A noise-tolerant algorithm for robot-sensor calibration using a planar disk of arbitrary 3D orientation

Wenyu Chen, Jia Du, Wei Xiong, Yue Wang, Shueching Chia, Bingbing Liu, Jierong Cheng, Ying Gu

Abstract—In a 3D scanning task, a robot-sensor system controls a robotic arm to move a laser sensor. In order to align the coordinate system of the robotic arm and laser sensor, prior calibration is required to derive the transformation between both coordinate systems. This paper proposes a new calibration method in three steps: manual data collection, sensing data calculation, and transformation solution. Firstly, at least four data are required to be collected by the user. The sensing data is then calculated from the collected data and adopted to provide the desired transformation. The proposed algorithm has two features: arbitrary placement of planar disk and noise tolerant. Using a planar disk, the algorithm will automatically derive the angular relationship between the disk and the sensor plane, enabling arbitrary orientation placement. Noise tolerant is guaranteed by fitting ellipses during the sensing data calculation and using a single set of sensing data in transformation solution. Experiments and comparisons are given to demonstrate the efficiency of the proposed calibration algorithm.

Keywords—Robot-sensor calibration, noise-tolerant, calibration disk, elliptic fitting.

I. INTRODUCTION

CALIBRATION is to determine the relationship among different measuring devices [1–8]. In any tool/flange or hand/eye integrated system involving moving sensor [4–6], finding the geometry transformation between two coordinate systems is crucial for achieving accurate three-dimensional geometry. Such calibration is widely used in reverse engineering [7], robot-assisted medical applications [8], visually guided robot grinding [9], and robotic visual inspection [10]. Calibration methods vary in different aspects, such as calibration targets, data acquisition, and ways to derive the transformation.

All calibrations require some form of calibration targets such as 2D boards or 3D sphere-shaped balls. Usually, different types of sensors use different calibration targets. For example, cameras uses 2D calibration boards [5] while the profile laser sensors use 3D calibration balls [11–15]. In comparison, a 2D disk is cheaper and convenient, but has not been used as calibration target for laser sensors. The challenge is to account for the orientation of the 2D disk in the base coordinate system during the calibration, and estimation of the orientation has not been discussed before. The fixed-point calibration algorithm was introduced in [16] where the calibration target is a fixed cross-point printed on a 2D disk rather than the entire disk.

A profile laser sensor projects a laser line on the object and provides a 2D profile in the sensor plane. Different calibration methods have been developed for profile laser sensors, which require users to manually move the laser line to certain positions to acquire the sensing data. For example, the fixed-point calibration algorithm [16] required the users to manually move the laser line until it passes through the specific fixed point. On the other hand, the ball-based calibration algorithms [11–14] perform better as users can project a laser line on any part of the ball. Normally, one set of sensing data is required for the desired transformation [15]. But in two-step calibration methods [11–14], users need to acquire two different sets of sensing data: one set uses robotic arm with a fixed orientation while the other set uses robotic arm with different orientations. The two sets of data are used to solve the rotation and translation aspects of the transformation, respectively.

Once the sensing data for calibration are obtained, the desired transformation is solved in different manners. Traditional calibration methods are by means of homogeneous transformation. The well-known hand/eye calibration was formulated by Shiu and Ahmad [4] and Tsai and Lenz [5] by solving a homogeneous transform equation using nonlinear minimization [7], dual quaternions [15], etc. Another solution is the two-step calibration algorithm which derives the rotation and translation portions separately [11–14]. Two limitations arise from the two-step algorithms. Firstly, noise in the sensing data and errors caused in the first step will be propagated to the second step [19]. Secondly, when more sensing data is required, the amount of intensive manual labour work is also increased. Minimal manual work in data collection for the

W. Chen, J. Du, W. Xiong, Y. Wang, S. Chia, B. Liu, J. Cheng, and Y. Gu are with Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore.(e-mail: chenw@i2r.a-star.edu.sg)
calibration is desired for better accuracy and faster processing. For the calibration of a profile laser sensor mounted onto a robotic arm, this paper targets to develop a new noise-tolerant calibration algorithm with a simple calibration target, easy data acquisition procedure, and lesser sensing data compared to existing techniques. A planar disk is chosen as the calibration target. To acquire the sensing data for the calibration, users can project the laser line onto the 2D disk without the need to pass through any specific points, including the disk center. The proposed algorithm will be based on a single set of sensing data. Not only the amount of manual effort in data collection is reduced, but the amount of error propagation is also reduced. Most importantly, the calibration algorithm is robust to noise in laser sensing data and mechanical vibrations during arm movement.

This paper is structured as follows. Section II introduces the research background, and Section III gives the preliminaries. After that, the proposed calibration method is presented in Section III. Section IV proves the formulae used during calibration. Experimental results and comparisons are given in Section V. Finally, Section VI concludes this paper.

II. SYSTEM AND NOTATIONS

A. Robot-sensor system

A robot-sensor system uses a robotic arm connected to a profile sensor to perform a 3D scanning on target objects. The coordinate systems are shown in Fig.1. The robotic arm has a fixed base coordinate system \( \Phi \). At the \( i \)-th position and orientation, the Tool Center Point (TCP, tool0) coordinate system and the sensor coordinate system are denoted as \( \Psi_i \) and \( \Gamma_i \), respectively. Notations are as follows

- \( R_{\text{TCP},i} \) is the rotation matrix from \( \Psi_i \) to \( \Phi \).
- \( T_{\text{TCP},i} \) is the translation vector from \( \Psi_i \) to \( \Phi \).
- \( R_{\text{sensor}} \) is the rotation matrix from \( \Gamma_i \) to \( \Psi_i \).
- \( T_{\text{sensor}} \) is the translation vector from \( \Gamma_i \) to \( \Psi_i \).

Since the profile sensor is mounted on TCP, for TCP at different positions with different orientations, \( R_{\text{sensor}} \) and \( T_{\text{sensor}} \) remain unchanged. The calibration aims to derive \( R_{\text{sensor}} \) and \( T_{\text{sensor}} \) based on the data from the robotic arm and the sensor. At the \( i \)-th position and orientation, the profile sensor generates a planar profile in the sensor plane \( \pi_i \). Its coordinate system \( \Gamma_i = \{ o_i, x_i, y_i, z_i \} \) is defined such that the origin \( o_i \) locates in the plane \( \pi_i \) and \( y_i \) is the normal of \( \pi_i \). Following such notations, the second coordinate of the 2D profile from the sensor is fixed as zero, i.e., \( y = 0 \) for the profile data.

B. Calibration target

A calibration target is an object where laser lights will be shone during data collection. Usually, 3D balls are preferred, because the sensor has a uniform view from any position and orientation [15]. A ball with known radius has been used as the calibration target for robot-laser sensor calibration [11]–[14]. Different from previous methods, we plan to calibrate the system using a 2D disk other than a 3D object. Fig.2 presents two forms of planar disks with known radius.

Profile data from the laser-scanning sensor can be divided into two types: chord data and non-chord data. On each profile, a chord is the line segment inside the disk, and the remnant data are non-chord data. Data preprocessing can remove the non-chord data in the profile. In Fig.2(a), non-chord data and chord data are on two different lines. Thus, non-chord data can be easily removed based on the profile’s geometry. In Fig.2(b), non-chord data in black regions can be removed based on the profile’s image. In this paper, experiments will be conducted using the 2D disk shown in Fig.2(a) and chords are derived using the split-and-merge algorithm [20]. Take the following notation:

- \( \pi_c \) is the plane where the disk locates,
- \( q \) is the center of the disk,
- \( r_c \) is the radius of the disk,
- \( n_c \) is the normal of \( \pi_c \), and
- \( c_{i,j} = [a_{i,j}, b_{i,j}] \) is a chord with two tips \( a_{i,j} \) and \( b_{i,j} \).

The new calibration method is dependent of \( q \) and \( n_c \). Namely, the position and the orientation of the planar disk do not affect the calibration. Therefore, the planar disk can be placed at any position with any orientation for the calibration.

C. Notations

For TCP at a position \( T_{\text{TCP},i} \) with orientation \( R_{\text{TCP},i} \), the sensor plane is \( \pi_i \) with the corresponding sensor coordinate system \( \Gamma_i \). For disk-based calibration, \( n \) sensing data are required as

\[
\{ T_{\text{TCP},i}, R_{\text{TCP},i}, q^i, D_{i1}, D_{i2}, \ldots, D_{in} \}, \quad i = 1, 2, \ldots, n,
\]
where

- $R_{tcp,i}$ and $T_{tcp,i}$ are TCP orientation and position for $\pi_i$ passing through the disk center $q_i$.
- $q_i = (q_i^x, 0, q_i^z)$ gives the coordinates of $q$ in $\Gamma_i$, and
- $D_i$ is a TCP offset with coordinates $D_i^x$ in $\Phi$, and coordinates $D_i^y$ in $\Gamma_i$.

Users can manually control the robotic arm and visually check the laser until it passes through the disk center $q_i$. However, due to human error and noise from the scanning system, it is difficult to achieve high accuracy. Thus, in our system, users only need to select a TCP orientation $R_{tcp,i}$ and two TCP positions. The system will automatically calculate the desired $T_{tcp,i}$.

For any physical point $p$, $p^{xyz}$ marks its coordinates in the coordinate system $Oxyz$. Following this notation, $p = p^{base}$ denotes the coordinates of $p$ in the base coordinate system $\Phi$. A TCP position $T_{tcp,i}$ is always in the base coordinate system, i.e. $T_{tcp,i} = T_{tcp,i}^{base}$. For the $i$-th sensing data, when the sensor plane passes through the disk center, the sensor coordinate system is $\Gamma_i = 0, x_i, y_i, z_i$. Then, $p^{\phi,\pi,\gamma,\delta}$ becomes the coordinates of $p$ in $\Gamma_i$. During the calibration, $\Gamma_i$ will be rotated to the rotated coordinate system $\Gamma_{i,r}$.

The point pair $p^{base}$ and $p^{\phi,\pi,\gamma,\delta}$ refers to the same physical point. Each pair is named as a conjugate pair. Multiple conjugate pairs are required to enable the calibration [15]. Referring to Fig.1, for the $i$-th sensing data, a conjugate pair $q^{i}$ and $q^{i}$ satisfies

$$q^{i} = R_{tcp,i}R_{sensor}q^{i} + R_{tcp,i}T_{sensor} + T_{tcp,i},$$

i.e.

$$q^{i} = R_{tcp,i}R_{sensor}q^{i} + R_{tcp,i}T_{sensor} + T_{tcp,i}. \quad (1)$$

In this equation, the coordinate $q^{i}$ is unknown before the calibration, as the disk is freely placed in the scanning system. The rotation matrix $R_{tcp,i}$ can be read from the robotic arm. The translation $T_{tcp,i}$ and the coordinate $q^{i}$ will be calculated according to the chords. The aim of the calibration is to obtain the rotation matrix $R_{sensor}$ and the translation $T_{sensor}$.

### III. NOISE-TOLERANT CALIBRATION ALGORITHM

Fig.3 is the flowchart for the calibration process, where

- $n$ is the number of sensing data to be collected,
- $m$ is the number of parallel chords to be scanned for the $i$-th sensing data,
- $T_{tcp,i}$ and $T_{tcp,i}$ are two TCP positions manually selected for the $i$-th sensing data,
- $s_j$ is of value 1 or -1,
- $c_{i,j}$ are $m$ parallel chords scanned when TCP moves from $T_{tcp,i}$ to $T_{tcp,i}$,
- $\pi_i$ is the sensor plane for TCP at the $i$-th position $T_{tcp,i}$ with orientation $R_{tcp,i}$, and
- $\alpha_i$ is an angle between the sensor plane $\pi_i$ and the disk plane $\pi_c$.

According to the flowchart, the calibration algorithm contains three parts: manual data collection, sensing data calculation, and transformation solution. For the $i$-th sensing data, users only need to manually move TCP to two different positions $T_{tcp,i}$ and $T_{tcp,i}$ with the same orientation $R_{tcp,i}$. During sensing data calculation, $m$ chords are obtained and the $i$-th sensing data calculation, $\{T_{tcp,i}, R_{sensor}, q^{i}, D^{i}\}$ is calculated based on these chords. Finally, all sensing data are combined together to derive $R_{sensor}$ and $T_{sensor}$. This section will detail the key formulae and steps in the flowchart. Section IV will show how to deduct these formulae.

#### A. Manual data collection

In each scan, the 3D ball-based calibration methods [11]–[14] obtained a circular arc from the sensor and located the ball center directly from this arc. In our case, only one single chord can be obtained in each scan (Fig.2). This chord alone is not sufficient to derive the coordinates of the disk center. Multiple chords are required in order to derive the $i$-th sensing data (see details in Section IV-B). However, users do not need to manually scan each chord, which will be painful and time consuming. According to Fig.3 users only need to manually select two different TCP positions $T_{tcp,i}$ and $T_{tcp,i}$, with a fixed orientation $R_{tcp,i}$. With $M_i = T_{tcp,i} - T_{tcp,i}$, the scanning system can automatically translate TCP from $T_{tcp,i}$ by $M_i / (m - 1)$ for ($m - 1$) times to obtain $m$ chords $\{c_{i,j}\}_{j=1}^m$ (the black chords in Fig.3). To achieve a better calibration result, $T_{tcp,i}$ is selected such that the TCP’s moving direction $M_i$ is neither parallel nor perpendicular to the calibration disk, i.e.

$$M_i \not\parallel \pi_c, M_i \not\perp \pi_c. \quad (2)$$

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig3.png}
\caption{Flowchart of the noise-tolerant calibration.}
\end{figure}

---

**Note:** The image contains a flowchart and some mathematical formulae that are not fully transcribed. It is recommended to refer to the original document for a complete understanding.
With the above constraints, users can choose different \( T_{TCP,i} \) and \( \hat{T}_{TCP,i} \). The scanning can be within half of the disk (Fig.4(a)) or the whole disk (Fig.4(b)).

Besides \( T_{TCP,i} \) and \( \hat{T}_{TCP,i} \), users also need to collect a value \( s_i \), which is defined according to the TCP movement from \( T_{TCP,i} \) to \( \hat{T}_{TCP,i} \). If it follows the \( y^+ \) direction in the sensor coordinate system (Fig.5(a)), then \( s_i = 1 \). Otherwise (Fig.5(b)), \( s_i = -1 \).

Referring to Fig.5, suppose \( \hat{\Gamma}_i \) is the sensor coordinate system in the sensor plane \( \pi_i \). If \( \hat{T}_{TCP,i} \) is formulated in \( \hat{\Gamma}_i \) as \( \hat{T}_{TCP,i} = (T_{i,x}, T_{i,y}, T_{i,z}) \), \( s_i \) can also be defined as

\[
 s_i = \begin{cases} 
 1, & T_{i,y} > 0, \\
 -1, & T_{i,y} < 0.
\end{cases}
\]

(3)

The value \( s_i \) is necessary for ball-based calibration algorithms as well (see details in Section IV-A). The value \( s_i \) gives users more degrees of freedom during data collection. Users can easily have \( s_i = 1 \) for all sensing data.

### B. Sensing data calculation

For the \( i \)-th sensing data, sensor automatically scan each chord \( e_{i,j} = [a_{i,j}, b_{i,j}] \) (Fig.4(a)) and provide 2D chord data as

\[
a_{i,j} = (a_{i,j,x}, 0, a_{i,j,z}), \quad b_{i,j} = (b_{i,j,x}, 0, b_{i,j,z}).
\]

(4)

These chord data can be proved to be on the same ellipse in the plane \( \pi_i \) (Fig.6) see details in Section IV-C, and there may be noise within these chord data (see details in Section IV-D). Therefore, according to Fig.5 these chord data are not directly used for sensing data calculation. Instead, ellipse fitting algorithm [21] is adopted for noise removal. The fitted ellipse will be used for calculating the desired sensing data.

There is an angle between \( x_i \) and \( \pi_i \) (Fig.5(a)). Applying a rotation matrix \( R_{\theta_i} \) in the sensor coordinate system \( \Gamma_i \) in the sensor plane \( \pi_i \), \( x_i(y_i,z_i) \) becomes the rotated coordinate system \( \hat{x}_i y_i z_i \) (Fig.7(b)). The rotation matrix \( R_{\theta_i} \) can be selected such that \( \hat{x}_i \parallel \pi_i \). In order to get the desired \( R_{\theta_i} \), the chord direction for the \( i \)-th sensing data is defined as (Fig.7(a))

\[
c_i = \frac{1}{m} \sum_{j=1}^{m} (b_{i,j} - a_{i,j}).
\]

Then, the rotation matrix \( R_{\theta_i} \) is defined by rotating \( x_i \) to \( e_i \) along \( y_i \) with angle \( \theta_i \). After rotation, points \( \{e_{i,j}\}_{j=1}^{m} \) can be derived from the ellipse under the rotated coordinate system \( \Gamma_{i,r} \) in the plane \( \pi_i \) as follows (Fig.6 see details in Section IV-D):

- \( e_{i,1} \) and \( e_{i,2} \) are two ellipse points whose tangent directions are parallel to \( x_i \) axis.
- \( e_{i,3} \) is the center of the ellipse, and
- \( e_{i,4}/e_{i,5} \) is the intersection of the chord \( e_{i,1}/e_{i,2} \) with the first/last chord.

The coordinates for \( e_{i,j} \) in \( \Gamma_{i,r} \) and \( \Gamma_i \) can be calculated as

\[
e_{i,j}^{\Gamma_{i,r}} = (e_{i,j,x}, 0, e_{i,j,z}),
\]

\[
e_{i,j}^{\Gamma_i} = R_{\theta_i}^{-1} (e_{i,j,x}, 0, e_{i,j,z}).
\]

(5)

(6)

Without loss of generality, as presented in Fig.6 \( e_{i,1} \) and \( e_{i,2} \) can be selected such that \( e_{i,1}/e_{i,2} \) and \( e_{i,4}/e_{i,5} \) are of the same direction.
For \( j = 1, 2, \ldots, 5 \), denote
\[
\eta_{i,j} = \frac{(e_{i,j} - e_{i,5}) \cdot (e_{i,4} - e_{i,5})}{|e_{i,4} - e_{i,5}|^2},
\]
and
\[
T_{i,j} = \eta_{i,j} \hat{T}_{tcp,i} + (1 - \eta_{i,j}) \hat{T}_{tcp,i}.
\]
Then, \( T_{i,1} \) and \( T_{i,2} \) are two TCP positions where the sensor plane tangent to the disk, and \( T_{i,3} \) is the TCP position where the sensor plane passes through the disk center (see details in Section IV-G).

With Eq. (5) and Eq. (8), an angle \( \alpha_i \) can be obtained as
\[
\alpha_i = \arccos \left( \frac{\|T_{i,1} - T_{i,2}\|^2 - \|e_{i,1} - e_{i,2}\|^2 - 4r^2}{4r(e_{i,1} - e_{i,2})} \right).
\]
Then, the i-th sensing data \( \{T_{tcp,i}, R_{tcp,i}, q^i, D_{i1}^t, D_{i2}^t\} \) is calculated as
\[
T_{tcp,i} = T_{i,3},
\]
\[
q^i = R_{i}^{-1} (e_{i,3,x}, 0, e_{i,3,y}),
\]
\[
D_{i1}^t = R_{i}^{-1} (e_{i,1} - e_{i,2}, 2rc_i \sin \alpha_i, e_{i,1} - e_{i,2}, 2rc_i \cos \alpha_i),
\]
\[
D_{i2}^t = T_{i,2} - T_{i,1}.
\]
Details on how to derive these formulae will be presented in Section IV-G.

**C. Transformation solution**

Robot-sensor calibration is based on all sensing data \( \{T_{tcp,i}, R_{tcp,i}, q^i, D_{i1}^t, D_{i2}^t\}_{i=1}^n \). Define two \( 3 \times n \) matrices as
\[
X = [D_{i1}^t, D_{i2}^t, \ldots, D_{in}^t],
\]
\[
Y = [R_{tcp,1}^{-1}, D_{i1}^t, R_{tcp,2}^{-1}, D_{i2}^t, \ldots, R_{tcp,n}^{-1}, D_{in}^t].
\]
Apply singular value decomposition (SVD) on the \( 3 \times 3 \) matrix \( H = XY^T \)
\[
H = USV^T,
\]
where \( U \) and \( V \) are \( 3 \times 3 \) orthogonal matrices while \( S \) is an \( 3 \times 3 \) diagonal matrix with nonnegative real numbers on the diagonal. Then, the rotation matrix can be derived as
\[
R_{sensor} = VU^T.
\]

Couppling with
\[
R_i = R_{tcp,i} - R_{tcp,i+1},
\]
\[
N_i = R_{tcp,i+1} R_{sensor} q_i^{i+1} R_{tcp,i} R_{sensor} q_i + T_{tcp,i+1} - T_{tcp,i},
\]
a \( 3 \times (n - 1) \) matrix \( M \) and a \( 3 \times (n - 1) \) matrix \( N \) can be defined as
\[
M = \begin{bmatrix}
R_1 \\
R_2 \\
\vdots \\
R_{n-1}
\end{bmatrix},
N = \begin{bmatrix}
N_1 \\
N_2 \\
\vdots \\
N_{n-1}
\end{bmatrix}.
\]

Then, the translation vector can be formulated as
\[
T_{sensor} = \left(M^T M\right)^{-1} \left(M^T N\right),
\]
where \( M^T \) is the transpose of \( M \). Section IV-H will detail how to formulate Eq. (16) and Eq. (19).

**D. Calibration accuracy**

In [13], the accuracy is estimated as the difference between the radius of the calibration target and the radius of the reconstructed target. In fact, as discussed during \( T_{sensor} \) calculation in [12], a ball can be reconstructed only if \( R_{sensor} \) is known. Thus, \( T_{sensor} \) will not affect the radius difference. In fact, radius difference only reflect the accuracy of \( R_{sensor} \). Instead of the radius difference, the position difference should be used to estimate the calibration accuracy [23].

After calibration, \( \{T_{tcp,i}, R_{sensor}, q^i\} \) can be used to derive a disk center following Eq. (1) as
\[
q^b = R_{tcp,i} R_{sensor} q^i + R_{tcp,i} T_{sensor} + T_{tcp,i},
\]
the disk center is estimated as
\[
q^b = \frac{1}{n} \sum_{i=1}^n q^b_i/n.
\]
The calibration accuracy is
\[
\max_k \|q^b - q^b_k\|.
\]
The calibration accuracy is affected by the following factors:
- the resolution and accuracy of the robotic arm,
- the resolution and accuracy of the sensor, and
- the accuracy of the ellipse fitting on the training data.

**IV. ALGORITHM FORMULATION**

During manual data collection, \( R_{tcp,i}, \hat{T}_{tcp,i}, \hat{R}_{tcp,i}, \) and \( s_i \) are collected. After that, chord data \( e_{i,j} \) are acquired from the sensor. This section will detail how the calibration algorithm is developed based on these data.

**A. Necessity of \( s_i \)**

The sign \( s_i \) in Eq. (3) is recorded during manual data collection, and used in Eq. (12) during sensing data calculation. The ball-based calibration methods [11–13] also need such a value when formulating the ball center in the sensor coordinate system. Suppose the ball centers at \( c_b \) with radius \( r_b \). In each scan, the sensor acquires a circular arc from the ball. Circle fitting algorithm is adopted to get the radius \( r_d \) and the center \( c_d = (c_{d,x}, 0, c_{d,z}) \) of the circle in the sensor plane. Then, the ball center can be formulated in the sensor coordinate system as
\[
c_b = \left(c_{d,x}, s_i \sqrt{r_b^2 - r_d^2}, c_{d,z}\right),
\]
where \( s_i \) can be -1 or 1. If the disk center locates on the positive direction in the sensor coordinate system, \( s_i = 1 \). Otherwise, \( s_i = -1 \).
B. Multiple chords based calculation

For TCP at any position, only a single chord is provided from the sensor (Fig.8(a)). If the chord passes through the disk center \( q \), the coordinates \( q' \) in the sensor coordinate system can be obtained as the chord center. Otherwise, \( q' \) cannot be directly derived from the single chord. Consider a scenario where the robotic arm and the sensor are fixed while the circular disk rotates along the chord from \( \pi_x \) to \( \pi_{x,1} \) and \( \pi_{x,2} \) (Fig.8(b)). The chord data from the sensor remains unchanged; because the geometric distance from the chord to the sensor is fixed. Meanwhile, the coordinates \( q' \) keep changing during the rotation. This means that a single chord alone is not sufficient to define \( q' \). The following sections will present that minimally two chords are required to locate the disk center. Moreover, to tolerate the noise during data acquisition, multiple chords should be collected.

C. Chord projection

Chords \( c_{i,j} \) in Fig.4 are on the disk. When the sensor scans these chords, the sensor coordinate system move. Thus, the readings from the sensor in Eq.4 are in different coordinate systems. This section will prove that the chord data in Eq.4 are on the same ellipse in the plane \( \pi_x \).

For the \( i \)-th sensing data, referring to Fig.4 TCP is translated by \( M_i/m \) times to move from \( \hat{T}_{TCP,i} \) to \( \hat{T}_{TCP,i} \). Simultaneously, \( m \) parallel sensor planes \( \{\pi_{i,j}\}^m_{j=1} \) are derived satisfying \( \pi_{i,1} = \pi_i \) and \( \pi_{i,m} = \pi_i \). In these sensor planes, \( m \) sensor coordinate systems \( \{\Gamma_{i,j} = (x_{i,j},y_{i,j},z_{i,j})\}^m_{j=1} \) also translate uniformly with (Fig.9)

\[
o_{i,j} = o_{i,1} + \frac{j - 1}{m - 1} M_i, \tag{21}\]

where \( \hat{\Gamma}_i = \Gamma_{i,1} = o_{i,1}x_{i,1}y_{i,1}z_{i,1} \) is the sensor coordinate system in the sensor plane \( \pi_i \). Since there is only one translation between the two coordinate systems \( \Gamma_{i,1} \) and \( \Gamma_{i,j} \), the coordinates of a vector in \( \Gamma_{i,1} \) equal to its coordinates in \( \Gamma_{i,j} \), i.e.

\[
M_i^o_{i,j}x_{i,j}y_{i,j}z_{i,j} = M_i^o_{i,j}x_{i,j}y_{i,j}z_{i,j}, \tag{22}\]

and the coordinates of any point \( p \) in \( \Gamma_{i,1} \) and \( \Gamma_{i,j} \) satisfy

\[
p^{o_{i,1}x_{i,1}y_{i,1}z_{i,1}} = p^{o_{i,j}x_{i,j}y_{i,j}z_{i,j}} + \frac{m - 1}{m - 1} M_i^o_{i,j}x_{i,j}y_{i,j}z_{i,j}. \tag{23}\]

Reformulate sensor data in Eq.4 for each chord \( c_{i,j} \) in \( \Gamma_{i,j} \) as (black parallel line segments in Fig.8(a))

\[
c_{i,j}^o_{i,j}x_{i,j}y_{i,j}z_{i,j} = (c_{i,j},0,0,0). \tag{24}\]

Define a line segment \( \hat{c}_{i,j} \) parallel to \( c_{i,j} \) as

\[
\hat{c}_{i,j} = c_{i,j} - \frac{j - 1}{m - 1} M_i, \tag{25}\]

and formulate it in \( \Gamma_{i,j} \) as

\[
c_{i,j}^o_{i,j}x_{i,j}y_{i,j}z_{i,j} = (c_{i,j},0,0,0). \tag{26}\]

On the other hand, substituting \( p = \hat{c}_{i,j} \) into Eq.22 gives

\[
c_{i,j}^o_{i,j}x_{i,j}y_{i,j}z_{i,j} = c_{i,j}^o_{i,j}x_{i,j}y_{i,j}z_{i,j} + \frac{j - 1}{m - 1} M_i^o_{i,j}x_{i,j}y_{i,j}z_{i,j}. \tag{27}\]

Since the \( y \) coordinate of \( c_{i,j}^o_{i,j}x_{i,j}y_{i,j}z_{i,j} \) is zero, \( \hat{c}_{i,j} \) locates in the plane \( \pi_i \). In fact, \( \{c_{i,j}\}^m_{j=1} \) are the projection of \( \{c_{i,j}\}^m_{j=1} \) to the plane \( \pi_i \) following a direction parallel to \( M_i \) (black arrows in Fig.8(b)). Thus, the input sensor data \( \{\{c_{i,j},0,0,0\}\}^m_{j=1} \) are not only the coordinates of chords \( \{c_{i,j}\}^m_{j=1} \) in different coordinate systems \( \{\Gamma_{i,j}\}^m_{j=1} \), i.e. \( \{\{c_{i,j}^o_{i,j}x_{i,j}y_{i,j}z_{i,j}\}^m_{j=1} \), but also the coordinates of line segments \( \{\hat{c}_{i,j}\}^m_{j=1} \) in the same coordinate system \( \pi_i \), i.e. \( \{c_{i,j}^o_{i,j}x_{i,j}y_{i,j}z_{i,j}\}^m_{j=1} \).

Geometrically, all \( \{c_{i,j}\}^m_{j=1} \) are on the same disk. However, not all sensor data in Eq.23 are on a disk, because they are under different coordinate systems \( \{\Gamma_{i,j}\}^m_{j=1} \). With the constraints in Eq.2, the projection of a circular disk to a plane following a direction parallel to \( M_i \) is an elliptic disk. Thus, \( \{\{c_{i,j},0,0,0\}\}^m_{j=1} \) are chords of this elliptic disk. The projected ellipse in \( \pi_i \) can be acquired using ellipse fitting algorithm [21], and the chord data in Eq.4 are on the projected ellipse.

Fig. 8. Disk rotation along the chord: (a) the sensor scan one chord on the intersection between the disk plane and \( \pi_i \), and (b) rotating the disk plane along the chord leads to the same chord from the sensor. (The green plane is perpendicular to both the disk plane and the chord)

Fig. 9. A plane perpendicular to \( \pi_{x,1} \) and passing through \( o_{i,j} \): (a) The plane in 3D with red lines as its intersection with \( \pi_{x,1} \) and red arrow as the moving direction, and (b) the plane in 2D with black dots as \( c_{i,j} \), red dots as \( o_{i,j} \), black line as \( \pi_{x,1} \), red lines as \( \pi_{x,1} \), red arrow as \( M_i \), and black arrows as projection directions.
D. Noise analysis in sensor data

In the sensor plane \( \pi_t \), the chords \( \{ \tilde{c}_{i,j} \}_{j=1}^m \) in Eq. (27) center at \( \{ s_{i,j} \}_{j=1}^m \). According to the construction of \( e_{1,1} \) and \( e_{1,2} \), all \( s_{i,j} \) are geometrically on the chord \( e_{1,1} e_{1,2} \). Formulate \( s_{i,j} \) in \( \Gamma_i \) as \( s_{i,j} = (x_{s_{i,j}}, y_{s_{i,j}}) \). Then, ideally, \( s_{i,j} \) should be uniformly distributed on a line. However, in the actual scanned data, \( (x_{s_{i,j}}, y_{s_{i,j}}) \) may not be on a same line (Fig. 10). This is due to the noise introduced by the surface condition of the calibration target, the resolution of the sensor, the vibration of the robotic arm during sensing data acquisition, and so on.

Fig. 10. Data \( (x_{s_{i,j}}, y_{s_{i,j}}) \) from the sensor do not lie exactly on the same line.

To tolerate these noises, the proposed algorithm uses the fitted ellipse instead of the original chords for sensing data calculation. Theoretically, minimally two chords are required to enable the ellipse fitting. However, to better tolerate the noises, \( m \geq 100 \) chords are collected for sensing data calculation (see details in Section V-B). For illustration purpose, figures in this paper adopts \( m = 10 \).

E. Elliptic points

During sensing data calculation, five elliptic points \( \{ e_{i,j} \}_{i,j=1}^5 \) on the projected ellipse instead of the raw sensor data are used. This section will introduce how these points are formulated as Eq. (5).

Referring to Fig. 11(a), on the circular disk, points

\[
q_{i,j} = (x_{q_{i,j}}, y_{q_{i,j}}), \quad i,j = 1,2,\cdots,5
\]  

(28)

are defined in the plane \( \pi_d \) such that

- \( q_{i,3} \) is the diameter geometrically perpendicular to the chords \( \{ e_{i,j} \}_{j=1}^m \),
- \( q_{i,3} = q \) is the disk center, and
- \( q_{i,4} \) and \( q_{i,5} \) are intersections of \( e_{i,1} q_{i,2} \) to \( c_{i,1} \) and \( c_{i,m} \), respectively.

Projecting \( q_{i,j} \) to the plane \( \pi_t \) following a direction parallel to \( M_i \) (Fig. 11(b)) leads to the projected points

\[
e_{i,j} = (x_{e_{i,j}}, y_{e_{i,j}}), \quad i,j = 1,2,\cdots,5.
\]  

(29)

According to the definition, \( \{ e_{i,j} \}_{i,j=1}^m \) are the projections of circular chords \( \{ c_{i,j} \}_{j=1}^m \),

- ellipse chords \( c_{i,1} e_{i,2} \) is the projection of the diameter \( q_{i,1} q_{i,2} \),
- center \( e_{i,3} \) is the projection of the center \( q = q_{i,3} \), and
- points \( e_{i,4} \) and \( e_{i,5} \) are intersections of \( e_{i,1} e_{i,2} \) to \( c_{i,1} \) and \( c_{i,m} \), respectively.

Fig. 11. Elliptic points via projection: (a) \( q_{i,j} \) in \( \pi_d \), and (b) \( e_{i,j} \) in \( \pi_t \). (The blue area is above \( \pi_d \) and the yellow area is below \( \pi_d \)).

In the plane \( \pi_t \), the chord \( e_{i,1} e_{i,2} \) passes through the centers of all ellipse chords parallel to \( \tilde{c}_{i,j} \). Therefore, the tangent directions at \( e_{i,1} \) and \( e_{i,2} \) are parallel to \( \tilde{c}_{i,j} \) as well. After the rotation \( R_{\Phi} \), Fig. 11(b) becomes Fig. 6(a), and the tangent directions at \( e_{i,1} \) and \( e_{i,2} \) are parallel to the axis \( \hat{x}_i \) in the rotated coordinate system \( \tilde{\Gamma}_{i,r} \), each point \( e_{i,j} \) is formulated in Eq. (5).

F. TCP positions

During data collection, users only need to manually collect two TCP positions \( T_{tcp,i} \) and \( T_{tcp,j} \). With the orientation \( R_{tcp,i} \) fixed, the proposed algorithm calculates other TCP positions where the sensor plane is passing through the disk center or tangent to the disk. Referring to Fig. 12(a), suppose that TCP positions \( \{ T_{i,j} \}_{j=1}^5 \) are linear combinations of \( T_{tcp,i} \) and \( T_{tcp,j} \). When TCP moves to \( T_{i,j} \), the sensor plane is \( \phi_{i,j} \) and the sensor coordinate system is \( \Phi_{i,j} \). With \( \phi_{i,j} \) defined in Eq. (28), the positions \( \{ T_{i,j} \}_{j=1}^5 \) are taken such that:

- \( \phi_{i,1} \) and \( \phi_{i,2} \) are tangent to the disk at the points \( q_{i,1} \) and \( q_{i,2} \), respectively.
- \( \phi_{i,3} \) passes through the disk center \( q = q_{i,3} \).
- \( \phi_{i,4} = \pi_{i,j} \) passes through the point \( q_{i,4} \).
- \( \phi_{i,5} = \pi_{i,j} \) passes through the point \( q_{i,5} \).

This section is to prove that \( T_{i,j} \) satisfies Eq. (8).

Referring to Fig. 12(b), since \( \{ e_{i,j} \}_{j=1}^5 \) are parallel projection of \( \{ q_{i,j} \}_{j=1}^5 \), similar with the chord projection in Eq. (27), the coordinates of \( q_{i,j} \) in the sensor coordinate system \( \Phi_{i,j} \) are equal to the coordinates of \( e_{i,j} \) in the sensor coordinate system \( \Phi_{i,4} \), i.e.

\[
\Phi_{i,j} = \Phi_{i,4} = q_{i,j} = e_{i,j} + \Phi_{i,4}.
\]

(30)

Since \( \phi_{i,4} = \pi_{i,j} \) and \( \Phi_{i,4} = \tilde{\Gamma}_{i,r} \), coupling with Eq. (5) and Eq. (6), the coordinates of \( q_{i,j} \) in the coordinate system \( \Phi_{i,j} \) are

\[
q_{i,j} = e_{i,j} + \Phi_{i,4} = e_{i,j} + e_{i,j} = R_{\Phi_{i,4}}^{-1} \left( e_{i,1}, e_{i,2}, 0, e_{i,3} \right).
\]

(30)

and its coordinates in the rotated coordinate system \( \Phi_{i,j,r} \) are

\[
q_{i,j} = e_{i,j} + \Phi_{i,4,r} = e_{i,j} + e_{i,j} = (e_{i,1}, e_{i,2}, 0, e_{i,3}).
\]

(31)

When TCP moves linearly, the sensor plane and the rotated coordinate system move linearly as well. Consider a point
Eq. (11). The TCP offset \( \Phi_i \), of sensing data formation coordinate system as.

Similarly, Eq. (33) leads to

\[
T_{i,j}r_{i,j} \parallel T_{i,1}r_{i,1}, \quad q_{i,j}r_{i,j} \parallel q_{i,1}r_{i,1}.
\]

Coping with Eq. (7), \( e_{i,j} \) can be formulated as

\[
e_{i,j} = \eta_{i,j}e_{i,4} + (1 - \eta_{i,j})e_{i,5}.
\]

Refering to Fig. 12(b), since \( \{e_{i,j}\}^{5}_{j=1} \) are parallel projection of \( \{q_{i,j}\}^{5}_{j=1} \), line segments \( \{e_{i,j}e_{i,5}\}^{4}_{j=1} \) and \( \{q_{i,j}q_{i,5}\}^{4}_{j=1} \) are proportional segments, i.e.

\[
\frac{\|e_{i,j}e_{i,5}\|}{\|e_{i,4}e_{i,5}\|} = \frac{\|q_{i,j}q_{i,5}\|}{\|q_{i,4}q_{i,5}\|}.
\]

Thus

\[
q_{i,j} = \eta_{i,j}q_{i,4} + (1 - \eta_{i,j})q_{i,5}.
\]

Similarly, Eq. (33) leads to

\[
r_{i,j} = \eta_{i,j}r_{i,4} + (1 - \eta_{i,j})r_{i,5}, \quad T_{i,j} = \eta_{i,j}T_{i,4} + (1 - \eta_{i,j})T_{i,5},
\]

which proves Eq. (5).

G. Sensing data formation

The \( i \)-th sensing data is \( \{T_{tcp,i}, R_{tcp,i}, q^i, D^i_1, D^i_2\} \), where \( R_{tcp,i} \) is from the robotic arm. According to the definition of \( T_{i,j} \) in Section IV.A, \( T_{i,3} \) is the TCP position where the sensor plane passes through \( q \). Thus, \( T_{tcp,i} \) can be defined as Eq. (10). According to Eq. (29), \( q^i = q^i_3 = \Phi_{i,3} = \Phi_{i,1} \), which proves Eq. (11). The TCP offset \( D_1 \) in the base coordinate system is given by Eq. (13). This section will show how to derive Eq. (12).

\( \Gamma_i \) is the sensor coordinate system for the sensor plane passing through the disk center. According to the definition, \( \Phi_{i,3} = \Gamma_i \). Suppose \( \Phi_{i,j} \) is the rotated coordinate system after applying \( R_{\theta_i} \) on \( \Phi_{i,j} \). To derive the sensing data, different points need to be formulated in the same coordinate system \( \Phi_{i,1,r} \).

Referring to Eq. (11), two different points \( r_{i,1} \) and \( q_{i,2} \) satisfy

\[
q_{i,1}^{b} = R_{tcp,i}R_{sensor}^{T_{i,1}^{\Gamma}}, \quad q_{i,2}^{b} = R_{tcp,i}R_{sensor}^{T_{i,2}^{\Gamma}} + T_{tcp,i}.
\]

Referring to Fig. 12(a), the TCP offset is

\[
D_1 = T_{i,2} - T_{i,1} = q_{i,2} - r_{i,1}.
\]

Coping with Eq. (34), \( D_1 \) can be formulated as

\[
D_1^b = q_{i,2}^{b} - r_{i,1} = R_{tcp,i}R_{sensor}D_1^{\Gamma} = R_{tcp,i}R_{sensor}D_1^{\Gamma}.
\]

Since \( p_{\Lambda} = R_{\theta_i}^{T_{i,1}^{\Gamma}} \) holds for any point \( p \), the above equation becomes

\[
D_1^b = R_{tcp,i}R_{sensor}R_{\theta_i}^{-1}D_1^{\Gamma}.
\]

The coordinates of a vector remain unchanged in \( \Phi_{i,1,r} \) and \( \Phi_{i,3,r} \), because there is only a translation between the two coordinate systems. Substituting

\[
D_1^{\Gamma} = D_1^{\Phi_{i,3,r}} = D_1^{\Phi_{i,1,r}}
\]

into Eq. (37) gives

\[
D_1^b = R_{tcp,i}R_{sensor}R_{\theta_i}^{-1}D_1^{\Phi_{i,1,r}}.
\]

In Fig. 13, \( \bar{x}_i, y_i, z_i \) is the rotated coordinate system, and \( \alpha_i \) is an angle from \( z_i \) to \( q_{i,1}q_{i,2} \). Noted that \( \pi_e \parallel \bar{x}_i \), \( y_i \parallel \bar{y}_i \), and \( \bar{z}_i \parallel \bar{z}_i \) and \( q_{i,1}q_{i,2} \) is perpendicular to \( \pi_e \) and \( \bar{x}_i \) (Fig. 13(a)). Thus, \( \alpha_i \) is an angle between \( \pi_e \) and \( \Phi_{i,1} \). Fig. 13(b) shows two possibilities for \( \alpha_i \): \( \alpha_i \geq 90^\circ \) and \( \alpha_i < 90^\circ \). According to Eq. (36), although TCP movement is not perpendicular to \( \pi_e \), \( \Phi_{i,1} \) may be perpendicular to \( \pi_e \), i.e. \( \alpha_i = 90^\circ \).

To derive \( \alpha_i \), points \( q_{i,1}, q_{i,2} \) and \( r_{i,1} \) also need to be formulated in the same coordinate system \( \Phi_{i,1,r} \). Eq. (31) and Eq. (32) gives

\[
\Phi_{i,1,r} = (e_{i,1,x}, 0, e_{i,1,z}) \quad \alpha_i.
\]

According to the definition of \( q_{i,1} \) in Eq. (28) and \( s_i \) in Eq. (4), if \( s_i = 1 \) (Fig. 13(b)), \( q_{i,2} \) is on \( y^+ \) direction of \( \Phi_{i,1,r} \), Otherwise, \( q_{i,2} \) is on \( y^- \) direction of \( \Phi_{i,1,r} \). Thus, in the yellow plane
perpendicular to $\bar{x}$, (Fig. 13), the offset $q_{i,2} - q_{i,1}$ depends on $s_i$ as
$$(q_{i,2} - q_{i,1}) \Phi_{i,1,r} = 2r_c (0, s_i \cdot \sin \alpha_i, \cos \alpha_i).$$
Coupling with Eq. (40), $q_{i,2}$ can be formulated as
$$q_{i,2} = q_{i,1} + 2r_c (0, s_i \cdot \sin \alpha_i, \cos \alpha_i) = (e_{i,1,x}, 2r_c \cdot s_i \cdot \sin \alpha_i, e_{i,1,z} + 2r_c \cdot \cos \alpha_i).$$
Combining Eq. (41) and Eq. (42), $D_i$ defined in Eq. (35) can be formulated in $\Phi_{i,1,r}$ as
$$D_i \Phi_{i,1,r} = q_{i,2} = q_i + \Phi_{i,1,r} = (e_{i,1,x} - e_{i,2,x}, 2r_c \cdot s_i \cdot \sin \alpha_i, e_{i,1,z} - e_{i,2,z} + 2r_c \cdot \cos \alpha_i).$$
According to Eq. (38), Eq. (12) can be obtained from
$$D_i = D_i^{\Gamma_i} = R_{i,q}^{-1} D_i^{\Gamma_i,r} = R_{i,q}^{-1} D_i^{\Phi_{i,1,r}}.$$ On the other hand, Eq. (33) gives $\|D_i^{\Gamma_i}\| = \|D_i^{\Phi_{i,1,r}}\|$, i.e.
$$\|T_{i,2} - T_{i,1}\|^2 = (e_{i,1,x} - e_{i,2,x})^2 + (e_{i,1,z} - e_{i,2,z})^2 + 4r_c^2 + 4r_c (e_{i,1,z} - e_{i,2,z}) \cos \alpha_i.$$ According to Eq. (2), the sensor plane does not move parallel to $\pi_c$, i.e., $e_{i,1,z} \neq e_{i,2,z}$. Therefore,
$$\cos \alpha_i = \frac{\|T_{i,2} - T_{i,1}\|^2 - (e_{i,1,x} - e_{i,2,x})^2 - (e_{i,1,z} - e_{i,2,z})^2 - 4r_c^2}{4r_c (e_{i,1,z} - e_{i,2,z})},$$
which gives Eq. (9).

H. Algorithm design

Robot-sensor calibration is to derive the desired $R_{sensor}$ and $T_{sensor}$ from all sensing data. Eq. (36) gives
$$R_{sensor} D_i^{\Gamma_i} = R_{i,c}^{-1} D_i^{\Gamma_i}, i = 1, 2, \cdots, n,$$
which can be reformulated into
$$R_{sensor} = \left[ D_i^{\Gamma_i}, \cdots, D_n^{\Gamma_i} \right] = \left[ R_{i,c}^{-1}, \cdots, R_{i,c}^{-1} \right].$$
Coupling with notations in Eq. (4), the above equations become
$$R_{sensor} X = Y.$$ The unknown $R_{sensor}$ in Eq. (45) can be solved using the singular value decomposition (SVD) (22), which gives the solution as Eq. (46).

On the other hand, according to Eq. (4), for $i = 1, \cdots, n - 1,$
$$q_i^b = R_{tcp,i+1} R_{sensor} q_i^{i+1} + R_{tcp,i+1} T_{sensor} + T_{tcp,i+1},$$
$$q_i^e = R_{tcp,i} R_{sensor} q_i^e + R_{tcp,i} T_{sensor} + T_{tcp,i}.$$ The substitution of the two equations provides
$$(R_{tcp,i+1} R_{sensor} q_i^{i+1} + R_{tcp,i+1} T_{sensor} + T_{tcp,i+1}) - (R_{tcp,i} R_{sensor} q_i^e + R_{tcp,i} T_{sensor} + T_{tcp,i}) = 0,$$
which gives
$$(R_{tcp,i} - R_{tcp,i+1}) T_{sensor} = R_{tcp,i+1} R_{sensor} q_i^{i+1} - R_{tcp,i} R_{sensor} q_i^e + T_{tcp,i+1} - T_{tcp,i}.$$
and the calibration accuracy is estimated as
\[
\max_{i=1}^{4} \| \hat{q}^b - q_i^b \| = 0.212 \text{mm},
\] (51)
which is jointly affected by the \( x \)-resolution of the sensor, which is 0.1 mm, the robotic arm’s positioning accuracy, and the ellipse fitting accuracy.

### B. Simulation and validation

A simulated robotic-laser scanning system is set up with a simulated calibration matrix \( \tilde{\Lambda} = \Lambda \) in Eq.(49). A virtual disk centered at \( q^b = q^b \) in Eq.(50) with orientation \( \hat{n}_{ik}^b = (0, 0, 1) \) is used for collecting simulation data. TCP data in Eq.(48) are adopted for the simulation. The simulated chord data \( \bar{a}_{i,j} \) and \( \bar{b}_{i,j} \) are collected as the intersections of the virtual disk and the moving laser planes. Then, the calibration based on the simulated chord data derives the transformation matrix \( \tilde{\Lambda} \) and the disk center \( q^b \). Simulation results show that
\[
\tilde{\Lambda} = \Lambda, \quad \hat{q}^b = q^b,
\]
which validates the proposed calibration algorithm.

The above simulation and validation do not take into account the noise in the chord data. In fact, the real chord data \( a_{i,j} \) and \( b_{i,j} \) in Eq.(49) may contain noise introduced by the accuracy of the robotic arm, the \( x \)-resolution of the sensor, the accuracy of the chord segmentation, and so on. Assume that such noise is the Gaussian noise \( G(\sigma) \) with mean value 0 and standard deviation \( \sigma \). A set of noise chord data \( \{q_{i,j,k}^b\} \) is derived as
\[
\hat{q}^b = \frac{1}{\sqrt{\sigma^2 + \bar{a}_{i,j}^2}} \bigg[ (\bar{a}_{i,j} - \bar{b}_{i,j}) \bigg] + \sigma (\bar{a}_{i,j} - \bar{b}_{i,j}),
\] (52)
where \( i = 1, 2, 3, 4, j = 1, 2, \ldots, m \), and \( \bar{a}_{i,j,k} \) are Gaussian noise \( G(\sigma) \).

For each standard deviation \( \sigma \in \{0.1, 0.2, \ldots, 0.9\} \), 500 sets of noise chord data \( \{\bar{a}_{i,j,k}, \bar{b}_{i,j,k}\}_{k=1}^{500} \) are simulated according to Eq.(52). The calibration based on the \( i \)-th set \( \{\bar{a}_{i,j,k}, \bar{b}_{i,j,k}\}_{k=1}^{500} \) gives a transformation matrix, a disk center \( \hat{q}^b_i \), and a calibration accuracy \( \varepsilon_i = \| q^b_i - \hat{q}^b_i \| \) (the error in reconstructing the disk center). The standard deviation of \( \{\varepsilon_i\}_{k=1}^{500} \) is denoted as \( \varepsilon \), which indicates the calibration accuracy with respect to \( \sigma \) and \( m \).

Fig.15 shows the trend of the calibration accuracy \( \varepsilon \) with respect to \( \sigma \) and \( m \). In general, with \( m \) fixed, \( \varepsilon \) increases as \( \sigma \) increases. For example, with \( m = 100 \), the \( \varepsilon \) increases from 0.1 to 0.9, \( \varepsilon \) increases from 0.052 to 0.449. On the other hand, with \( \sigma \) fixed, bigger \( m \) gives better accuracy. In this simulation, when \( m \) increases from 10 to 200 while \( \sigma \) fixed at 0.4, the accuracy \( \varepsilon \) increases from 0.591 to 0.135 (the forth curve from the bottom in Fig[15]). According to curves in Fig[15], bigger \( m \) provides better accuracy. For the experimental setup in Fig[14(a), we adopt \( m = 100 \).

![Fig. 15. Disk-based calibration accuracy with respect to \( \sigma \) and \( m \) (the horizontal coordinate is the number of chords \( m \), the vertical coordinate is the calibration accuracy \( \varepsilon \) in mm, and curves in different colors correspond to different noise level \( \sigma \) in mm).](image1)

![Fig. 16. Reconstructed disk centers in repeatability test (the white dots are disk centers from different tests; the red dot is the average of the 32 white dots; the size of the bounding box is 0.071mm x 0.131mm x 0.287mm).](image2)

### C. Repeatability

To analyze the repeatability of the proposed calibration algorithm, with position and orientation of the disk fixed, calibration data is collected based on the same set of TCP data in Eq.(48). The data collection is repeated for 32 times providing 32 calibration results. The disk centers \( \{q_{i,j}^b\} \) are used to estimate the repeatability (the white dots in Fig[16]). For \( \{q_{i,j}^b\} \) are used to estimate the repeatability (the white dots in Fig[16]), the mean is denoted as (the red dot in Fig[16])
\[
\hat{q} = \frac{1}{32} \sum_{i=1}^{32} q_{i,j}^b,
\]
and the standard deviation is
\[
\frac{1}{32} \sum_{i=1}^{32} \| q_{i,j}^b - \hat{q} \| / 32 = 0.071.
\]
Thus, the repeatability of the proposed algorithm based on the same set of TCP data is 0.071mm.

### D. Comparisons

1) Different methods: The proposed method is different from the methods presented in [11]–[15] with different calibration targets and different calibration procedures (Table II). Balls had been used for the calibration between a CMM.
Multiple conjugate pairs with augmented objective function formulate a constrained least-squares optimization problem. The calibration target to be freely placed in the system like the balls. During calibration, $R_{\text{sensor}}$ and $T_{\text{sensor}}$ are also solved separately with at least four conjugate pairs in total. Different from previous methods [11]–[15], our method collects more data $\{T_{\text{tcp},i},R_{\text{tcp},i},q_i^b,q_i^q\}$ from one conjugate pair. First, data $\{T_{\text{tcp},i},R_{\text{tcp},i},D_{i}^b,D_{i}^q\}$ are used to formulate Eq. (45) to derive $R_{\text{sensor}}$. Then, Eq. (47) is formulated using $R_{\text{sensor}}$ and data $\{T_{\text{tcp},i},R_{\text{tcp},i},q_i^b\}$ to obtain $T_{\text{sensor}}$. Sensing data are usually collected manually. Users are required to try different $T_{\text{tcp}}$ and $R_{\text{tcp}}$ to locate different conjugate pairs. Smaller number of conjugate pairs make the sensing data collection easier. The ball-based calibration methods [11]–[13] need to collect two sets of sensing data, which needs at least 7 conjugate pairs in total. The first four conjugate pairs used for solving rotation portion required at least four $T_{\text{tcp}}$ with the same $R_{\text{tcp}}$. In the second set of data for solving the translation portion, two $T_{\text{tcp}}$ with the same $R_{\text{tcp}}$ are required for each conjugate pair. Compared with the ball-based calibration, the proposed algorithm only requires one set of the sensing data with at least 4 conjugate pairs in total.

2) Similar calibration accuracy: The ball in Fig. 14(b) is used to illustrate that the proposed calibration method can achieve similar accuracy as the ball-based method does. In this comparison, the disk-based calibration and ball-based calibration are based on four and eight conjugate pairs, respectively. Table [13] lists the eight sensing data $\{T_{\text{tcp},i},R_{\text{tcp},i},q_i^b\}_{i=1}^8$ derived for the ball-based calibration, where the first four sensing data with the same orientation are used for solving $R_{\text{sensor}}$ and the others with different orientations are used for deriving $T_{\text{sensor}}$. The ball-based calibration gives

$$
T = \begin{bmatrix}
0.99983 & -0.0115 & -0.01411 & 6.30066 \\
-0.01129 & -0.99983 & 0.01485 & 54.38647 \\
-0.01428 & -0.01468 & -0.99979 & 364.14357 \\
0 & 0 & 0 & 1
\end{bmatrix}.
$$

The deviation between Eq. (49) and Eq. (53) is

$$
\Delta^{-1}T = \begin{bmatrix}
1.0000 & -0.0059 & -0.0070 & 0.2751 \\
0.0059 & 0.9999 & -0.0119 & -0.2055 \\
0.0071 & 0.0119 & 0.9999 & 0.1104 \\
0 & 0 & 0 & 1
\end{bmatrix}.
$$

### TABLE I. THE SENSING DATA FOR DISK-BASED CALIBRATION.

<table>
<thead>
<tr>
<th>$i$</th>
<th>$T_{\text{tcp},i}$</th>
<th>$R_{\text{tcp},i}$</th>
<th>$q_i^b$</th>
<th>$D_{i}^b$</th>
<th>$D_{i}^q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-222.966, -295.994, 360.786</td>
<td>(3.057, -0.103, -2.008)</td>
<td>(5.947, 0.597)</td>
<td>(0.042, 0.336, -0.941)</td>
<td>(0.181, -0.246, -0.952)</td>
</tr>
<tr>
<td>2</td>
<td>-232.113, -298.165, 344.196</td>
<td>(3.115, -0.2, -2.269)</td>
<td>(-12.315, 0.247, 0.041)</td>
<td>(-0.146, -0.424, -0.694)</td>
<td>(0.287, -0.294, -0.912)</td>
</tr>
<tr>
<td>3</td>
<td>-374.932, -347.138, 338.836</td>
<td>(-2.981, 0.256, -2.491)</td>
<td>(9.328, 0.34, 0.277)</td>
<td>(-0.220, 0.589, -0.777)</td>
<td>(0.391, -0.593, -0.704)</td>
</tr>
<tr>
<td>4</td>
<td>-377.572, -373.611, 352.241</td>
<td>(3.092, 0.208, 3.11)</td>
<td>(9.675, 0.15, 0.973)</td>
<td>(-0.024, 0.674, -0.738)</td>
<td>(0.164, -0.631, -0.758)</td>
</tr>
</tbody>
</table>

### TABLE II. COMPARISON OF DIFFERENT CALIBRATION METHODS.

<table>
<thead>
<tr>
<th>Calibration target</th>
<th>Number of conjugate pairs</th>
<th>Method</th>
<th>Required data for each conjugate pair</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tetra-ball</td>
<td>More than 3</td>
<td>Constrained least-squares optimization</td>
<td>$q_i^b$ and $q_i^q$</td>
</tr>
<tr>
<td>Ball</td>
<td>At least 7</td>
<td>At least 4 data for $R_{\text{sensor}}$, and at least 3 data for $T_{\text{sensor}}$</td>
<td>$R_{\text{sensor}}$ and $T_{\text{sensor}}$</td>
</tr>
<tr>
<td>Ours Planar disk</td>
<td>At least 4</td>
<td>${T_{\text{tcp},i},R_{\text{tcp},i},D_{i}^b,D_{i}^q}$</td>
<td>$T_{\text{sensor}}$</td>
</tr>
</tbody>
</table>

The table shows that our method collects more data than previous methods [11]–[15], which is more accurate and efficient for disk-based calibration compared to the ball-based calibration.
Following Section III-D, the ball center from each orientation with Euler angles

\[(0.0119, -0.0071, 0.0059)\].

(55)

The ball center is estimated as

\[\mu = \frac{\sum_{i=0}^{8} \mu_i}{4},\]

and the calibration accuracy is estimated as

\[\max_{i \neq j} \|\mu_i - \mu_j\| = 0.208\text{mm}.

(56)

Eq. (51) and Eq. (56) are similar. However, they are estimated according to different fixed points. In (53), calibration accuracy is evaluated as the standard deviation of the ball centers reconstructed from different scans. We follow the same way to compare the two calibration results Eq. (49) and Eq. (53). The scanning system scans the ball in Fig. 14(b) with the robotic arm in different orientations. In each scan, 50 profiles are sampled with the robotic arm in a fixed orientation. For \(i = 1, 2, \cdots, 8\), applying a transformation matrix (Eq. (49) or Eq. (53)) to the \(i\)-th scan gives one point cloud \(C_i\). On average, there are 45000 points in each point cloud. The ninth point cloud is defined as \(C_9 = \bigcup_{i=1}^{8} C_i\). A ball is fitted to the point cloud \(C_i\) to derive the ball center \(\mu_i\), \(i = 1, 2, \cdots, 9\) (Table IV). The standard deviation of the ball centers is calculated as

\[\sum_{i=1}^{8} \|\mu_i - \mu_9\| / 8.

The standard deviations for using Eq. (49) and Eq. (53) are 0.342 mm and 0.355 mm, respectively. Thus, we conclude that the disk-based calibration can provide a similar calibration result as the ball-based method does.

3) Different noise-tolerance: The disk-based and ball-based calibrations are affected by different sources of noise. Disk-based calibration relies on the two tips on each chord, thus the \(x\)-resolution of the sensor affect the accuracy. One the other hand, the ball-based calibration depends on the arc data, which are affected by the \(z\)-accuracy of the sensor and the surface condition of the ball. The fitting algorithms adopted during the calibration may also introduce errors as well. Thus, instead of comparing their accuracies under the same noise level, we study their capability to tolerate a given noise. As presented in the last column of Table III, increasing \(m\) is one way to tolerate the noise in disk-based calibration.

As presented in Fig. 15, increasing \(m\) one way to tolerate the noise in disk-based calibration.
VI. CONCLUSION

For calibrating profile sensors to robotic arms, all existing methods require 3D calibration balls. Previously, 2D disks are not used for robot-sensor calibration because it is difficult to estimate its orientation during the calibration. In this paper, a new calibration algorithm is proposed to perform robot-sensor calibration based on a 2D disk. The novel feature is that, in the calibration procedures, the 2D disk can be placed in arbitrary orientation and this makes the placement of the disk as easy as that of a 3D ball. To acquire the sensing data for our calibration algorithm, users can project the line laser onto the 2D disk without passing any specific point. The proposed algorithm can derive the angular relationship between the laser and the 2D disk. The 2D disk center can be obtained in the calibration. A new two-step calibration is proposed based on a single set of sensing data. This can reduce not only the manual work in data collection but also the error propagation. Minimal manual work in data collection for the calibration is desired for better accuracy and faster processing. Simulations are performed to validate the proposed algorithm and analyze the effect of noise. Experiments show that the proposed algorithm can achieve similar accuracy as the ball-based method does.

ACKNOWLEDGEMENTS

This work is partially supported by SERC grant (1225100006) from A*STAR of Singapore.

REFERENCES