

Simplified Framework for Robot Coordinate Registration for Manufacturing Applications

Jeremy A. Marvel Karl Van Wyk

Abstract—A simplified framework is introduced for automatically and quickly registering the Cartesian coordinate systems of industrial robots to any other arbitrary coordinate system. This framework includes both explicit and implicit (sensor-based) registration techniques using as few as three reference poses per robot, and presents different methods for measuring registration uncertainty. Driven by the guiding principles of simplifying the registration process to enable rapid installation by non-expert users, a mathematical basis for fast system registration is presented. We also present methods for quickly and inexpensively approximating the registration errors, and outline mechanisms for improving registration performance. Several case study examples are provided in which the registration performance is captured across four different registration methods, and two different robots. A reference motion capture system is used to capture post-registration positioning accuracy of the robots, a sampling-based registration estimation technique is assessed, and results are systematically quantified.

Keywords: robot registration, robot performance, multi-robot coordination

I. INTRODUCTION

Flexible robot automation encapsulates the idea that a single robot can be quickly and repeatedly reconfigured and retasked to accommodate multiple applications. With the current trends in lightweight and mobile robotic systems, the versatility of such robotic platforms is increased further with their ability to be rapidly relocated to new workstations for new applications. Moreover, with the advent of collaborative robot technologies, the historical need for hard mounts and safeguards is relaxed. However, such flexibility introduces new performance challenges for manufacturing applications.

Relocating or re-tasking robots requires the reprogramming of hard-coded, application-dependent robot positions in their respective coordinate systems. This brittle programming approach impacts the productivity of the manufacturing line by undermining software reusability. This inflexibility makes robot adoption by small- and medium-sized enterprises (SMEs) both needlessly complicated and prohibitively expensive. Two different schools of thought seek to minimize this impact. The first seeks to reduce the barrier to programming, effectively making the robots easier to program for targeted applications (e.g., [1]). The second removes the need to re-program robot positions by programming robot positions in the workspace or world coordinate frame (to

which the robot registers itself), e.g., [2]. The first method pushes the process intelligence onto the robot's controller, and limits the functionality of robots to specific applications for which software has already been written. The second method requires quality registration to the world coordinate frame, which can be a burdensome process to ensure the minimization of registration errors.

The registration process and associated errors have been identified as one particular pain point for SMEs [3]. Coordinate system registration refers to the mapping of measurements in one system's frame of reference to the frame of reference of another. This process corrects for differences in coordinate system placement and orientation. In this paper, a framework is presented that reduces this burden by providing a mechanism for easy and automatic coordinate system registration with an equally easy method for estimating registration uncertainty. This framework automatically computes the homogeneous transformation from one coordinate system of any robot to any other arbitrary coordinate system and then measures the registration uncertainty using simple math and fast pose sampling. This work is inspired by the desire to make robotic solutions for smart manufacturing more accessible. It aims to do so by enabling multiple robots to be more flexible and adaptive, such that robot systems can be installed, configured, and programmed to operate quickly and with minimal negative impact on productivity. Further, this work provides a necessary precursor to enabling heterogeneous configurations of multiple robots (i.e., robots from different manufacturers with incompatible controllers) to work collaboratively on a single task.

This paper is split into the following sections: Section II provides a brief account of existing, commercial solutions for robot coordinate frame registration; Section III gives an outline of the registration process; Section IV enumerates various methods for acquiring the necessary input data for coordinate frame registration; Section V presents experimental results of four different registration methods across two different robotic platforms that demonstrate functionality, feasibility, and validity of the framework; and Section VI gives a discussion of results and future work.

II. ROBOT COORDINATE SYSTEMS AND REGISTRATION

In robotic applications, registration transforms poses from one coordinate system to poses in the robot's coordinate system. This transformation facilitates the hand-off of parts from one robot to another, and converts measurements from external camera systems to guide path planning for part acquisition and collision avoidance. For many industrial

Jeremy A. Marvel is with the National Institute of Standards and Technology, Gaithersburg, MD, USA (e-mail: jeremy.marvel@nist.gov).
Karl Van Wyk is with the National Institute of Standards and Technology, Gaithersburg, MD, USA (e-mail: karl.vanwyk@nist.gov).
Official contribution of the National Institute of Standards and Technology; not subject to copyright in the United States.

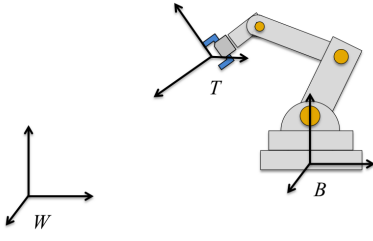


Fig. 1: Typical coordinate systems for industrial robots include the base (B) and tool (T) frames, and may also include external definitions (W) for world or workpiece frames.

robot systems, 6 degree-of-freedom (6DoF) poses (translations and rotations in Cartesian space) are typically defined in either the base frame or the tool frame (Fig. 1). The base frame, B , is usually located with its origin inside the base at or near the mounting flange. The tool frame, T , is located with its origin at the tool center point (TCP). Some robot vendors provide mechanisms for defining additional coordinate systems to enable a unifying world frame, W , for multiple robots or sensors. Furthermore, functionality may allow for the creation of a work frame (a task-relevant coordinate system) to which the robot is registered such that a robot or workpiece may be moved without requiring retraining of all hard-coded robot positions. Unfortunately, these capabilities are not universally available.

The process of registering between sensor systems is required for measuring more complex information about the world than is possible with a single sensor (e.g., measuring 3D point clouds using multiple 2D cameras [4]). The registration between a sensor system and a robot similarly enables more complex control capabilities. Depending on the level of integration, registering sensors to a robot coordinate system requires significant effort on the part of the system programmer. Sensor systems specifically designed for use with automation may provide the capability to define their own coordinate systems relative to the robot to minimize the effort needed for registration (e.g., [5]). Additionally, there exist some approaches in the literature to automatically register robot systems with external and robot-mounted (e.g., [6]) camera systems, though their use in general manufacturing applications is limited.

Registering multiple robots to accept command coordinates in a common coordinate system requires either a central point of control (e.g., [7]) or the provision of external world or work frames (e.g., [8]). Some multi-robot systems in the literature build internal representations of maps, and registration takes the form of aligning these maps using techniques common to machine vision [9]–[11].

III. 3-POINT REGISTRATION PROCESS

At the heart of our registration framework is a basic procedure for calculating a homogeneous transformation from one coordinate system (N) to another (R) by means of a third, intermediate coordinate system (O , Fig. 2). Such transformations may be used to register one robot's coordinate system to

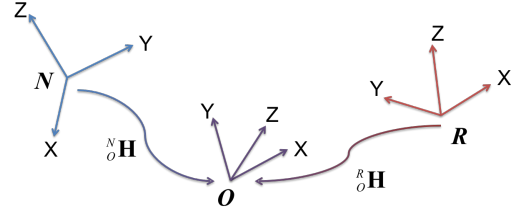


Fig. 2: The transformation from one arbitrary coordinate system, R , to another, N , relies on defining an intermediate coordinate system, O , with a known transformation to and from R and N .

the coordinate system of another, or to register both robots' coordinate systems to a third coordinate system. This process consists of capturing 3D coincidental Cartesian points (X , Y , and Z coordinates) in both the N and R coordinate systems, and then constructing an arbitrary coordinate system, O , such that known homogeneous transformations exist from O to N , and from O to R . To accomplish this, the following assumptions must be satisfied:

- N and R are static (or have known transformations to static reference frames) for the collection of the registration data,
- The systems operating within N and R are calibrated to be within the users specifications for accuracy and repeatability, and
- All systems follow a right-handed orientation.

The process begins with the selection of at least three non-collinear points, \mathbf{p}_1 , \mathbf{p}_2 , \mathbf{p}_3 , that can be measured in both the robot and world coordinate systems (Fig. 3a). Specifically, for each $\mathbf{p}_i = [p_{i,x}, p_{i,y}, p_{i,z}]^T$, $1 \leq i \leq 3$, in the R coordinate system, there is a corresponding, co-located \mathbf{p}_i in the N coordinate system. From these two sets of points, corresponding vectors, \mathbf{v}_1 and \mathbf{v}_2 , linking \mathbf{p}_1 to \mathbf{p}_2 and \mathbf{p}_1 to \mathbf{p}_3 (Fig. 3b), respectively, are defined in both coordinate systems by:

$$\mathbf{v}_1 = \mathbf{p}_2 - \mathbf{p}_1, \quad (1)$$

$$\mathbf{v}_2 = \mathbf{p}_3 - \mathbf{p}_1, \quad (2)$$

Definitions for \mathbf{v}_1 and \mathbf{v}_2 are given in terms of the both the R and N coordinate systems (i.e., ${}^R\mathbf{v}_1$ and ${}^N\mathbf{v}_1$). These two vectors provide the bases for defining the intermediate right-hand coordinate system, O . The unit vector $\hat{\mathbf{x}}$ defines the X axis of O , and is aligned with \mathbf{v}_1 in both N and R (Fig. 3c).

$$\hat{\mathbf{x}} = \frac{\mathbf{v}_1}{\|\mathbf{v}_1\|} \rightarrow {}^N\hat{\mathbf{x}}, {}^R\hat{\mathbf{x}} \quad (3)$$

The unit vector $\hat{\mathbf{y}}$ defines the Y axis of O , and is orthogonal to both \mathbf{v}_2 and $\hat{\mathbf{x}}$ (Fig. 3d).

$$\hat{\mathbf{y}} = \frac{\hat{\mathbf{x}} \times \mathbf{v}_2}{\|\hat{\mathbf{x}} \times \mathbf{v}_2\|} \rightarrow {}^N\hat{\mathbf{y}}, {}^R\hat{\mathbf{y}} \quad (4)$$

And the unit vector $\hat{\mathbf{z}}$ defines the Z axis of O , and is orthogonal to both $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ (Fig. 3e).

$$\hat{\mathbf{z}} = \hat{\mathbf{x}} \times \hat{\mathbf{y}} \rightarrow {}^N\hat{\mathbf{z}}, {}^R\hat{\mathbf{z}} \quad (5)$$

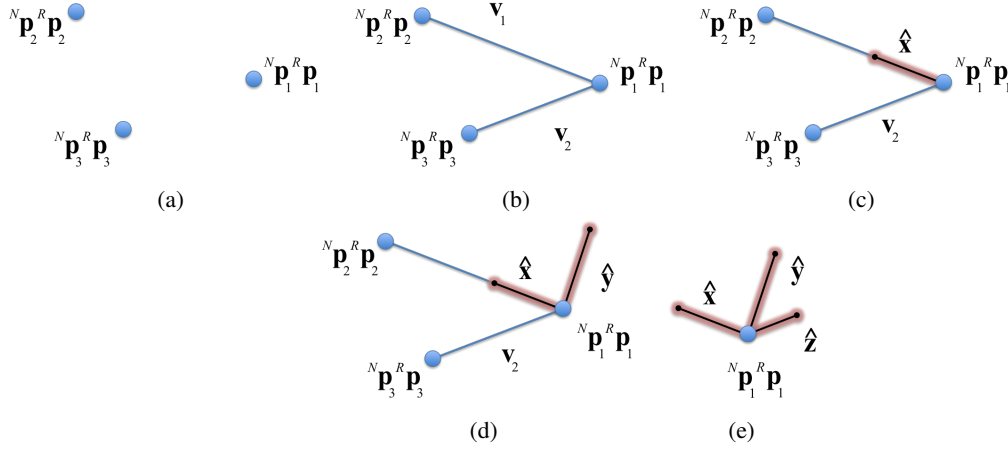


Fig. 3: The registration process is centered around the establishment of an intermediate coordinate system, O , using three, non-collinear sets of points (a). These three point clusters, and the vectors defining the edges connecting them (b), are then used to generate unit vectors (c-e) that establish the coordinate system O .

Given that $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, and $\hat{\mathbf{z}}$ are orthonormal, they define the orientation of O in both N and R :

$${}^N\mathbf{R} = \begin{bmatrix} {}^N\hat{\mathbf{x}} & {}^N\hat{\mathbf{y}} & {}^N\hat{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} {}^N\hat{\mathbf{x}}_x & {}^N\hat{\mathbf{y}}_x & {}^N\hat{\mathbf{z}}_x \\ {}^N\hat{\mathbf{x}}_y & {}^N\hat{\mathbf{y}}_y & {}^N\hat{\mathbf{z}}_y \\ {}^N\hat{\mathbf{x}}_z & {}^N\hat{\mathbf{y}}_z & {}^N\hat{\mathbf{z}}_z \end{bmatrix}, \quad (6)$$

$${}^R\mathbf{R} = \begin{bmatrix} {}^R\hat{\mathbf{x}} & {}^R\hat{\mathbf{y}} & {}^R\hat{\mathbf{z}} \end{bmatrix} = \begin{bmatrix} {}^R\hat{\mathbf{x}}_x & {}^R\hat{\mathbf{y}}_x & {}^R\hat{\mathbf{z}}_x \\ {}^R\hat{\mathbf{x}}_y & {}^R\hat{\mathbf{y}}_y & {}^R\hat{\mathbf{z}}_y \\ {}^R\hat{\mathbf{x}}_z & {}^R\hat{\mathbf{y}}_z & {}^R\hat{\mathbf{z}}_z \end{bmatrix}. \quad (7)$$

And because the origin of O is centered on \mathbf{p}_1 , the homogeneous transformations from the origins of O to the origins of N and R , respectively, are defined as

$${}^N\mathbf{H} = \begin{bmatrix} {}^N\mathbf{R} & {}^N\mathbf{p}_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (8)$$

$${}^R\mathbf{H} = \begin{bmatrix} {}^R\mathbf{R} & {}^R\mathbf{p}_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (9)$$

Conversely, the homogeneous transformation from R to O is defined simply as the inverse of Eq. 9:

$${}^O\mathbf{H} = ({}^R\mathbf{H})^{-1} \quad (10)$$

The homogeneous transformation matrix, ${}^N\mathbf{H}$, takes a Cartesian point in the R coordinate system and transforms it into the N coordinate system, effectively registering the two coordinate systems together.

$${}^N\mathbf{H} = {}^O\mathbf{H}{}^R\mathbf{H} \quad (11)$$

Any arbitrary 6DoF pose, expressed as a homogeneous matrix, ${}^R\mathbf{H}_j$, in R is thus transformed to N by multiplying it by ${}^N\mathbf{H}$:

$${}^N\mathbf{H}_j = {}^N\mathbf{H}{}^R\mathbf{H}_j \quad (12)$$

Similarly, any arbitrary 6D pose in N can be transformed back to R by multiplying it by the inverse:

$${}^R\mathbf{H}_j = ({}^N\mathbf{H})^{-1} {}^N\mathbf{H}_j \quad (13)$$

IV. POSE MEASUREMENTS

All registration information is derived from the ability to measure poses in Cartesian space. A software package can be written to take only the initial set of pose data, $P = \{^R\mathbf{p}_1, ^N\mathbf{p}_1, ^R\mathbf{p}_2, ^N\mathbf{p}_2, ^R\mathbf{p}_3, ^N\mathbf{p}_3\}$, and automatically output ${}^N\mathbf{H}$ or provide a mechanism to convert any arbitrary pose in N or R to its respective counterpart. The requirements of the user are thus reduced to simply generating the pose data. In this section, a number of mechanisms for retrieving this data are outlined briefly, and many sources of measurement uncertainty associated with each mechanism are enumerated.

A. Manual Entry

The most direct (and potentially the least accurate and precise) mechanism for providing pose information is manually entering X , Y , and Z values. These measurements may be directly measured using external tools such as 1) digital or analog distance and angle indicators (e.g., measuring tape, calipers, and protractors), 2) manual transference of information acquired directly from the systems (e.g., transcribing pose measurements as displayed on a teach pendant or other human-computer interface (HCI) devices), or 3) through the design of the working environment (e.g., placing fiducials at positions known *a priori*, or using design features of environment components as measurement cues).

As the robot is moved throughout the work volume, the TCP poses are manually recorded. This process may be overly tedious and time consuming, particularly if multiple coordinate systems must be registered together. Moreover, manually measuring positions is prone to precision and measurement errors, which impacts the performance of the registration as a whole. Similarly, maneuvering through the design of the working environment may result in unexpected registration errors due to the unverified assumptions of product quality control. Specifically, extrusions, cuts, and connections may not be perfect, thus the assumptions of flatness, orientation, position, and distribution of components

need to be independently measured to compensate for small construction defects. Accuracy and precision may be improved by means of precision metrology equipment, but at the expense of increased cost and complexity.

B. Direct Kinematic Interfacing

Many robot systems allow for direct feedback access via remote connection to pose and state information. This information may be provided with higher precision than the readouts on HCI devices, and enables near instantaneous information regarding the robot's Cartesian pose, joint configuration, and input/output states. For multi-robot configurations in which the robots must register themselves automatically to one another, such functionality is required such that the robots can convey pose information directly.

It has been empirically observed with some robot systems that the robot state reported via the remote interfaces is not necessarily identical to the information displayed on the HCI. In some cases it was observed that this discrepancy was due to approximation errors of the HCI. In other instances, however, it was discovered that the errors originated from the discrepancies in reporting mechanisms. In general, measurement errors arise if the remotely accessed robot state is not provided in real time, if the information reported is stale, or if the pose data is not reported at a reliable frequency. Moreover, there may exist errors in the robot's kinematic model, which directly impacts positioning accuracy. Some compensation for such errors may be provided through externally tracking the robot, or correcting for uncertainty using noise models.

C. Force-based Center of Pressure (CoP)

Perhaps a more flexible method for collecting points required for spatial registration is through the use of a 6DoF force/torque transducer (load cell). When interacting with a load cell at a single point of application, the center of pressure may be estimated using the load cell's sensory data. For a rigid body, the well-known relationship between an extrinsically acting force $\mathbf{F} = [F_x, F_y, F_z]^T$, at a single point $\mathbf{p} = [p_x, p_y, p_z]^T$, and the induced moment $\mathbf{M} = [M_x, M_y, M_z]^T$ is defined as,

$$\mathbf{M} = \mathbf{p} \times \mathbf{F}. \quad (14)$$

Performing the cross product, a system of three scalar equations is established in matrix form,

$$\begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} 0 & -F_z & F_y \\ F_z & 0 & -F_x \\ -F_y & F_x & 0 \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}, \quad (15)$$

where \mathbf{M} and \mathbf{F} are known via the load cell. This system of linear equations has infinitely many solutions since the matrix of forces is rank deficient. Therefore, at least one element of \mathbf{p} must be known or one element can be expressed in terms of the remaining two elements using geometric information of the rigid body attached to the load cell. In the

simplest case, a rigid plate of known, uniform thickness can be attached to the load cell to render one of the elements of \mathbf{p} as known. Consequently, only two of the previous equations need to be solved to obtain the remaining elements of \mathbf{p} . A robot needs to interact with the load cell to yield at least three, noncollinear points, which is the minimum number necessary for spatial registration.

Given that there is more freedom in readily generating points for registration, one should avoid the following pitfalls: 1) do not generate all the contact points in a line, and 2) do not cluster contact points extremely close to one another as that makes the registration process more susceptible to sensory noise in the load cell. Ultimately, the accuracy of center of pressure estimation is dictated by the load cell's calibration, resolution, and noise level.

D. Camera-Based Localization

Camera-based observer systems are common within manufacturing for localizing parts and workpieces for processing. In some cases, similar systems can be used for monitoring the positions of robot systems for performance validation, collision avoidance, and coordinate system registration. In both instances, the camera systems and robots must be registered such that one can operate within the coordinate system of the other. In this subsection, a number of camera-based solutions for measuring the 6DoF pose of robot systems are described.

1) *Fiducial-Based Localization*: Rather than directly measuring the 6DoF pose of a robot, its pose may be inferred by measuring the locations of multiple target fiducials affixed to the robot, the robot's tooling, or a held workpiece. Such fiducials include 2D barcodes [12], passive retroreflective or active infrared emitter markers for motion capture systems, or artifacts that look different based on viewing angle [13].

Such mechanisms and artifacts are widely available and have documented performance benefits and limitations. For instance, 2D barcodes are cheaper and easier to distinguish than retroreflective markers due to their unique designs, but are subject to lighting and perspective issues. Similarly, commercial off-the-shelf motion capture systems are inherently easier to configure, calibrate, and integrate, but are negatively impacted by sources of light interference and marker size (smaller markers are harder to see, but the available density of larger markers within a given region is reduced). In all cases, the performance of the fiducial-based tracking system is subject to errors in the intrinsic and extrinsic camera calibrations, visual occlusions in the sensing volume, and decreased accuracy and precision as a function of fiducial size and distance from the cameras.

2) *Shape-Based Localization*: Markerless robot tracking methods include fitting robot models to 3D point cloud data [14], [15], evaluation of depth-space data to measure distance [16], and shape-based estimation of robot pose [17]. Some low-cost sensor solutions intended as input devices for HCI have been used as measurement systems (e.g., [18]), but are not intended for the context of robot registration. Subsequently, they are subject to errors from manufacturing quality, low sensor resolution, and shortcuts made both to

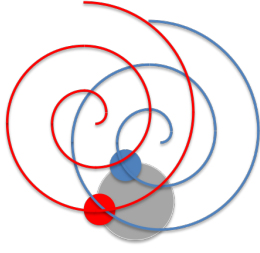


Fig. 4: Some search-based localization routines may halt as soon as a target is “found,” which may occur at the periphery of the target. Initiating the search at different initial poses may have different localization results.

approximate measurements in the absence of full data and to keep costs low. In some cases, even when used for their intended purposes, such systems may still perform worse than more readily available and cheaper alternatives (e.g., using a computer mouse for system inputs [19]).

E. Laser-Based Localization

Similar to camera-based metrology solutions, laser-based systems may be used to localize the position of the robot by tracking fiducials located on or around the robot. Laser-based systems are generally more accurate than camera-based systems, but, whereas camera-based solutions can track multiple fiducials simultaneously, a given laser system is typically limited to tracking a solitary marker.

The use of extrinsic laser metrology systems (i.e., a laser tracking a fiducial mounted on the robot) is common, and is frequently used to track the tool flange of robot systems for basic robot performance evaluations (e.g., [20]). A single laser tracker will provide the 3D translation from the tracker’s base to the retroreflective target. For full 6DoF solutions, however, an active target must be affixed to the robot’s tool flange.

A different approach uses an intrinsic laser configuration in which the robot itself becomes the laser base. A laser emitter/detector is attached to the tool flange, and the robot servos either to maintain its bearing on a retroreflective target, or to find a target at some nominally defined location relative to the robot base. In cases where direct dynamic control of the robot is not possible, the latter of these two options would be used to register a robot to the world based on assumptions of the location of the retroreflective target plus an initial, rough approximation of the robot’s base. The robot searches for the target based on where it expects to find it, and moves in a regular pattern until the target is located. Such searches are ultimately subject to the resolution of the search parameters, the size of target, and the initial search location. Simple implementations may stop the robot’s search as soon as the laser detects the target (e.g., when performing a spiral search, Fig. 4), which is most likely to be on the target’s periphery.

V. REGISTRATION PERFORMANCE

As described in Section IV, there are a myriad of sources of measurement error. Depending on where the registration

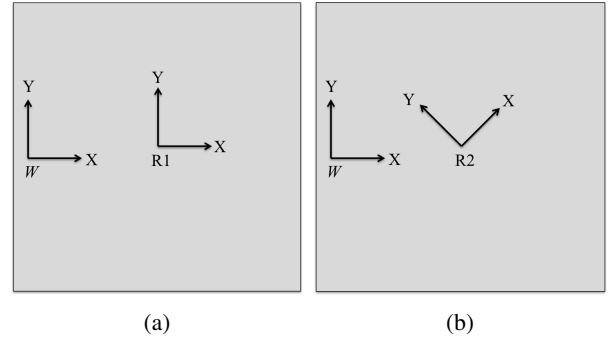


Fig. 5: The locations and orientations of robots R1 and R2 relative to the world center, W . R1 has a lateral shift on both the X and Y axes, but its orientation is aligned with W (a). R2 has the same lateral shifts, but is rotated 45° on the Z axis (b).

points are taken and the nature of the working environment, the errors may grow as a function of distance from the initial registration region. As such, it is important to understand how the selection of training points impacts registration performance. Here, the performance of the 3-point registration process using different pose measurement techniques on different robot platforms is evaluated and the results summarized.

A. Test methodology

The registration methodology was validated in a heterogeneous robot configuration featuring two different robots with different capabilities, kinematics, and controllers. The first robot, R1, was a 7DoF, collaborative open-chain manipulator with sub-millimeter repeatability. It was rigidly mounted in an upright configuration, with its base frame oriented inline relative to the world center (Fig. 5a). The second robot, R2, was a 6DoF, collaborative open-chain manipulator with sub-millimeter repeatability. It was also rigidly mounted, but had its base frame oriented at a 45° angle relative to the world origin (Fig. 5b). The motions of both robots were commanded via Ethernet from a central computer, but the robots and their controllers were kept in their stock configurations to handle kinematics and dynamics.

Performance testing started with randomly selecting 32 unique Cartesian poses expressed in the world coordinate system that were reachable by both robotic arms. Each of these poses were between 264 mm and 1223 mm (median 774 mm) from the world origin. These poses served as measurement sites for quantifying the error in TCP commands expressed in the world coordinate system (registration error) for either robot.

Three-point registration was performed for four different test cases in this paper. The first test case (“Short Hand”) used a hand-guiding methodology in which the robot’s TCP was physically moved to poses predefined in the world frame. These poses were defined by rigidly-mounted, 3D printed seats, distributed in an isosceles right triangle pattern with 150 mm leg lengths, with which the robots’ tools docked

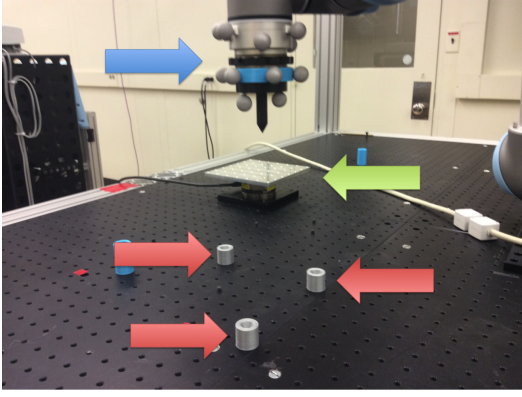


Fig. 6: Hand-guiding target seats (lower three red arrows) were rigidly mounted to the workspace, and had known translations to the world origin. A 6-axis load cell with an attached strike plate (center green arrow) was used for measuring the CoP, while tool-mounted retroreflective markers (top blue arrow) enabled a motion capture system to track the robot's TCP.

(Fig. 6). Hand-guiding the end effector enabled the robot to be coupled with the seats faster than if the robots had been jogged using the teach pendant. The second test case (“Long Hand”) used the same hand-guiding methodology as the first, but with a larger registration pattern (300 mm leg lengths). The third test case (“CoP”) used the CoP methodology described in Section IV-C. The fourth test case (“Average”) used an element-wise average of the prior three homogeneous transformation matrices obtained from their respective registration methods.

Both R1 and R2 were used to evaluate the performance of each test case. For each method, the positions of the robots’ TCPs were self-reported as described in Section IV-B, but were externally verified using an 8-camera motion capture system monitoring the $2\text{m} \times 2\text{m} \times 2\text{m}$ work volume. The motion capture system was calibrated such that the tracking of rigid objects (defined by a series of retroreflective targets mounted to the tool flange, Fig. 6) was accurate and repeatable to within 0.5 mm and 0.5° . The origin of the motion capture system was set to coincide with the origin of the world coordinate system.

B. Experimental Results

The results of the four previously mentioned test cases are presented in Table I. There are four key performance metrics for all test cases. These metrics capture the differences between the desired and actual poses of the robot in the world coordinate system, as measured by the reference system. At each of the 32 measurement sites, the L_2 norm was calculated between the desired and measured translations of the robot end-effector as expressed in the world coordinate system. Separately, the L_2 norm was calculated between the desired and measured rotations of the robot end-effector as expressed in the world coordinate system. Next, the mean and standard deviation were calculated across the 32 trials for the translations and rotations, separately.

TABLE I: Translation and orientation errors of the four test cases: Short hand-guiding pattern (“Short Hand”), long hand-guiding pattern (“Long Hand”), center-of-pressure (“CoP”), and averaging the registrations of Short Hand, Long Hand, and CoP (“Average”).

Test Case	Translation Error (Avg., Stdv.), mm	Orientation Error (Avg., Stdv.), deg
<i>ShortHand(R1)</i>	(8.824, 4.851)	(1.566, 0.797)
<i>ShortHand(R2)</i>	(4.639, 1.226)	(0.893, 0.585)
<i>LongHand(R1)</i>	(11.478, 6.242)	(1.684, 0.992)
<i>LongHand(R2)</i>	(2.711, 0.943)	(0.744, 0.443)
<i>CoP(R1)</i>	(9.771, 5.274)	(1.568, 0.878)
<i>CoP(R2)</i>	(3.011, 1.105)	(0.773, 0.448)
<i>Average(R1)</i>	(10.000, 4.822)	(1.515, 0.895)
<i>Average(R2)</i>	(3.198, 1.064)	(0.786, 0.492)

A statistical analysis was performed on the experimental data to determine any significant differences in the results. In particular, the data sets were passed through the two sample Kolmogorov-Smirnov test at a confidence level of $\alpha = 0.05$. At this confidence level, there existed no statistical difference between any of the test cases for R1. Specifically, the average performances for each of the evaluated registration methods for R1 are subsumed by the standard deviations of performance. However, there does exist a statistical difference between all of the test cases for R2, except when comparing the CoP and Average registrations. This result suggests that the Long Hand registration yielded the smallest registration errors (of statistical significance) followed by the CoP and Short Hand registration methods. These results imply that larger spreads of registration points yield more accurate and more precise performance across the work volume of the robot.

Another perspective of the data is presented in Fig. 7 and Fig. 8. In these figures, the robot TCP translation and rotation errors are plotted against the translation and rotation distances (L_2 norm) from the world coordinate system. This depiction of the data was used to unveil any trends in world-based positioning errors. When looking at Fig. 7, R1 experienced a positive correlation between translation error and translation distance for all test cases. In contrast, this positive trend does not appear to exist for R2. However, there does exist an increasing disparity between test cases for either robot with increasing translation distances from the world coordinate system. When looking at Fig. 8, trends do not appear as clearly. The only perceptible trend is that there is a slight positive correlation between rotation error and rotation distance for R1.

C. Alternative Verification Methodology

Recognizing that specialized tracking and metrology sensor systems are atypical for most SME environments, an alternative, low-cost methodology for registration verification was also evaluated. This methodology focuses on placing single collared seats throughout the work volume to test and correct minor translation errors. Moving under force control, the robot inserts the tool tip into the seats. The seat collars force translation and rotation adjustments of the TCP. The

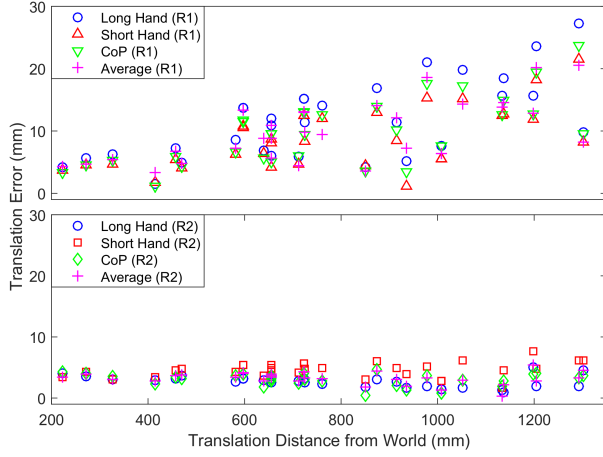


Fig. 7: Translation error as a function of the L_2 norm of translation from World.

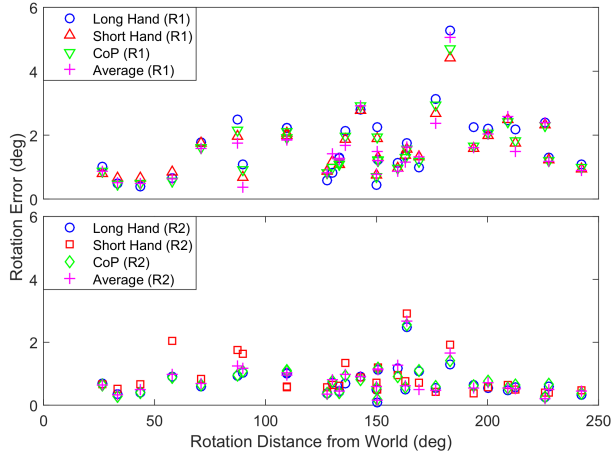


Fig. 8: Rotation error as a function of the L_2 norm of rotation from World.

nominal pose at the final insertion depth is known *a priori*, and provides a localized correction for the registration errors at that location. Six such seats were placed throughout the work volume, and the results of R1 and R2 docking with them using the Long Hand registration are reported in Table II and Table III, respectively.

The translation and rotation errors (“Trans. Error” and “Rot. Error,” respectively) are reported as the L_2 norm separating the self-reported, seated poses of the robots and the nominal poses of the seats. As was observed using the motion capture reference system, the distance from world (“World Dist.”) and the magnitudes of translation and rotation error are positively correlated. Moreover, the results of this self-reporting of errors are roughly equivalent with the results described in Section V-B. The difference between the self-reported and measured translations and rotations (“Trans. Delta.” and “Rot. Delta.” respectively) are proportional to

TABLE II: Reduced point-based verification results for R1 using the Long Hand registration based on a series of known poses throughout the work volume.

Nominal Seat Pose	World Dist. (mm)	Trans. Error (mm)	Rot. Error (deg)	Trans. Delta (mm)	Rot. Delta (deg)
(50, -400, 10, 180, 0, 0)	403.24	2.27	0.82	1.01	0.64
(575, -525, 10, 180, 0, 0)	778.68	4.56	0.86	1.48	0.65
(1175, -400, 10, 180, 0, 0)	1241.26	15.89	0.61	14.57	0.46
(1175, 425, 10, 180, 0, 0)	1249.54	24.11	0.39	17.46	0.93
(575, 550, 10, 180, 0, 0)	795.75	16.30	0.55	10.50	1.64
(50, 425, 10, 180, 0, 0)	428.05	9.09	0.87	5.25	1.27

TABLE III: Reduced point-based verification results for R2 using the Long Hand registration based on a series of known poses throughout the work volume.

Nominal Seat Pose	World Dist. (mm)	Trans. Error (mm)	Rot. Error (deg)	Trans. Delta (mm)	Rot. Delta (deg)
(50, -400, 10, 180, 0, 0)	403.24	0.57	0.25	0.24	0.25
(575, -525, 10, 180, 0, 0)	778.68	1.40	0.97	0.74	1.72
(1175, -400, 10, 180, 0, 0)	1241.26	3.13	0.48	1.28	1.03
(1175, 425, 10, 180, 0, 0)	1249.54	2.83	0.37	0.10	0.47
(575, 550, 10, 180, 0, 0)	795.75	1.40	0.90	0.89	0.08
(50, 425, 10, 180, 0, 0)	428.05	1.03	0.91	0.90	0.50

the registration errors as a function of distance. This method may thus be used to quickly and inexpensively self-report rough approximations of registration errors without requiring external measurement systems for verification.

This process has the added benefit of producing even more input data that may be subsequently and automatically used to improve overall registration performance. A new coordinate system origin may be placed at any of these verification sample locations when combined with the original registration data set. Registration performance may thus be improved by utilizing multiple associated coordinate systems, where the coordinate system origin is chosen as that which has the shortest distance to the target robot pose.

VI. DISCUSSION

In this paper, a simplified framework was presented that registers robots to any arbitrary coordinate systems. The performances of several registration mechanisms were compared through a systematic evaluation test method that measured translation and rotation error. In general, the performance of these different registration mechanisms was found to be largely comparable, though a clear indicator of registration performance is the spread magnitude of the sampling pattern P . Moreover, several sources of registration uncertainty were

identified, including measurement error, robot kinematic uncertainty, and manufacturing irregularities.

Furthermore, the performance metrics described in this paper provide a clear, general indication of position errors associated with spatial registration. Since robot system integration will likely require a specific positioning tolerance to yield successful applications, the described test method and performance metrics can be used to quantify post-registration position performance. The test method may also be used for benchmarking registration techniques.

A. Planned Improvements to the System

From the results discussed earlier, it is clear that the accumulation of errors in the system impairs the performance of a global registration as a function of translation and orientation magnitude. If one were to notice a trend of worsening performance as a function of distance, as was seen in Section V-B, one may resort to providing multiple local (e.g., quadrant-based) registrations to “re-center” the robot depending on where it is operating (e.g., the alternative verification seats used in Section V-C).

Likewise, it is believed that both the precision and accuracy can be improved by applying machine learning to automatically correct for registration errors. For example, statistical analysis (e.g., k-means clustering [21]) and data mining (e.g., bagging [22]) may be used to automatically generate multiple local registrations throughout the work volume. Alternatively, neural networks may be used to approximate and correct for the nonlinear shifts in registration performance throughout the work volume.

B. Ongoing Work

Based on the experience gained from this exercise, ongoing efforts are focused on further developing and simplifying the interface to improve the user experience. For example, based on the system inputs described in Section IV, a simplified interface with more functionality is being developed. Planned improvements include the ability to dynamically switch between registrations, and the development of a more robust mechanism for combining multiple registrations than the simple averaging described in V-B.

Similarly, an automatic registration correction mechanism is being developed that automatically “re-centers” the global registration for localized regions. This process moves the robot to *a priori* defined test poses, measuring the pose error, and creating homogeneous transformations that correct for the error.

Finally, a more comprehensive suite of registration evaluation test methods is being developed that uses both systematic and application-based performance metrics. This test suite will include specifications for inexpensive verification and validation mechanisms (e.g., on-arm sensing such as cameras for fine-tuning registration performance at a local level). In contrast, the methodology used during the initial registration evaluations was dependent on an expensive motion capture system, the calibration of which required an extensive front-loaded effort.

DISCLAIMER

Certain commercial equipment, instruments, or materials are identified in this paper to foster understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

REFERENCES

- [1] C. Fitzgerald, “Developing baxter: A new industrial robot with common sense for u.s. manufacturing,” *Proc. Technol. Pract. Rob. Applic.*, pp. 1–6, 2013.
- [2] H. Akeel and A. Watnabe, *Design of robot controllers*. John Wiley and Sons, Inc., 1999, ch. 12, pp. 213–244.
- [3] J. Marvel, E. Messina, B. Antonishek, L. Fronczek, and K. V. Wyk, “NISTIR 8093: Tools for collaborative robots within SME workcells,” National Institute of Standards and Technology, Tech. Rep., 2015.
- [4] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. Upper Saddle River, New Jersey, USA: Prentice Hall, 2003.
- [5] N. B. Figueroa, F. Schmidt, H. Ali, and N. Mavridis, “Joint origin identification of articulated robots with marker-based multi-camera optical tracking systems,” *Rob. Auton. Syst.*, vol. 61, no. 6, pp. 580–592, 2013.
- [6] K. Danilidis, “Hand-eye calibration using dual quaternions,” *Int. J. Rob. Res.*, vol. 18, pp. 286–298, 1998.
- [7] Y. Gan, X. Dai, and D. Li, “Off-line programming techniques for multirobot cooperation system,” *Int. J. Adv. Rob. Syst.*, vol. 10, 2013.
- [8] T. Tarn, A. Bejczy, and X. Yun, “Design of dynamic control of two cooperating robot arms: Closed chain formulation,” *Proc. IEEE Int. Conf. Rob. Autom.*, pp. 7–13, 1987.
- [9] R. Rocha, J. Dias, and A. Carvalho, “Cooperative multi-robot systems: A study of vision-based 3-d mapping using information theory,” *Rob. Auton. Syst.*, vol. 53, pp. 282–311, 2005.
- [10] S. Carpin, “Fast and accurate map merging for multi-robot systems,” *Auton. Rob.*, vol. 25, pp. 305–316, 2008.
- [11] P. Koch, S. May, M. Schmidpeter, M. Kuehn, and C. Pfizner, “Multi-robot localization and mapping based on signed distance functions,” *Proc. IEEE Int. Conf. Auton. Rob. Syst. Compet.*, pp. 77–82, 2015.
- [12] M. Fiala, “Artag, a fiducial marker system using digital techniques,” *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit.*
- [13] C. English, G. Okouneva, and A. Choudhuri, “Shape-based pose estimation evaluation using expectancy index artifacts,” *Proc. Perf. Metrics Intell. Syst. Work.*, pp. 64–68, 2012.
- [14] A. Broun, C. Beck, T. Pipe, M. Mirmehdi, and C. Melhuish, “Building a kinematic model of a robot’s arm with a depth camera,” *Adv. Auton. Rob.*, pp. 105–116, 2012.
- [15] M. Klingensmith, T. Galluzzo, C. Dellin, J. Bagnell, and N. Pollard, “Closed-loop servoing using real-time markerless arm tracking,” *Proc. IEEE Int. Conf. Rob. Autom. Hum. Works.*, 2013.
- [16] F. Flacco, T. Kroger, A. Luca, and O. Khatib, “A depth space approach to human-robot collision avoidance,” *Proc. IEEE Int. Conf. Rob. Autom.*, pp. 338–345, 2012.
- [17] P. Hebert, N. Hudson, J. Ma, T. Howard, T. Fuchs, M. Bajracharya, and J. Burdick, “Combined shape, appearance and silhouette for simultaneous manipulator and object tracking,” *Proc. IEEE Int. Conf. Rob. Autom.*, pp. 2405–2412, 2012.
- [18] J. A. Marvel, M. Franaszek, J. L. Wilson, and T. Hong, “Performance evaluation of consumer-grade 3d sensors for static 6dof pose estimation systems,” *Proc. SPIE 8499, Appl. Digital Image Process. XXXV*, pp. 849 905 1–11, 2012.
- [19] A. Pino, E. Tzemis, N. Ioannou, and G. Kouroupetroglou, “Using kinect for 2d and 3d pointing tasks: Performance evaluation,” *Human-Computer Interaction. Interaction Modalities and Techniques*, pp. 358–367, 2013.
- [20] T. Chang, T. Hong, M. Shneider, G. Holguin, J. Park, and R. Eastman, “Dynamic 6dof metrology for evaluating a visual servoing system,” *Proc. Perf. Metrics. Intel. Syst. Work.*, pp. 173–180, 2008.
- [21] J. A. Hartigan, *Clustering Algorithms*. John Wiley and Sons, Inc, 1975.
- [22] L. Breiman, “Bagging predictors,” *Mach. Learn.*, vol. 24, pp. 123–140, 1996.