 5. As seen in Fig. 5, the RRBC method reduces RMS_T and improves the registration performance. The average reduction of the RMS_T is about 60 % when using the RRBC method compared to not using the RRBC.

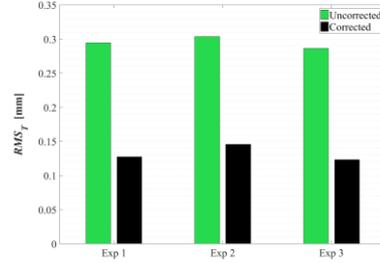


Fig. 5. Comparison of RMS_T for Experiments 1-3.

5.3 Tighter Tolerances

Because of the drift in the failure rate with time as described in Section 5.1, a valid comparison of failure rates between the corrected and uncorrected targets locations can only be made for the first runs in Exp. 2 and 3. From Fig. 3c, failure rates for Exp. 2 are 15 % and 1 % for the first run for the uncorrected and corrected by RRBC target locations, respectively. From Fig. 3d, failure rates for Exp. 3 are 37 % and 10 % for the first run for the uncorrected and corrected locations, respectively. The higher failure rate for the uncorrected locations contradicts the expectation that the use of five fiducials (Exp. 3) would result in a better registration than the registration with three fiducials (Exp. 2). A reason for the higher rate of failures for the uncorrected locations in Exp. 3 was because the diameter of the hole in the adapter used in Exp. 2 was between 12.751 mm and 12.776 mm (0.502 in and 0.503 in) while the diameters of the holes in the adapters used for Exp. 3 ranged from 12.700 mm to 12.725 mm (0.500 in to 0.501 in). The diameter of the flat peg was 12.675 mm (0.499 in) at the start of Exp. 2 and 12.662 mm (0.4985 in) after Exp. 3. Thus, the higher failure rate for the uncorrected locations, an increase of about 22 %, in Exp. 3 can be attributed to the tighter tolerance – a decrease of 0.025 mm (0.001 in). The tighter tolerance increased the failure percentage from 1 % (1st run, Exp. 2) to 10 % (1st run, Exp. 3) for the RRBC method.

The efficacy of the RRBC method applied to an insertion task depends strongly on the difference Δ (see Table 1) between the hole and peg diameters and the corrected mean target registration error RMS_T . If the peg diameter D_p is very small relative to the hole diameter D_h , then there is no benefit to using the RRBC method as the insertion will always be successful (unless there is a very large uncorrected error RMS_T comparable with the hole diameter D_h). In the other extreme when $D_p = D_h$, the use of the RRBC method is futile because the insertion will always fail. Thus, the largest benefit of using the RRBC method to improve the insertion task is when the difference Δ is comparable with the corrected RMS_T . In Exp. 1 and 2, the ratio Δ/RMS_T was between 0.536 and 0.842, for Exp. 3 the ratio was between 0.208 and 0.417, based on values of Δ in Table 1 and values of RMS_T shown in Fig. 5.

Another observation is that in Exp. 3, the percent of failed insertions for the uncorrected locations started out at about 37 %. Since the baseline value was about 6 % (i.e., 6 % of the failures is due to noise), 31 % of the Fails was due to the registration error. This rate of failure may be unacceptable for many applications. Thus, for insertions where tight tolerances are needed, the RRBC method may have to be used, as only 4 % of the failure can be attributed to registration error, as shown in Fig. 3d, Run 1.

6 Conclusions

Peg-in-hole experiments were performed to quantify reduction in the failed insertion rate when the target registration error is reduced by using the RRBC method. No search algorithms or force/torque sensors were used to aid the insertion task. The successful insertion of the peg depended almost entirely on the accuracy of the hole location which, in turn, depends on the quality of world to robot registration.

The RRBC method improves registration by reducing the registration error, RMS_T , by 60 % when compared to uncorrected targets locations. This study shows that this level of improvement in positional accuracy led to a 11-fold reduction in the failed insertion rate and a 4-fold reduction for a tighter tolerance.

Manufacturing tasks which require accurate positional data would be more efficient if the registration process was complemented by the application of small corrections to the target locations. The use of the RRBC method may be especially useful for part assembly with tight tolerances where search algorithms are commonly used. Reduced registration error should result in a starting point closer to the desired point, and thus, lead to a shorter search time and smoother assembly. Other tasks such as drilling, welding, or inspection where traditional control techniques based on force/torque feedback are not available, should also benefit from reduced positional error.

7 Disclaimer

Certain commercial equipment are mentioned in this paper to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the equipment is necessarily the best available for the purpose.

References

1. J.L.Nevins and D.E.Whitney, *Computer-Controlled Assembly*. Scientific American, 1978. **238**: p. 62-74.

2. M.W.Abdullah, et al., *An Approach for Peg-in-Hole Assembling using Intuitive Search Algorithm based on Human Behavior and Carried by Sensors Guided Industrial Robot*. IFAC-PapersOnLine, 2015. **48**(3): p. 1476-1481.
3. Newman, W.S., Y. Zhao, and Y.-H. Pao, *Interpretation of Force and Moment Signals for Compliant Peg-in-Hole Assembly*, in *Int. Conference on Robotics & Automation ICRA*. 2001, IEEE. p. 571-576.
4. T.Yamashita, et al. *Peg-and-hole task by robot with force sensor: Simulation and experiment*. in *Proc. Int. Conf. on IECON*. 1991.
5. L.L.Lin, et al. *Peg-in-Hole assembly under uncertain pose estimation*. in *Proc. 11th World Congress on Intelligent Control and Automation 2014*. Shenyang, China.
6. M.Franaszek and G.S.Cheok, *Improving Rigid-Body Registration Based on Points Affected by Bias and Noise*, in *Precision Assembly in the Digital Age*, S.Ratchev, Editor. 2018, Springer: Chamonix, France. p. 226 - 234.
7. Bristow, D.A., M. Tharayil, and A.G. Alleyne, *A Survey of Iterative Learning Control*. IEEE Control Systems Magazine, 2006. June p. 96-114.
8. Galetto, M., et al. *Volumetric Error Compensation for the MScMS-II*. in *12th CIRP Conference on Computer Aided Tolerancing*. 2013. Elsevier.
9. Cajal, C., et al., *Efficient volumetric error compensation technique for additive manufacturing machines*. Rapid Prototyping Journal, 2016. **22**(1): p. 2 - 19.
10. S.Huang, et al., *Applying High-Speed Vision Sensing to an Industrial Robot for High-Performance Position Regulation under Uncertainties*. Sensors, 2016. **16**.
11. W.Zhu, et al., *An off-line programming system for robotic drilling in aerospace manufacturing*. Int. J. Advanced Manufacturing Technology, 2013. **68**: p. 2535–2545.
12. L.Tingelstad, et al., *Multi-Robot Assembly of High-Performance Aerospace Components*, in *10th IFAC Symposium on Robot Control*. 2012: Dubrovnik, Croatia. p. 670-675.
13. G.S.Cheok, et al., *Improving Automated Insertion Applications by Restoring Rigid-Body Condition in Point-Based Registration*. National Institute of Standards and Technology Report NISTIR **8198r1**, 2017.
14. Whitney, D.E., *Quasi-Static Assembly of Compliantly Supported Rigid Parts*. Journal of Dynamic Systems, Measurement, and Control, 1982. **104**(1): p. 65-77.
15. J.L.Navarro-Gonzalez, et al., *On-line incremental learning for unknown conditions during assembly operations with industrial robots*. Evolving Systems, 2015. **6**(2): p. 101-114.
16. Jokesch, M., et al., *Generic Algorithm for Peg-In-Hole Assembly Tasks for Pin Alignments with Impedance Controlled Robots*, in *Robot 2015: Second Iberian Robotics Conference: Advances in Robotics*, L.Reis and al., Editors. 2016, Springer International Publishing: Cham. p. 105-117.
17. M.W.Abdullah, et al. *Force/Torque Data Modeling for Contact Position Estimation in Peg-in-Hole Assembling Application*. in *Proc. of 4th Int. Conf. on Advances in Mechanics Engineering*. 2015. MATEC Web of Conferences.
18. J.Bos, A.Wahrburg, and K.D.Listmann, *Iteratively Learned and Temporally Scaled Force Control with Application to Robotic Assembly in Unstructured*

- Environments*, in *IEEE Int. Conf. on Robotics and Automation (ICRA)*. 2017: Singapore. p. 3000-3007.
19. I.F.Jasim, P.W.Plapper, and H.Voos. *Position Identification in Force-Guided Robotic Peg-in-Hole Assembly Tasks*. in *Conference on Assembly Technologies and Systems, Procedia CIRP*. 2014. Elsevier.
 20. Y.Xu, Y.Hu, and L.Hu, *Precision Peg-in-Hole Assembly Strategy Using Force-Guided Robot*, in *3rd Int. Conf. on Machinery, Materials and Information Technology Applications (ICMMITA)*. 2015: Qingdao, China.
 21. H.C.Song, Y.L.Kim, and J.B.Song, *Guidance algorithm for complex-shape peg-in-hole strategy based on geometrical information and force control*. *Advanced Robotics*, 2016. **30**(8): p. 552-563.
 22. Z.Liu, et al. *Laser tracker based robotic assembly system for large scale peg-hole parts*. in *Proc. 4th IEEE Annual Int. Conf. on Cyber Technology in Automation, Control, and Intelligent Systems*. 2014. Hong Kong, China.
 23. M.P.Polverini, et al., *Sensorless and Constraint Based Peg-in-Hole Task Execution with a Dual-Arm Robot*, in *ICRA*. 2016, IEEE: Stockholm, Sweden. p. 415-420.
 24. H.Park, et al., *Compliance-Based Robotic Peg-in-Hole Assembly Strategy Without Force Feedback*. *IEEE Trans. Industrial Electronics*, 2017. **64**(8): p. 6299-6309.
 25. R.C.Luo, A.Chang, and Ch.Li, *A Novel Peg-in-Hole Approach based on Geometrical Analysis for Inclined Uncertainty*, in *IEEE Int. Conf. on Advanced Intelligent Mechatronics (AIM)*. 2017: Munich, Germany. p. 891-896.
 26. J.H.Su, et al., *Study on dual peg-in-hole insertion using of constraints formed in the environment*. *Industrial Robot: An International Journal*, 2017. **44**(6): p. 730-740.
 27. K.Sharma, V.Shirwalkar, and P.K.Pal, *Peg-In-Hole search using convex optimization techniques*. *Industrial Robot-an International Journal*, 2017. **44**(5): p. 618-628.
 28. Wyk, K.V., et al., *Comparative Peg-in-Hole Testing of a Force-based Manipulation Controlled Robotic Hand*. *IEEE Trans. on Robotics*, 2018.
 29. Arun, K.S., T.S. Huang, and S.D. Blostein, *Least-Squares Fitting of Two 3-D Point Sets*. *IEEE Trans. PAMI*, 1987. **9**(5): p. 698-700.
 30. Fitzpatrick, J.M., J.B. West, and C.R. Maurer, *Predicting Error in Rigid-Body Point-Based Registration*. *IEEE Trans. Medical Imaging*, 1998. **17**(5): p. 694 -702.