



Faculty Publications

1997-11-01

Robotic TCF and rigid-body calibration methods

Edward Red

Xuguang Wang

Follow this and additional works at: <https://scholarsarchive.byu.edu/facpub>



Part of the [Mechanical Engineering Commons](#)

Original Publication Citation

Robotica 15.6(Nov 1997): 633-644.

BYU ScholarsArchive Citation

Red, Edward and Wang, Xuguang, "Robotic TCF and rigid-body calibration methods" (1997). *Faculty Publications*. 657.

<https://scholarsarchive.byu.edu/facpub/657>

This Peer-Reviewed Article is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Faculty Publications by an authorized administrator of BYU ScholarsArchive. For more information, please contact ellen_amatangelo@byu.edu.

Robotic TCF and rigid-body calibration methods

*Xuguang Wang and †Edward Red

(Received in Final Form: January 8, 1997)

SUMMARY

For off-line programming to work, systematic methods must be developed to account for non-ideal performance of the parts and devices in the manufacturing cell. Although much of the literature focuses on robot inaccuracy, this paper considers practical methods for the tool control frame (TCF) calibration and rigid-body compensation required to close the inverse kinematics loop for target driven tasks.

In contrast to contemporary estimation methods, a closed-form, easily automated, solution is introduced for calibrating the position and orientation (pose) of orthogonal end-effectors when the distal robot joint is revolute. This paper also considers methods for measuring and compensating the small rigid-body perturbations that result from non-repeatable part delivery systems or from geometric distortion. These methods are designed to eliminate $r\theta$ error from the rigid-body prediction and can be conducted in real-time. Without accurate TCF calibration and rigid-body compensation, even the most accurate robot will fail to complete an off-line programmed task if the task tolerances are stringent.

KEYWORDS: Tool and sensor calibration; Off-line programming; Rigid-body compensation

INTRODUCTION

Off-line programming uses computers to program physical mechanisms such as robots without direct access to the equipment when the programming is performed. Obviously, on-line programming limitations, such as the visual limitations faced by operators using a teach pendant, are potentially eliminated by off-line methods. While on-line programming depends on robot repeatability, robot accuracy is extremely important for off-line programming, where the coordinates that define a target pose (position and orientation) are obtained from a database.

Unfortunately, compensating robot inaccuracy is only a necessary condition for successful off-line programming. Several additional issues must be addressed for off-line programming to be practical, namely the automation of

- TCF calibration procedures for orthogonal end-effectors. This paper introduces a closed-form solution that differs from contemporary estimation methods which estimate the TCF pose, command some motion

relative to a part feature, then measure the resulting error, which is then used to provide a better estimate. Once the error approaches the robot repeatability, the process is terminated.

- rigid-body correction of rigid-body deviations with multiple measurement devices. Rigid-body deviations are normal because of variability in part delivery systems, and variability in the manufacture of the parts themselves.
- calibration control methods which use a real-time database to automate the calibration sequence based on application features which have been defined in an off-line programmed task. These procedures are briefly considered in this paper. Refer to Wang¹ for more detail.

This research has been motivated by the difficulty encountered in trying to make off-line methods actually work. These methods require an understanding of the significant research in robot calibration, usually directed towards positioning accuracy improvement. Global calibrations methods are considered by Hollerbach,² Ziegart,³ and Roth,^{4,5} whereas Davies⁶ and Red^{7,8} present local calibration techniques.

Global calibration, though concerned with modifying the robot kinematic model to account for errors inherent to the robot, does not easily lend itself to an automated approach to off-line programming for a number of reasons:

- a robot's corrected kinematic model does not easily adapt to time-variable error sources such as compliance under variable loads, or thermal distortion;
- global calibration does not focus on integrating the other critical elements of the manufacturing process, such as the tools and parts;
- global calibration procedures are time consuming and may require the cell to be taken off-line.

In contrast, local calibration is concerned with measuring and modifying robot performance relative to parts by modifying target poses to account for relative pose (position and orientation) error. Local methods are less concerned with global accuracy, but instead fit statistical models to the robot error measured relative to parts or part templates by sensors such as vision systems.

By lumping numerous error sources into the local models, local calibration can adapt to changes in part geometry or in the environment, and can be applied on-line within minutes. Local methods are particularly compatible with the tool calibration methods introduced in this paper, once the methods are integrated with a workcell model database, which can be updated in real-time. Local methods are more suitable for tasks

* Cimatrix, Inc. (U.S.A.)

† Dept. of Manufacturing Engineering and Engineering Technology, Brigham Young Univ., Provo, UT 84602 (USA).

conducted in defined and smaller work volumes, such as robotic assembly.

Substantial research in robot calibration is directed towards improving autonomous robot performance and accuracy, rather than an integrated approach to cell calibration. In an industrial cell, robot performance is typically more stable than the dynamic tool and sensor environment. And even the parts being operated on may experience geometric deviations due to subtle changes in upstream manufacturing processes. Thus, a well calibrated robot may not be sufficient to make off-line programming practical.

A dominant research theme for TCF calibration is to solve the $\mathbf{AX} = \mathbf{XB}$ equation of homogeneous transformations. \mathbf{A} represents the relative change of the last joint frame and \mathbf{B} the relative change of the tool or sensor frame as the robot moves between two configurations. \mathbf{X} represents the unknown tool/sensor homogeneous transformation which is to be determined relative to the last joint frame.

Park⁹ solves the $\mathbf{AX} = \mathbf{XB}$ equation using Lie theory, demonstrating that the method works with experimental data from a previously published paper. The authors fail to consider the difficult problems of measuring the homogeneous transformations.

Although Preising¹⁰ presents a unique inverse measurement approach, the general measurement complexity for solving the $\mathbf{AX} = \mathbf{XB}$ equation is apparent in their methods. A backlit calibration plate is attached to the robot distal frame, and a vision camera is fixed in space (generally, the calibration plate is fixed and the camera is attached to the robot). The robot moves the calibration plate having multiple features within the camera's field of view to generate the multiple configurations required to solve the $\mathbf{AX} = \mathbf{XB}$ equation. Although using Shiu's¹¹ algorithm successfully to solve for \mathbf{X} , the author's note that their methods may experience application difficulties in a real industrial cell.

It is not surprising to see new approaches arise that circumvent the TCF calibration problem. For example, in the paper by Yoshimi¹² a camera is attached to the end of the robot to view the gripper motion relative to some known feature on a part. As the robot rotates the camera about the gripper's rotational axis, the camera views the path of the gripper relative to some feature. The relative shape of the path is used to change the relative pose of the gripper relative to the feature. For example, an elliptical path indicates non-orthogonality. Pose modification continues until the gripper to feature path becomes a point, at which time the gripper has achieved the target pose.

The purpose of this paper, then, is to demonstrate a practical on-line calibration method that can be integrated as a real-time process. It is limited to orthogonal end-effectors and sensors, which are common to many robotic applications, particularly robotic assembly.

The following sections derive the closed form equations for locating a tool frame with respect to a

measurement sensor, either attached to or external to the robot. Rigid-body compensation procedures are also considered for parts which experience small deviations in the range up to 5°. Finally, application results are presented which demonstrate the real-time efficiency of these methods when integrated with local calibration methods.

TCF CALIBRATION

Sensors such as vision cameras and probes, and end-effectors such as vacuum tools and welding torches are synonymously referred to as tools, sensors, or end-effectors in this paper. In data-driven tasks tools and sensors have attached XYZ frames represented mathematically by homogeneous transformations—we call these operational frames the tool/terminal control frame or simply TCF.

The command *move tool to target with 10 mm Z offset* requires an accurate pose for the TCF and an accurate pose for the target frame. This section introduces an easily automated and closed-form method for determining the TCF pose used in the kinematic loop closure equations. Having a closed-form method reduces the number of iterations required to get a good estimate of the TCF pose, in light of other statistical variations that naturally occur in the measurement process (such as robot repeatability).

The methods herein assume that the distal (terminal) robot joint is revolute and that the robot can orient the joint Z axis normal to the part surface. This is not a limiting assumption since most 6-axis, cylindrical, and Scara robots have a distal joint of this type. Calibration will typically determine the TCF relative to the last joint frame. Orthogonal end-effectors are oriented so that either the TCF X, Y, or Z axis is parallel to the joint Z axis.

Off-line task planning for more sophisticated tasks, such as the welding, may require TCF calibration for a non-orthogonal end-effector such as the tip of a welding gun, but these methods are not considered here. See Wang¹ for a discussion of these methods, including a discussion of TCF calibration for non-axisymmetric tools.

For any particular robot configuration typical robot controllers will configure the robot terminal frame $(XYZ)_e$ in robot base coordinates (Cartesian space) or in the form of robot joint values (joint space) – see Figure 1.

A given TCF configuration can be represented in either Cartesian space or joint variable space, and mapped from one space to the other using forward kinematics or inverse kinematics. However, the robot controller does not by default know where the attached tool or measurement device is located. The function of TCF calibration is to determine the mathematical transformation which relates an end-effector TCF frame $(XYZ)_T$ to the parent joint frame $(XYZ)_e$. Given such information, the robot can be commanded to move a tool or sensor to poses that are either obtained from a database and which may be continually updated by a sensor measurement.

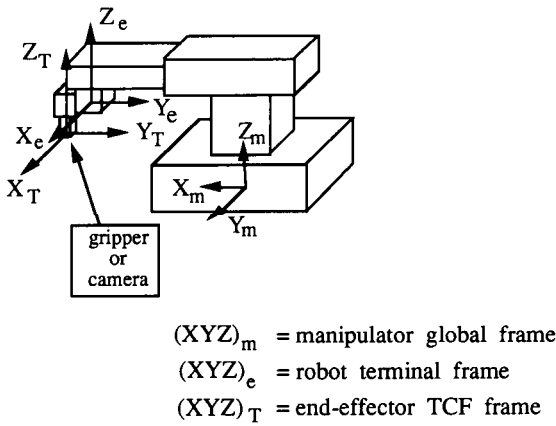


Fig. 1. Manipulator frames.

Automated TCF calibration methods require robot relative movement in localized regions and statistical methods for TCF calibration. The end-effector is treated as a rigid-body; therefore, the displacement of the tool from the parent joint frame is constant, which can be determined from the relative robot movement or the differences between corresponding robot configurations.

To explain the necessary transformations and transformation equations used for automatic TCF calibration, the following convention regarding the homogeneous transformation between two space frames will be used throughout this paper. A bold face letter **T** with superscript *a* and subscript *b*, or T_b^a , represents the homogeneous transformation that locates frame $(XYZ)_a$ relative to frame $(XYZ)_b$. This 4×4 matrix can be expressed as a 3×3 rotational submatrix and a position vector as shown in equation (1).

$$T_b^a = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_b^a & \mathbf{p}_b^a \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (1)$$

where:

$$\text{rotational submatrix: } \mathbf{R}_b^a = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2)$$

$$\text{position vector: } \mathbf{p}_b^a = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \quad (3)$$

$$\text{zero vector transposed: } \mathbf{0}^T = [0 \ 0 \ 0] \quad (4)$$

a. TCF Calibration for a vision camera

Although described for a camera, the TCF calibration methods presented in this section are applicable for tools or sensors which have a frame attached to them and can have their motion measured with respect to part geometry or sensor reference geometry. When a vision camera is attached to a robot link and used to monitor part pose deviations, its coordinate frame needs to be

calibrated relative to the link frame to which it is attached, typically the robot distal (last) joint frame.

In this section we assume that methods to calibrate the internal parameters of a vision camera such as pixel ratio (pixel height to pixel width) and pixel size (pixels per standard linear measure) can be applied to calibrate the camera. The camera is mounted so that the TCF Z axis is parallel to the joint Z axis. The TCF is not located on the camera, but at the object plane of the lens.

The aperture and focal length on the camera are set according to need. It may be necessary to set a larger focal length because of obstructions in the robot's path. A large focal length also gives a larger viewing area on the display monitor. A small focal length allows a close-up view of a small object for measurements requiring a finer tolerance. The aperture is set to allow the desired amount of light into the camera. The smaller the aperture, the less light the camera will see. Once the focal length and aperture have been set, they will typically not be changed.

The automatic TCF calibration procedures presented here only require a **single vision sensitive dot** which will hereafter be referred to simply as a dot or calibration dot. The calibration dot provides a reference point in the robot workspace for relative robot end-effector movement in localized regions. The relative motion of the robot is then used to calibrate the relative rigid displacement between the camera coordinate frame and the robot link frame to which the sensor is attached.

The following steps are required to complete the TCF calibration operation. Note that these steps can be easily automated since the calibration feature is a simple dot.

1. Place a calibration plate having a calibration dot in the robot workspace.
2. Adjust the camera such that the camera is near normal to the calibration surface (one time process). Camera orthogonalization routines may be necessary for high tolerance applications.
3. Move the robot arm to a position such that the calibration dot is near the center of the vision window. Call this configuration 1 in Figure 2.
4. Activate the vision system to take a picture of the calibration dot, process the image, then record the dot coordinate (X_{v1}, Y_{v1}) in the vision coordinate frame $(XYZ)_v$ at configuration 1.
5. Record the robot configuration either as a set of joint angles or as a homogeneous transformation relating

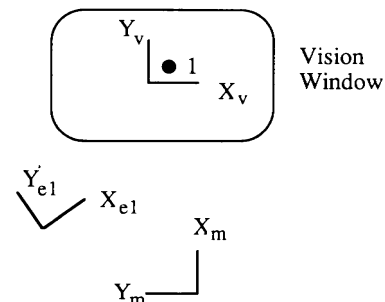


Fig. 2. Camera TCF calibration configuration 1.

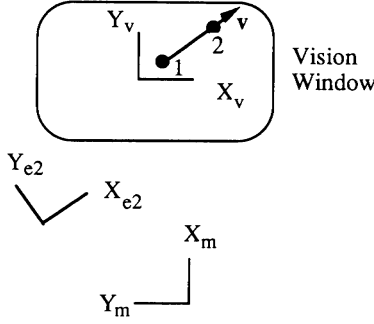


Fig. 3. Camera TCF calibration configuration 2.

the last joint frame $(XYZ)_e$ relative to base frame $(XYZ)_m$.

6. Move the robot in the X -axis direction of its last joint frame. While moving the robot arm, make sure the dot is still inside the vision window. The dot is now in vision screen position 2. Call this configuration 2, Figure 3.
7. Activate the vision system to take and process the image, and record the dot coordinate (X_{v2}, Y_{v2}) in the vision coordinate frame $(XYZ)_v$ at configuration 2.
8. Move the robot arm such that its last joint frame is at a different orientation from that of configuration 1. Make sure the calibration dot is inside the vision window. The dot is now at vision screen configuration 3, Figure 4. Initial calibration requires the robot to move relative to its last joint frame a small amount to guarantee that the dot is inside the vision window. Later calibration can move the robot relative to the initially calibrated vision camera frame, and guarantee that the dot is almost at the center of the vision window.
9. Activate the vision system to take and process the image, and record the dot coordinate (X_{v3}, Y_{v3}) in the vision coordinate frame $(XYZ)_v$ at configuration 3.
10. Record current robot configuration in the form of a set of joint angles or a homogeneous transformation relating the last joint frame $(XYZ)_e$ relative to the robot base frame $(XYZ)_m$.
11. Carry out TCF calibration calculation using data collected in steps 4 to 10.

From Figure 2 the relative transformation of the calibration dot with respect to the robot base frame can be found, equation (5). The calibration dot can only define the origin of a dot coordinate frame whose

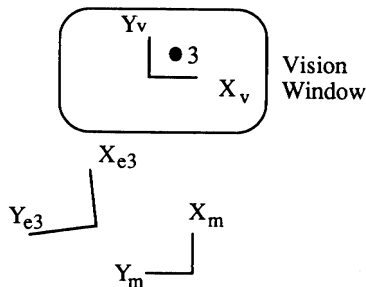


Fig. 4. Camera TCF calibration configuration 3.

orientation is arbitrary.

$$\mathbf{T}_m^{dot} = \mathbf{T}_m^1 = {}^1\mathbf{T}_m^e \mathbf{T}_e^v \mathbf{T}_v^1 \quad (5)$$

Representing the transformations on the right side of (5) in the form of a rotational submatrix and a position vector gives

$$\begin{aligned} \mathbf{T}_m^{dot} &= \begin{bmatrix} {}^1\mathbf{R}_m^e & {}^1\mathbf{p}_m^e \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_e^v & \mathbf{p}_e^v \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_v^1 & \mathbf{p}_v^1 \\ \mathbf{0} & 1 \end{bmatrix} \\ &= \begin{bmatrix} {}^1\mathbf{R}_m^e \mathbf{R}_e^v & {}^1\mathbf{R}_m^e \mathbf{p}_e^v + \mathbf{p}_m^e \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_v^1 & \mathbf{p}_v^1 \\ \mathbf{0} & 1 \end{bmatrix} \\ &= \begin{bmatrix} {}^1\mathbf{R}_m^e \mathbf{R}_e^v \mathbf{R}_v^1 & {}^1\mathbf{R}_m^e \mathbf{R}_e^v \mathbf{p}_v^1 + {}^1\mathbf{R}_m^e \mathbf{p}_e^v + \mathbf{p}_m^e \\ \mathbf{0} & 1 \end{bmatrix} \quad (6) \end{aligned}$$

Similarly, from Figure 4 the relative transformation of the calibration dot with respect to the robot base frame can be found, equation (7).

$$\begin{aligned} \mathbf{T}_m^{dot} &= \mathbf{T}_m^3 = {}^3\mathbf{T}_m^e \mathbf{T}_e^v \mathbf{T}_v^3 \\ &= \begin{bmatrix} {}^3\mathbf{R}_m^e & {}^3\mathbf{p}_m^e \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_e^v & \mathbf{p}_e^v \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_v^3 & \mathbf{p}_v^3 \\ \mathbf{0} & 1 \end{bmatrix} \\ &= \begin{bmatrix} {}^3\mathbf{R}_m^e \mathbf{R}_e^v & {}^3\mathbf{R}_m^e \mathbf{p}_e^v + {}^3\mathbf{p}_m^e \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_v^3 & \mathbf{p}_v^3 \\ \mathbf{0} & 1 \end{bmatrix} \\ &= \begin{bmatrix} {}^3\mathbf{R}_m^e \mathbf{R}_e^v \mathbf{R}_v^3 & {}^3\mathbf{R}_m^e \mathbf{R}_e^v \mathbf{p}_v^3 + {}^3\mathbf{R}_m^e \mathbf{p}_e^v + {}^3\mathbf{p}_m^e \\ \mathbf{0} & 1 \end{bmatrix} \quad (7) \end{aligned}$$

Notice that even though the calibration dot appears in different positions 1, 2, and 3 in the vision window, it is fixed relative to the robot base frame $(XYZ)_m$ during the calibration process. The position components of equations (6) and (7) are, therefore, equal.

$${}^1\mathbf{R}_m^e \mathbf{R}_e^v \mathbf{p}_v^1 + {}^1\mathbf{R}_m^e \mathbf{p}_e^v + \mathbf{p}_m^e = {}^3\mathbf{R}_m^e \mathbf{R}_e^v \mathbf{p}_v^3 + {}^3\mathbf{R}_m^e \mathbf{p}_e^v + {}^3\mathbf{p}_m^e \quad (8)$$

Rearranging (8),

$$({}^1\mathbf{R}_m^e - {}^3\mathbf{R}_m^e) \mathbf{p}_e^v = {}^3\mathbf{R}_m^e \mathbf{R}_e^v \mathbf{p}_v^3 - {}^1\mathbf{R}_m^e \mathbf{R}_e^v \mathbf{p}_v^1 + {}^3\mathbf{p}_m^e - {}^1\mathbf{p}_m^e \quad (9)$$

Since the robot is moved in its last joint X -axis direction in automatic TCF calibration step 6, vector \mathbf{v} formed by calibration dot images 1 and 2 on the vision window has the same orientation as that of the X -axis direction of the robot last joint frame $(XYZ)_e$. Therefore, the relative orientation between the vision coordinate frame $(XYZ)_v$ and the robot last joint frame $(XYZ)_e$ is readily available from equations (10) and (11):

$$\mathbf{R}_e^v = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (10)$$

where

$$\theta = \tan^{-1} \left(\frac{Y_{v2} - Y_{v1}}{X_{v2} - X_{v1}} \right) \quad (11)$$

Knowing \mathbf{R}_e^v , the orientation components of the vision camera TCF, equation (9) can then be solved for the position components of the TCF, namely, \mathbf{p}_e^v . Equation (9) suggests that to have a non-singular solution, ${}^1\mathbf{R}_m^e$ and ${}^3\mathbf{R}_m^e$ must be different. In other words, the robot last joint frame must have different orientations for configurations 1 and 3 in calibration steps 3 and 8, which can be implemented automatically.

Due to robot inaccuracy, vision system resolution and numerical roundoff, calibration steps 3 to 11 can be repeated a number of times to establish statistical certainty. The statistical average can then be used to

predict more accurate TCF values and the standard deviations used to quantitatively verify the TCF calibration accuracy.

Note that estimation methods use a convergence process and thus are not likely to improve the TCF location beyond the repeatability of the robot and other process variables. In contrast, by not depending on convergence to determine the TCF coordinates, exact methods may not be limited by the robot repeatability in locating the TCF coordinates and thus TCF location accuracy will improve on the estimation methods.

b. TCF Calibration for an axisymmetric tool

Unlike a vision camera, an axisymmetric tool (e.g., a vacuum tool) requires an external reference frame which can measure relative changes in pose of robot end-effectors or devices attached to the end-effectors. Typical measurement devices include a digitizing tablet, a vision camera, or a laser triangulation system.

The automated TCF calibration procedures presented in this research employ a digitizing tablet and a non-contact stylus (probe). Even though a digitizing tablet is used to present the automatic TCF calibration methods, this method itself is general and has been demonstrated successfully using a fixed vision camera as the measurement device, which may be the easier method to automate, if cost is not an issue.

A digitizing tablet is an electronic device that transmits the relative position of a stylus to software running on a host computer. Modern digitizing devices have resolution of 0.001 inches and an accuracy of 0.01 inches or better. A stylus probe is also an electronic device which is used to activate digitizing tablet components for transmitting its coordinates on the tablet. A typical stylus probe is composed of an excitable LC circuit. The coordinate system of the digitizing tablet will be used as an intermediate reference calibration frame. Unique relative robot movements establish close loop transformation equations which are then solved for the tool TCF origin.

A physical interface is used to attach and align the probe on the tool being calibrated. If high calibration accuracy is required, the interface must be built to the appropriate tolerances. The Z axis of the probe should be parallel to the tool TCF Z axis, which is the axis of symmetry. TCF calibration of an axisymmetric tool only requires the TCF origin relative to the joint to which it is attached.

Together, the probe and the digitizing tablet determine the X and Y probe positions. The interface should be capable of positioning the probe frame origin in line with the TCF Z axis within the desired tolerances. The interface should be designed to place the probe frame as close to the TCF as possible. Figure 5 shows an example of an interface for an axisymmetric touch probe with the probe attached to it. Set screws may be required to prevent sliding of the interface along the Z axis. This would not be necessary if we are calibrating a vacuum tool, since proper design of the interface would use the vacuum to suck up the probe.

The centering pin of the probe unit is designed to

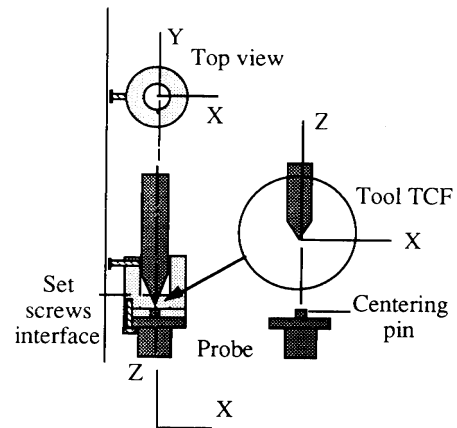


Fig. 5. Probe attached to axisymmetric tool.

align the probe electronic center (center of the stylus coil) with the tool axis of symmetry. For improved TCF calibration accuracy, the interface could permit the probe to be rotated 180° from its original position, and calibrated for a second set of TCF values. Averaging the two TCF values minimizes misalignment errors.

Since the TCF of an axisymmetric tool does not have a unique orientation, it will be convenient to assume that the TCF is parallel to the robot last joint frame $(XYZ)_e$. The detailed calibration procedures then follow:

1. Attach the axisymmetric tool to be calibrated to the robot distal link either manually or through a tool changer.
2. Attach the stylus interface unit to the axisymmetric tool.
3. Place the digitizing tablet in an appropriate location in the robot workspace.
4. Adjust the digitizing tablet or the robot such that the stylus Z axis is near normal to the digitizing tablet.
5. Move the robot such that the stylus is near the center of the digitizing tablet, designated configuration 1, Figure 6.
6. Record the stylus position (X_1, Y_1) in the digitizing tablet frame $(XYZ)_D$ at configuration 1.
7. Record the current robot configuration in the form of a set of joint angles or a homogeneous transformation relating the last joint frame $(XYZ)_e$ to the robot base frame $(XYZ)_m$.
8. Move the robot in the X-axis direction of its last joint frame. While moving the robot arm, make sure

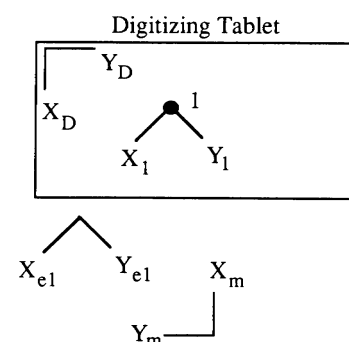


Fig. 6. Axisymmetric tool TCF calibration configuration 1.

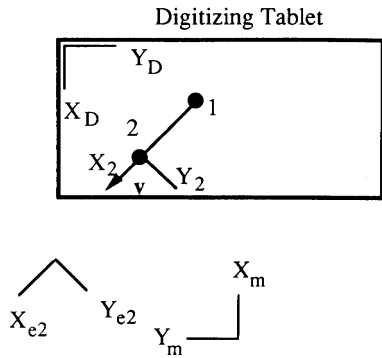


Fig. 7. Axisymmetric tool TCF calibration configuration 2.

the stylus is within the active area of the digitizing tablet. The stylus is now at configuration 2 relative to the digitizing tablet, Figure 7.

9. Record the stylus position (X_2, Y_2) in the digitizing tablet frame $(XYZ)_D$ at configuration 2.
10. Move the robot arm such that its last joint frame is at a different orientation from that of configuration 1. Make sure the stylus is within the active area of the digitizing tablet. The stylus is now on the digitizing tablet at configuration 3, Figure 8. Initial calibration requires the robot to move relative to its last joint frame a small amount to guarantee that the stylus is within the active area of the digitizing tablet. After an initial estimate, later calibration can move robot relative to the initially calibrated axisymmetric tool frame, and guarantee that the stylus remains close to the center of the digitizing tablet.
11. Record the stylus position (X_3, Y_3) in the digitizing tablet frame $(XYZ)_D$ at configuration 3.
12. Record the current robot configuration in the form of a set of joint angles or a homogeneous transformation relating the last joint frame $(XYZ)_e$ to the robot base frame $(XYZ)_m$.
13. Carry out TCF calibration calculation using data collected in steps 6, 7, 9, 11, and 12.

Notice that during the calibration processes the digitizing tablet reference frame is fixed with respect to the robot base frame $(XYZ)_m$. This allows us to merge configurations in Figures 6 to 8 into one configuration by matching frames $(XYZ)_D$ and $(XYZ)_m$, Figure 9.

To form the closed loop transformation equations for TCF calibration, an artificial frame $(XYZ)_{3'}$ with the

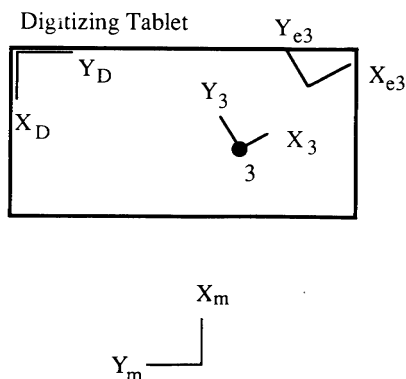


Fig. 8. Axisymmetric tool TCF calibration configuration 3.

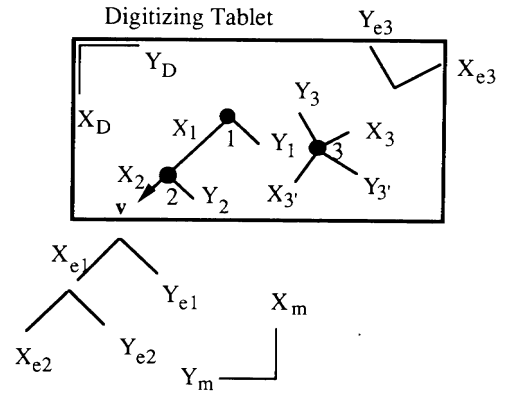


Fig. 9. Axisymmetric tool TCF calibration merging configuration.

origin of frame $(XYZ)_3$ and orientation of frame $(XYZ)_1$ is created. The relative transformation between frame $(XYZ)_{3'}$ and the robot base frame $(XYZ)_m$ can be represented by either (12) or (13):

$${}^1\mathbf{T}_m^{3'} = {}^1\mathbf{T}_m^e {}^e\mathbf{T}_1^{3'} \quad (12)$$

$${}^1\mathbf{T}_m^{3'} = {}^3\mathbf{T}_m^e {}^e\mathbf{T}_3^{3'} \quad (13)$$

From equations (12) and (13), we get

$${}^1\mathbf{T}_m^e {}^e\mathbf{T}_1^{3'} = {}^3\mathbf{T}_m^e {}^e\mathbf{T}_3^{3'} \quad (14)$$

The transformation from the axisymmetric tool $(XYZ)_1$ or $(XYZ)_3$ to the robot last joint frame $(XYZ)_e$ is a constant transformation because both frames are attached to the same rigid-body. As a matter of fact, this is the TCF transformation to be calibrated.

$$\mathbf{T}_e^1 = \mathbf{T}_e^3 = \mathbf{T}_e^{cf} = \begin{bmatrix} \mathbf{R}_e^{cf} & \mathbf{p}_e^{cf} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{p}_e^{cf} \\ \mathbf{0} & 1 \end{bmatrix} \quad (15)$$

Combining equations (14) and (15), and representing the transformations equation (14) in the form of rotation submatrices and position vectors gives

$$\begin{bmatrix} {}^1\mathbf{R}_m^e & {}^1\mathbf{p}_m^e \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{p}_e^{cf} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{p}_1^{3'} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} {}^3\mathbf{R}_m^e & {}^3\mathbf{p}_m^e \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{p}_e^{cf} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_3^{3'} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \quad (16)$$

Multiplying the transformation matrix on both sides of the equation (16) gives

$$\begin{bmatrix} {}^1\mathbf{R}_m^e & {}^1\mathbf{R}_m^e \mathbf{p}_1^{3'} + {}^1\mathbf{R}_m^e \mathbf{p}_e^{cf} + {}^1\mathbf{p}_m^e \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} {}^3\mathbf{R}_m^e \mathbf{R}_3^{3'} & {}^3\mathbf{R}_m^e \mathbf{p}_e^{cf} + {}^3\mathbf{p}_m^e \\ \mathbf{0} & 1 \end{bmatrix} \quad (17)$$

For the position components, we get

$${}^1\mathbf{R}_m^e \mathbf{p}_1^{3'} + {}^1\mathbf{R}_m^e \mathbf{p}_e^{cf} + {}^1\mathbf{p}_m^e = {}^3\mathbf{R}_m^e \mathbf{p}_e^{cf} + {}^3\mathbf{p}_m^e \quad (18)$$

Rearranging equation (18) gives the TCF calibration equation

$$({}^1\mathbf{R}_m^e - {}^3\mathbf{R}_m^e) \mathbf{p}_e^{cf} = {}^3\mathbf{p}_m^e - {}^1\mathbf{p}_m^e - {}^1\mathbf{R}_m^e \mathbf{p}_1^{3'} \quad (19)$$

Notice that frames $3'$ and 1 are parallel. Since the origins of the two frame are known with respect to the

digitizing tablet coordinate frame $(XYZ)_D$, the stylus position $3'$ relative to the coordinate frame 1 can be found by the following equations;

$$\mathbf{T}_1^{3'} = \begin{bmatrix} \mathbf{R}_1^{3'} & \mathbf{p}_1^{3'} \\ \mathbf{0} & 1 \end{bmatrix} = \mathbf{T}_1^D \mathbf{T}_D^{3'} = [\mathbf{T}_D^1]^{-1} \mathbf{T}_D^{3'} \quad (20)$$

Carrying out the 4×4 homogeneous matrices on the right side of the equation, we get

$$\begin{aligned} \mathbf{T}_1^{3'} &= \begin{bmatrix} \cos \theta & -\sin \theta & 0 & X_1 \\ \sin \theta & \cos \theta & 0 & Y_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \\ &\times \begin{bmatrix} \cos \theta & -\sin \theta & 0 & X_3 \\ \sin \theta & \cos \theta & 0 & Y_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos \theta & \sin \theta & 0 & -X_1 \cos \theta - Y_1 \sin \theta \\ -\sin \theta & \cos \theta & 0 & X_1 \sin \theta - Y_1 \cos \theta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &\times \begin{bmatrix} \cos \theta & -\sin \theta & 0 & X_3 \\ \sin \theta & \cos \theta & 0 & Y_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 & (X_3 - X_1) \cos \theta + (Y_3 - Y_1) \sin \theta \\ 0 & 1 & 0 & -(X_3 - X_1) \sin \theta + (Y_3 - Y_1) \cos \theta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (21) \end{aligned}$$

and therefore,

$$\mathbf{p}_1^{3'} = \begin{bmatrix} (X_3 - X_1) \cos \theta + (Y_3 - Y_1) \sin \theta \\ -(X_3 - X_1) \sin \theta + (Y_3 - Y_1) \cos \theta \\ 0 \end{bmatrix} \quad (22)$$

where

$$\theta = \tan^{-1} \left(\frac{Y_2 - Y_1}{X_2 - X_1} \right) \quad (23)$$

To have a non-singular solution, equation (19) suggests that ${}^1\mathbf{R}_m^e$ and ${}^3\mathbf{R}_m^e$ must be different. In other words, the robot last joint frame must assume different orientations for configurations 1 and 3 in calibration steps 5 and 10. Due to robot inaccuracy, digitizing tablet resolution and numerical roundoff, calibration steps 5 to 13 can be repeated a number of times to establish a statistical average.

RIGID-BODY CORRECTION

Rigid-body correction refers to the updating of an ideal workcell geometry database using part rigid-body deviations measured in the workcell. The accuracy of the extracted rigid-body pose depends on the sensor resolution and the measurement noise, particularly

orientation noise. Positional error due to orientation noise is proportional to the distance r between the target frame and the rigid-body updating frame. This effect is hereafter referred to as $r\theta$ error (often referred to as Abbe error).

Because of $r\theta$ error, the orientation information obtained directly by sensor systems may not be used for rigid-body correction or database rigid-body updating unless r is small. This makes rigid-body pose extraction non-trivial since only positional information obtained by sensor systems may be used.

Local calibration methods map various error sources which contribute to robot positioning inaccuracy in localized regions. Robot inaccuracy can then be corrected for any target within the mapped region. If the parts on which the robot operates are delivered to the robot workcell to within specific tolerances, inaccuracy compensation need only be applied one time to the targets. Otherwise, the part rigid-body deviation must be detected with measurement devices, the database corrected, and the target recompensated in real-time. This section will consider 2-D and 3-D rigid-body pose extraction algorithms, along with correction algorithms for compensating small rigid-body deviations in the range of 1° – 5° . Larger deviations are considered by Wang.¹

a. 3-D rigid-body correction

In this section the detailed 3-D derivation of a generic rigid-body extraction method is presented. This method assumes that the rigid-body deviations are small ($<5^\circ$), and a differential homogeneous transformation can be used to represent the rigid-body perturbation of a target coordinate frame. Non-linear equations are linearized to obtain a general linear equation for the rigid-body perturbation detected by the sensors.

Figure 10 represents a perturbed rigid-body in 3-D space, defined by

F = part reference frame, used to update the rigid-body pose.

F' = perturbed part reference frame.

M_i = measurement feature frame for feature i .

M'_i = perturbed measurement feature frame for feature i .

S_i = sensor frame for feature i .

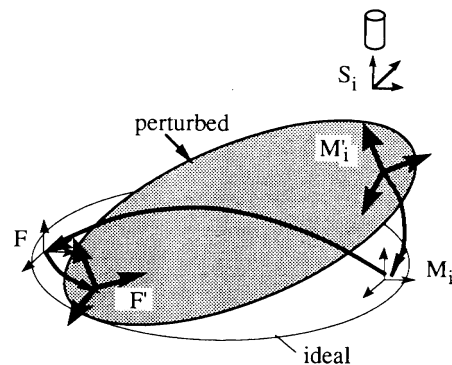


Fig. 10. Rigid-body perturbation in 3-D space.

To update the geometry database using the new rigid-body pose, this perturbation will be represented in the form of a homogeneous transformation $\mathbf{T}_F^{F'}$, as determined by various sensors. The transformation equation of the closed loop $M'_i \rightarrow M_i \rightarrow F \rightarrow F' \rightarrow M'_i$ follows:

$$\mathbf{T}_F^{F'} \mathbf{T}_F^{M'_i} = \mathbf{T}_F^{M_i} \mathbf{T}_F^{M'_i} \quad (24)$$

The transformation from frame M'_i to F is the same as that from frame M_i to F , equation (25), because M'_i and F are the corresponding perturbed frames of frames M_i and F which are attached to the same rigid-body.

$$\mathbf{T}_F^{M'_i} = \mathbf{T}_F^{M_i} \quad (25)$$

Substituting equation (25) into equation (24) and representing the transformations in the form of rotation submatrices and position vectors gives

$$\begin{bmatrix} \mathbf{R}_F^{F'} & \mathbf{p}_F^{F'} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_F^{M'_i} & \mathbf{p}_F^{M'_i} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_F^{M_i} & \mathbf{p}_F^{M_i} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{M_i}^{M'_i} & \mathbf{p}_{M_i}^{M'_i} \\ 0 & 1 \end{bmatrix} \quad (26)$$

Carrying out matrix multiplication,

$$\begin{bmatrix} \mathbf{R}_F^{F'} \mathbf{R}_F^{M'_i} & \mathbf{R}_F^{F'} \mathbf{p}_F^{M'_i} + \mathbf{p}_F^{F'} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_F^{M_i} \mathbf{R}_{M_i}^{M'_i} & \mathbf{R}_F^{M_i} \mathbf{p}_{M_i}^{M'_i} + \mathbf{p}_F^{M_i} \\ 0 & 1 \end{bmatrix} \quad (27)$$

Using positional perturbations only,

$$\mathbf{R}_F^{F'} \mathbf{p}_F^{M'_i} + \mathbf{p}_F^{F'} = \mathbf{R}_F^{M_i} \mathbf{p}_{M_i}^{M'_i} + \mathbf{p}_F^{M_i} \quad (28)$$

Rearranging equation (28) and singling out the positional perturbations results in (29), the rigid-body pose extraction equation in matrix form becomes

$$\begin{aligned} \mathbf{p}_{M_i}^{M'_i} &= [\mathbf{R}_F^{M_i}]^{-1} [\mathbf{R}_F^{F'} \mathbf{p}_F^{M'_i} + \mathbf{p}_F^{F'} - \mathbf{p}_F^{M_i}] \\ &= [\mathbf{R}_F^{M_i}]^{-1} [(\mathbf{R}_F^{F'} - I) \mathbf{p}_F^{M_i} + \mathbf{p}_F^{F'}] \quad (29) \end{aligned}$$

The screw rotation transformation \mathbf{R} represents a rotation around an arbitrary unit vector \mathbf{k} through the origin, equation (30):

$$\mathbf{R} = \begin{bmatrix} k_x k_x \text{vers } \theta + \cos \theta & k_y k_x \text{vers } \theta - k_z \sin \theta & & \\ k_x k_y \cos \theta + k_z \sin \theta & k_y k_y \text{vers } \theta + \cos \theta & & \\ k_x k_z \text{vers } \theta - k_y \sin \theta & k_y k_z \text{vers } \theta + k_x \sin \theta & & \\ & k_z k_x \text{vers } \theta + k_y \sin \theta & & \\ & k_z k_y \text{vers } \theta - k_x \sin \theta & & \\ & k_z k_z \text{vers } \theta + \cos \theta & & \end{bmatrix} \quad (30)$$

where k_x, k_y, k_z are the direction cosines of \mathbf{k} , and

$$\text{vers } \theta = 1 - \cos \theta. \quad (31)$$

For small rotational perturbations, $\theta = d\theta$, $\sin d\theta = d\theta$, and $\cos d\theta = 1$. Let the origin perturbation be represented by d_x, d_y , and d_z , and let $\delta_x = k_x d\theta$, $\delta_y = k_y d\theta$, $\delta_z = k_z d\theta$ represent the rotational perturbation. This results in the differential transformation

$$d\mathbf{T} = \begin{bmatrix} 1 & -\delta_z & \delta_y & d_x \\ \delta_z & 1 & -\delta_x & d_y \\ -\delta_y & \delta_x & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (32)$$

For rigid-body perturbation, equation (32) relates the perturbed part reference frame F' to the part reference frame F . The transformation from the measurement feature frame for feature i to the part reference frame F

is stored in the ideal database, and represented by

$$\mathbf{T}_F^{M_i} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}_i \quad (33)$$

Cell initialization establishes the ideal value represented by (33) when the part first enters the processing station. Thereafter, rigid-body deviations, sometimes referred to as offsets, are measured by external sensors. Positional deviations of the measurement feature frame M_i can be determined by sensors, either attached to the robot or attached externally:

$$\mathbf{p}_{M_i}^{M'_i} = \begin{bmatrix} \Delta x_i \\ \Delta y_i \\ \Delta z_i \end{bmatrix} \quad (34)$$

Generalizing for any measurement frame (dropping the i subscript for simplification), and substituting equations (32)–(34) into equation (29), we get

$$\begin{aligned} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} &= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}^{-1} \left\{ \begin{bmatrix} 1 & -\delta_z & \delta_y \\ \delta_z & 1 & -\delta_x \\ -\delta_y & \delta_x & 1 \end{bmatrix} \right. \\ &\quad \left. - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right\} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} \\ &= \begin{bmatrix} r_{11} & r_{21} & r_{31} \\ r_{12} & r_{22} & r_{32} \\ r_{13} & r_{23} & r_{33} \end{bmatrix} \left\{ \begin{bmatrix} 0 & -\delta_z & \delta_y \\ \delta_z & 0 & -\delta_x \\ -\delta_y & \delta_x & 0 \end{bmatrix} \right. \\ &\quad \left. \times \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} \right\} \\ &= \begin{bmatrix} r_{11} & r_{21} & r_{31} \\ r_{12} & r_{22} & r_{32} \\ r_{13} & r_{23} & r_{33} \end{bmatrix} \begin{bmatrix} -\delta_z p_y + \delta_y p_z + d_x \\ -\delta_x p_z + \delta_z p_x + d_y \\ -\delta_y p_x + \delta_x p_y + d_z \end{bmatrix} \quad (35) \end{aligned}$$

The unknowns $\delta_x, \delta_y, \delta_z, d_x, d_y$, and d_z , can be determined by measuring six positional deviations from at least three measurement frames to generate six linearly independent equations of the type shown in (36), where Δx_i is used to represent a position deviation in the x, y , or z direction as measured in the i th measurement frame.

$$\begin{aligned} \Delta x_i &= (r_{i3} p_y - r_{i2} p_z) \delta_x + (r_{i1} p_z - r_{i3} p_x) \delta_y \\ &\quad + (r_{i2} p_x - r_{i1} p_y) \delta_z + \sum_{j=1}^3 r_{ij} d_{xj} \quad (36) \end{aligned}$$

In the next section we demonstrate how equation (36) can be implemented for a typical 2-D application.

b. 2-D rigid-body application

Consider a rigid-body lying the X – Y plane, Figure 11, such that $\delta_x = \delta_y = d_{x3} = 0$, $r_{13} = r_{23} = r_{31} = r_{32} = 0$, and $r_{33} = 1$. The three unknowns, d_x, d_y , and δ_z , can be determined from three independent equations of type

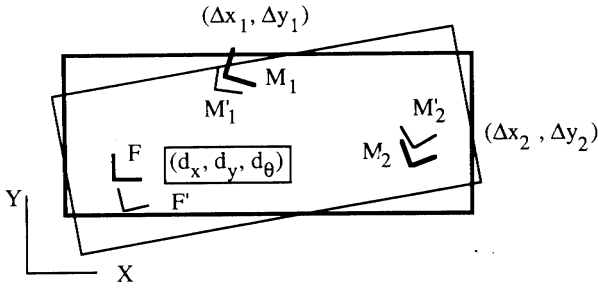


Fig. 11. Small rigid-body perturbation in 2-D space.

(36), measured in at least two measurement frames. A vision camera can measure Δx , Δy , and $\Delta \theta$. We assume that r is too large to use $\Delta \theta$ in the rigid-body correction. Using two vision cameras we determine perturbations Δx_1 , Δy_1 , Δx_2 , and Δy_2 . Since only three are required, we could use multiple combinations to statistically improve our solution or handle special cases that may occur.

First apply Δx_1 , Δy_1 , and Δx_2 in the form of (36) to get

$$\Delta x_1 = (r_{12}p_x - r_{11}p_y)\delta_z + r_{11}d_x + r_{12}d_y \quad (37a)$$

$$\Delta y_1 = (r_{22}p_x - r_{21}p_y)\delta_z + r_{21}d_x + r_{22}d_y \quad (37b)$$

$$\Delta x_2 = (r'_{12}p'_x - r'_{11}p'_y)\delta_z + r'_{11}d_x + r'_{12}d_y \quad (37c)$$

where the prime refers to the second measurement frame. The three unknowns d_x , d_y , and δ_z can now be determined, provided the equations are linearly independent.

For the special case where frames M_1 , M_2 , and F are parallel ($r_{12} = r_{21} = r'_{12} = r'_{21} = 0$, and $r_{11} = r_{22} = r'_{11} = r'_{22} = 1$), equations (37) reduce to

$$\Delta x_1 = -p_y\delta_z + d_x \quad (38a)$$

$$\Delta y_1 = p_x\delta_z + d_y \quad (38b)$$

$$\Delta x_2 = -p'_y\delta_z + d_x \quad (38c)$$

Solving these equations,

$$\delta_z = \frac{\Delta x_1 - \Delta x_2}{p'_y - p_y} \quad (39a)$$

$$d_x = \Delta x_1 + p_y\delta_z \quad (39b)$$

$$d_y = \Delta y_1 - p_x\delta_z \quad (39c)$$

It is obvious that there will be no solution if $p'_y = p_y$, or if frames M_1 and M_2 sit on a line which is parallel to the X -axis of frame F . In this case, we apply Δx_1 , Δy_1 , and Δy_2 to get

$$\Delta x_1 = -p_y\delta_z + d_x \quad (40a)$$

$$\Delta y_1 = p_x\delta_z + d_y \quad (40b)$$

$$\Delta y_2 = p'_x\delta_z + d_y \quad (40c)$$

Solving this set of equations,

$$\delta_z = \frac{\Delta y_1 - \Delta y_2}{p_x - p'_x} \quad (41a)$$

$$d_x = \Delta x_1 + p_y\delta_z \quad (41b)$$

$$d_y = \Delta y_1 - p_x\delta_z \quad (41c)$$

Again, there will be no solution if $p'_x = p_x$, or if frames M_1 and M_2 sit on a line which is parallel to the Y -axis of frame F , but in this case we resort back to equations (39).

APPLICATIONS

TCF calibration accuracy depends on the accuracy and repeatability of the calibration measurement devices and the robot on which the end-effectors are attached. For example, vision camera TCF calibration accuracy depends on the repeatability and accuracy of the robot, the resolution of the CCD array, and the accuracy of the pixel calibration which in turn is a function of the camera resolution and manufacturing tolerance of the image features used.

TCF calibration utilizing the digitizing tablet depends on the resolution and accuracy of the tablet, and the repeatability and accuracy of the robot. Fortunately, robot inaccuracy in localized regions is relatively small and deterministically distributed. The methods employed here move the robot in small regions (e.g., the field of view of a camera). Statistical methods can then average out the uncertainty, and the final TCF calibration results meet the desired tolerance requirements.

One is always interested in how accurately these methods actually locate the TCF and how this accuracy relates to the robot repeatability. Off-line assembly tests using an RT3000 robot successfully placed cylinders and blocks into holes and slots with accuracy near the robot's repeatability (within 0.03 mm clearance for the cylinder and 0.05 mm clearance for the block), without failure. Before local inaccuracy compensation, the robot demonstrated an inaccuracy propagation of 0.5 mm in about 150 mm; see Davies⁶ and Red⁷.

a. TCF calibration results

The automated TCF calibration methods introduced in this paper have been implemented using an X-window user interface and tested in a robotic assembly workcell using a SEIKO RT3000 robot with repeatability of 0.025 mm (0.001"). Table I and Table II list typical TCF calibration results for a low resolution (240 × 300) vision camera and a vacuum gripper, respectively. The results represent a statistical average of 10 independent tests.

From Table I, it can be seen that the methods are quite stable, with a 3σ standard deviation of the TCF origin location of about 0.025 mm and a 3σ orientation deviation of 0.23°. Likewise, from Table II, it can be concluded that the vacuum gripper has a 3σ standard deviation of the origin of 0.1 mm. Comparing the two methods, the vision camera results appear more stable than the digitizing tablet. But, with statistical averaging, both methods proved reliable in locating the camera and tool TCF's.

The TCF calibration process is fast, requiring less than a minute per end-effector. Using a high resolution camera will reduce the calibration time to seconds per end-effector. And at the conclusion of each TCF calibration, the newly acquired TCF pose data automatically updates a process database contained in the ROBLINE Off-Line Task Planning System developed at Brigham Young University. The assembly task can now be conducted as a set of target moves with the new TCF pose contained in the database.

Table I. TCF calibration results for a vision camera.

Test No	X (mm)	Y (mm)	θ (degree)
1	-46.350300	-88.038078	-27.048596
2	-46.356750	-88.047714	-26.946753
3	-46.359779	-88.046516	-26.797978
4	-46.354568	-88.052582	-27.017435
5	-46.348026	-88.066490	-26.997996
6	-46.363510	-88.061516	-26.992614
7	-46.362331	-88.056442	-26.852039
8	-46.364265	-88.041992	-26.900480
9	-46.353237	-88.046227	-26.961149
10	-46.357750	-88.059761	-27.009386
Average	-46.357052	-88.051732	-26.952443
Standard Deviation	0.005264	0.008692	0.075597

b. Assembly application

Figure 12 depicts the layout of the robotic vertical assembly workcell. Two tasks are developed to observe robot performance, with and without inaccuracy compensation. The first task commands the robot to move the vision camera (or simply camera) to several mapping features, with compensation on and off. This allows us to verify quantitatively the robot inaccuracies, and then prove that these inaccuracies can be compensated to within satisfactory tolerances. The second task uses the vacuum gripper to pick/place blocks and IC chips. The parts (IC chips 1 and 2, cylinder 3, and square blocks 4, 5, and 6) are picked up from the part board and placed in the corresponding locations on the assembly board. The operations require positioning tolerances of 0.002" (0.0508 mm) and 0.2° for orientation.

The Z distance for the vacuum gripper Z_{vac} can be directly measured by a vertical caliper relative to the robot's physical tool interface surface. The Z distance for the vision camera Z_{cam} can also be measured directly, except that the distance between the object plane and the last joint frame (LJF) needs to be measured when the camera is focused on features on the object plane. The Z_{cam} distance for the vision camera is calibrated by the following steps:

1. Jog the robot such that the vacuum gripper is touching the object plane.
2. Save LJF position in Z axis direction, Z_{LJF1} .
3. Jog the robot such that the camera is focused on features on the object plane.

4. Save LJF position in Z axis direction, Z_{LJF2} .

5. Calculate camera $Z_{cam} = Z_{vac} + Z_{LJF2} - Z_{LJF1}$.

The robot tasks are conducted in two local areas bounded by the part board and the assembly board. As part of the initial rigid-body updating of the part and assembly boards, the vacuum gripper is used to touch off the parts, automatically adjusting the Z component of the part frames in the database. The two local areas of interest have been defined as local regions p_region (for part board) and a_region (for assembly board). These procedures are only conducted one time at cell initialization.

The vision camera and the vacuum gripper are attached to the SEIKO RT-3000 robot distal link, then the vision camera TCF and the vacuum gripper TCF are calibrated for x , y , and θ using the automatic TCF calibration techniques introduced earlier.

The calibration procedures require that the end-effector TCF Z distances be calibrated first. Direct contact measurement can be applied to calibrate the TCF Z distances for the vision camera and the vacuum gripper, Figure 13.

Mapping features are set in the database through a workcell editing panel. For regions p_region and a_region, eight mapping features are available, namely p_ref, p1, p2, p3, a_ref, a1, a2, and a3. Features p_ref and a_ref are defined as the rigid body reference frames for regions p_region and a_region respectively. Figure 14 shows the features, feature frames and working targets (frames) in the regions.

Table II. TCF calibration results for a vacuum gripper.

Test No	X (mm)	Y (mm)	Test No	X (mm)	Y (mm)
1	-54.047077	-17.161591	6	-54.137531	-17.248152
2	-54.063084	-17.177086	7	-54.123566	-17.174816
3	-54.048439	-17.152706	8	-54.116657	-17.165077
4	-54.140965	-17.193359	9	-54.066551	-17.164625
5	-54.085217	-17.215057	10	-54.066856	-17.141228
Average:		$X_{av} = -54.089594$			$Y_{av} = -17.179370$
Standard Deviation:		$\sigma_x = 0.034777$			$\sigma_y = 0.030215$

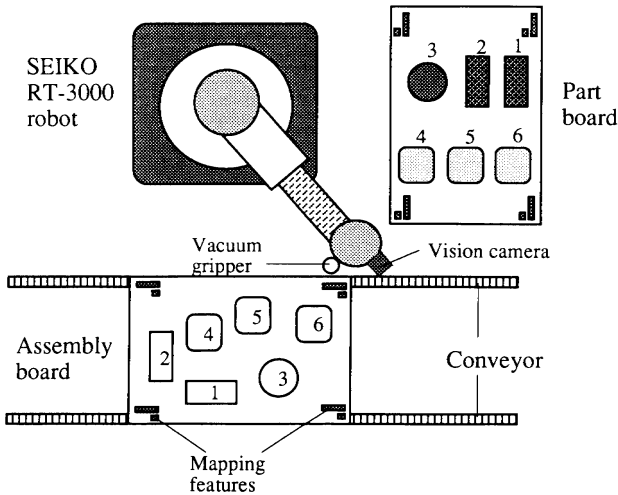


Fig. 12. Vertical assembly workcell.

The SEIKO robot is then calibrated using the local calibration techniques by Wang,¹ resulting in the accuracy improvements shown in Table III. Table III presents a statistical summary of the robot inaccuracies without compensation and with compensated positioning accuracy for 12 of the robot configurations. Among the 12 configurations, four configurations result from the camera acquiring features on the assembly board, while the rest of the configurations result from the camera acquiring features on the part board.

As verified experimentally the SEIKO RT-3000 has advertised positioning repeatability of $\pm 0.001''$ (0.0254 mm). From Table III, it can be concluded that local compensation along with the TCF and rigid-body calibration methods improved the robot positioning accuracy to near the repeatability of the robot, about a 15 times decrease in the robot positioning inaccuracy. The orientation accuracy improved from $0.05^\circ \sim 0.06^\circ$ to about 0.02° .

The assembly task was then carried out with inaccuracy compensation. All six components (IC chips and pegs of different shapes) were picked up from the part board, and successfully placed into holes or pockets on the assembly board. The IC chips, parts 1 and 2,

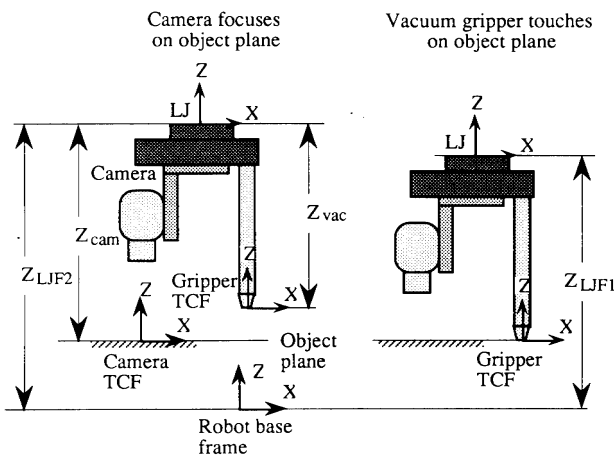


Fig. 13. TCF Z distance for a vision camera and a vacuum gripper.

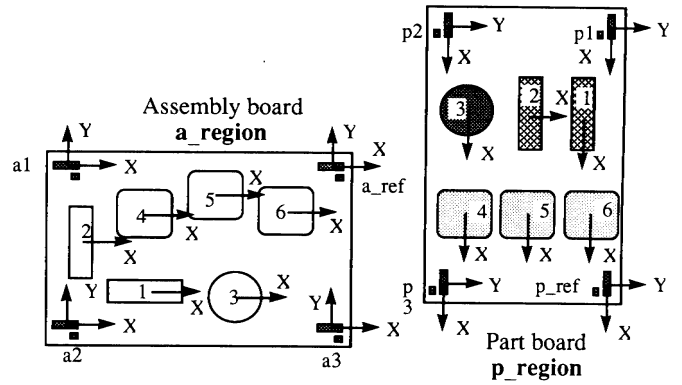


Fig. 14. Features in Regions "p_region" and "a_region".

require $\pm 0.005''$ (0.127 mm) positioning accuracy and orientation compensation. The cylindrical peg, part 3, required $\pm 0.0015''$ (0.0381 mm) positioning accuracy. Obviously, no orientation compensation was necessary. The square blocks 4, 5 and 6 required different positioning accuracy, $\pm 0.006''$ (0.1524 mm), $\pm 0.004''$ (0.1016 mm), and $\pm 0.002''$ (0.0508 mm) respectively, with orientation compensation. Without fail, the parts were assembled correctly. Tolerance stackup analysis, which accounts for tool repeatability, camera resolution, part geometry database error, part delivery system repeatability, robot repeatability, and robot inaccuracy correction error, demonstrates that the TCF pose prediction is better than the robot's repeatability.

CONCLUSIONS

Off-line programming can only work if a robot is accurate and repeatable, or, if not, methods can be applied to correct for its inaccuracy. But this is a necessary condition, since uncertainties in the tool pose and the part rigid-body pose will make it impossible for the kinematic loop to be closed. This paper presented a closed form solution for TCF calibration of orthogonal end-effectors. The methods considered TCF calibration for a camera and for a simple vacuum gripper. The measurement methods successfully locate a TCF origin with 3σ standard deviations near the repeatability of the robot and the frame orientation with 3σ orientation deviations of about 0.25° .

The TCF calibration methods are easily automated and require less than a minute of relative moves to predict the TCF accurately. Assembly tests have been successfully conducted with tolerances approaching the robot repeatability. These tests were programmed off-line and did not use a teach pendant.

This paper also considered methods for measuring and compensating small rigid-body perturbations ($< 5^\circ$) that result from non-repeatable part delivery systems or from geometric distortion. These methods are designed to eliminate $r\theta$ error from the rigid-body prediction and can be conducted in real-time. Without accurate TCF calibration and rigid-body compensation, even the most accurate robot will fail to complete an off-line programmed task if the task tolerances are stringent.

Table III. Seiko RT-3000 robot performance in a_region and p_region with compensation ON and OFF.

TEST #1			
Compensation	ΔX (mm)	ΔY (mm)	$\Delta \theta$ (degree)
OFF (worst case)	0.6985	-0.6198	-0.1254
OFF (average)	0.1829	-0.0051	0.0530
OFF (standard deviation)	0.3327	0.3531	0.0567
ON (worst case)	-0.0330	-0.0229	-0.0443
ON (average)	0.0000	0.0000	0.0009
ON (standard deviation)	0.0203	0.0127	0.0228
TEST #2			
Compensation	ΔX (mm)	ΔY (mm)	$\Delta \theta$ (degree)
OFF (worst case)	0.6756	0.6248	-0.1251
OFF (average)	0.1803	0.0000	-0.0289
OFF (standard deviation)	0.3302	0.3556	0.0625
ON (worst case)	0.0483	-0.0305	0.0517
ON (average)	0.0025	0.0025	0.0040
ON (standard deviation)	0.0203	0.0127	0.0218

References

1. X. Wang, "Automation of Robot Local Calibration and Compensation Methods" *PhD Dissertation* (Brigham Young University, August, 1991).
2. J.M. Hollerbach, "A Survey of Kinematic Calibration". In: *The Robotics Review 1, 1989* (ed O. Khatib, J.J. Craig, and T. Lozano-Perez). (The MIT Press, Cambridge, Mass., 1989) pp. 207-242.
3. J. Ziegert and P. Datsoris, "Basic Considerations for Robot Calibration" *Proceedings of IEEE International Conference on Robotics and Automation* (1988) pp. 932-938.
4. Z. Roth, B.W. Mooring and B. Ravani, "An Overview of Robot Calibration" *IEEE Journal of Robotics and Automation*, **RA-3**, 377-385 (1987).
5. Z. Roth, B.W. Mooring and B. Ravani, "Robot Precision and Calibration Issues in Electronic Assembly" *Proceedings of Southern 86* (Orlando, FL, March 18-20, 1986) pp. 227.
6. B.R. Davies, W.E. Red, and J.S. Lawson, "The Local Calibration Method for Robot Inaccuracy Compensation" *J. Robotic Systems* **7**, No. 6, 833-864 (1990).
7. W.E. Red, X. Wang and E. Turner, "Device and Method for Correction of Robot Inaccuracy" *United States Patent* 4,831,549 (May 16, 1989).
9. F.C. Park and B.J. Martin, "Robot Sensor Calibration: Solving $AX = XB$ on the Euclidean Group" *IEEE Transactions on Robotics and Automation* **10**, No. 5, 717-721 (1994).
10. B. Preising and T.C. Hsia, "Robot Performance Measurement and Calibration using a 3D Computer Vision System" *Robotica*, **13**, 327-337 (1995).
11. Y.C. Shiu and S. Ahmad, "Calibration of Wrist-Mounted Robotic Sensors by Solving Homogeneous Transform Equations of the Form $AX = XB$ " *IEEE Transactions on Robotics and Automation*, **5**, No. 1, 16-29 (1989).
12. B.H. Yoshimi and P.K. Allen, "Alignment Using an Uncalibrated Camera System" *IEEE Transactions on Robotics and Automation*, **11**, No. 4, 516-521 (1995).