Active Robot Calibration Algorithm

Yu Sun and John M. Hollerbach

Abstract— **This paper presents a new updating algorithm to reduce the complexity of computing an observability index for kinematic calibration of robots. An active calibration algorithm is developed to include an updating algorithm in the pose selection process. Simulations on a 6-DOF PUMA robot with 27 unknown parameters shows that the proposed algorithm performs more than 50,000 times better than exhaustive search based on randomly generated designs.**

I. Introduction

In robot calibration, pose selection is an important topic since different poses and different combinations of poses contribute to the calibration very much differently. Past research in kinematic calibration has adapted optimal design algorithms from the experimental design literature for pose selection [7], [18], [20], [13], [14], [6], [5]. The algorithms are often iterative algorithms using exchange schemes. Fedorov and Dubova (1968) developed the first general algorithm (translated to English in [11]). It allows the variance of the observation error to be a function of the design point. Mitchell [16] developed an algorithm called DETMAX that allows the number of design points to increase or decrease for a better search and to escape from local optima. Since then, many improvements have been made with respect to computational time and space [12], [3], [9], [22], [1], [21], [8].

Current robot calibration algorithms usually search for an optimal pose set with an adapted DETMAX in that the optimal criterion is replaced with an observability index. The computation of observability index is the most frequent computation in such algorithms. For each iteration, the observability index is computed for every new possible design. Usually there are hundreds of thousands of candidates. The computational complexity of the observability index directly decides the capability of the search for the optimal pose. If the observability index can be computed more efficiently, more candidate pose sets can be included in the search and the search will more likely reach the optimal pose set.

The existing observability indexes are all eigenvalue-based (or singular-value-based). In the DETMAX, after adding a pose or exchanging a pose, a new Jacobian matrix is formed and the new eigenvalues need to be computed. The computation complexity of eigenvalues is proportional to the size of the design matrix. When the number of the unknown parameters is large, computing its eigenvalues becomes a big burden.

This paper replaces the eigenvalue-based observability index with a determinant and provide formulas to update it for adding and removing a pose. The computation complexity is significantly reduced. An exchange-add-exchange algorithm is designed to select poses for optimal robot calibration with such an alternative criterion.

The new robot calibration pose selection routine is demonstrated with a 6-DOF PUMA 560 simulation. 27 unknown kinematic parameters are calibrated with the goal of minimizing the variance of parameters. The results show that the new active robot algorithm performs more than 50,000 times better than exhaustive search on randomly generated designs.

Y. Sun and J. M. Hollerbach are with School of Computing of University of Utah. {ysun,jmh}@cs.utah.edu

II. Existing Robot Calibration Algorithms

A robot kinematic model can be calibrated with a set of poses. The measurements of the end-effector and the joint encoder readings at selected poses are collected. We define a pose set as a *design*, namely $D = \{p_1, p_2, ..., p_n\}$. The Jacobian matrices $\mathbf{X}^i$ of poses p_i's are computed. For pose i, there are 3 or 6 equations depending on the measurements of the end-effector. For simplicity, we only consider the case where each pose has 3 measurements, namely x, y, and z positions of the end-effector. There are 3 rows in $\mathbf{X}^i$ and denoted as $\mathbf{x}_x$, $\mathbf{x}_y$, and $\mathbf{x}_z$. We also assume that the distribution of the measurement errors are independent and identically distributed.

Assuming each design has N unique poses, if there are M candidate poses, there are $(MN) = \frac{M!}{N!(M-N)!}$ unique designs. For a robot with s-DOF, if each joint angle has k samples, there are $((k^s)N) = \frac{k^s!}{N!(k^s-N)!}$ possible designs. For example, a 6-DOF Puma robot with 10 samples of each joint angle, there are 10^6 candidate poses. If the unknown parameter number $p = 30$, it needs at least 30 equations to just possibly avoid singularity. To achieve an optimal design, it needs at least $p * (p-1)/2$ to fully determine the design matrix. For $p = 30$, it needs $p * (p-1)/2/3 = 145$ poses. Then there are $\frac{10^6!}{145!(10^6-145)!}$ possible designs. To search an optimal design from such a big design set is a factorial complexity problem. It is infeasible with current computation power.

The same problem has been faced in the general experimental design literature, where many design algorithms have been developed. The one mostly used in the robot calibration literature is DETMAX, developed in [16]. DETMAX starts with a randomly selected initial N-pose set and exchanges a pose at each iteration.

1) Begin with a randomly selected N-pose design $\boldsymbol{\xi}_0(n)$.
2) Find a pose p from the remaining candidate pose set such that the new design $\boldsymbol{\xi}_i$ after including the pose p has the maximum increase.

3) Find a pose p^* in the current design pose set such that the new design $\boldsymbol{\xi}_i^*$ after removing the pose p^* has the minimum decrease.
4) Repeat steps 2 and 3 until no increase in the value of the observability indices is obtained by an exchange.

The main idea of this algorithm is that the initial design set is improved by adding a new pose that maximally increases the criterion and removing a pose in the design set that minimally decreases the criterion. Each time, an excursion improves the currently best pose design. The algorithm stops when the added pose is instantly removed.

The algorithm has numerous variations. [4], [18] used the DETMAX algorithm in optimal experimental design to select robot poses. [6] used a GA algorithm to generate new poses. For each search step, an eigenvalue-based criterion needs to be computed. The eigenvalues can either be computed from the design matrix $\mathbf{X}$ by singular value decomposition (SVD) or from the covariance matrix $\mathbf{X}'\mathbf{X}$ by eigenvalue decomposition. In either case, the computation is expensive. For example, in [17] the computation complexity of eigenvalues was proved to be bounded by $O(n^3+(n\log^2 n)\log b)$ where n is the number of unknown parameters and b indicates the error is bounded by 2^{-b}. Also the complexity of matrix multiplication is $O(m^{2.376})$ [2] where m is the number of measurements. When the number of unknown parameters and the number of measurements are large, the computation of the eigenvalues is very expensive.

For both the optimal experimental design literature and the robot calibration literature, the local optimization problem has been recognized. The problem is tackled by randomly selecting initial runs many times.

III. Alternative Criterion

As proved in [19], if the goal of the calibration is to achieve minimum variance of estimated parameters, the best observability index is

$$OI = \frac{\sqrt[L]{\sigma_1\ \sigma_2\ \ldots\ \sigma_L}}{L} \tag{1}$$

where σ_i^2s are the eigenvalues of the covariance matrix $\mathbf{X}'\mathbf{X}$, L is the number of non-zero eigenvalues. Its inverse represents the volume of the confidence hyper-ellipsoid for the parameters. For parameter estimation, L is equal to the number of the unknown parameters. It is a constant. So, the real criterion is the multiplication of all the non-zero eigenvalues, which is the determinant of the covariance matrix, denoted as OI_D.

When adding a new row $\mathbf{x}_i'$ into the design matrix $\mathbf{X}_0$, the new design matrix becomes

$$\mathbf{X}_1 = \begin{bmatrix} \mathbf{X}_0 \\ \mathbf{x}_i' \end{bmatrix}. \tag{2}$$

The covariance matrix is

$$\begin{aligned} \mathbf{M}_1 &= \mathbf{X}_1{}'\mathbf{X}_1 \\ &= \begin{bmatrix} \mathbf{X}_0{}' & \mathbf{x}_i \end{bmatrix} \begin{bmatrix} \mathbf{X}_0 \\ \mathbf{x}_i' \end{bmatrix} \\ &= \mathbf{X}_0{}'\mathbf{X}_0 + \mathbf{x}_i\mathbf{x}_i' \\ &= \mathbf{M}_0 + \mathbf{x}_i\mathbf{x}_i' \end{aligned} \tag{3}$$

As we know, the determinant of the new covariance matrix can be updated with

$$OI_D = |\mathbf{M}_0 + \mathbf{x}_i\mathbf{x}_i'| \tag{4}$$

$$= |\mathbf{M_0}|(1+\mathbf{x}_i'\mathbf{M_0}^{-1}\mathbf{x}_i). \tag{5}$$

The determinant updating method has been used in the experimental design literature since the 1970's [10], [11]. [15] also provides a procedure to update $\mathbf{M}$ and $\mathbf{M}^{-1}$.

The inverse of the covariance matrix $\mathbf{M}^{-1}$ can be updated as

$$(\mathbf{M}+\mathbf{x}_i\mathbf{x}_i')^{-1} = \mathbf{M}^{-1} - \frac{\mathbf{M}^{-1}\mathbf{x}_i\mathbf{x}_i'\mathbf{M}^{-1}}{1+\mathbf{x}_i'\mathbf{M}^{-1}\mathbf{x}_i}. \tag{6}$$

When removing a row $\mathbf{x}_i$ from the design matrix $\mathbf{X}_0$, the relation between the new design matrix $\mathbf{X}_1$ and the old design matrix $\mathbf{X}_0$ is

$$\mathbf{X}_0 = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{x}_i' \end{bmatrix}. \tag{7}$$

Similar to Equation 8, the new covariance matrix is

$$\mathbf{M}_1 = \mathbf{M}_0 - \mathbf{x}_i\mathbf{x}_i' \tag{8}$$

The determinant of the new covariance matrix after removing a row can be updated with

$$OI_D = |\mathbf{M}_0 - \mathbf{x}_i\mathbf{x}_i'| \tag{9}$$

$$= |\mathbf{M_0}|(1-\mathbf{x}_i'\mathbf{M_0}^{-1}\mathbf{x}_i). \tag{10}$$

The inverse of the covariance matrix for removing a row is

$$(\mathbf{M}-\mathbf{x}\mathbf{x}')^{-1} = \mathbf{M}^{-1} + \frac{\mathbf{M}^{-1}\mathbf{x}\mathbf{x}'\mathbf{M}^{-1}}{1-\mathbf{x}'\mathbf{M}^{-1}\mathbf{x}}. \tag{11}$$

With such formulas, the matrix inverse and the determinant can be updated with vector-matrix multiplications. Its complexity is only $O(n^2)$ where n is the number of unknown-parameters. Its complexity is independent of the number of measurements.

For robot calibration, since each Jacobian matrix associated with a pose has more than one row, in each iteration more than one row is added or exchanged simultaneously. For adding two rows, the new Jacobian matrix is

$$\mathbf{X}_1 = \begin{bmatrix} \mathbf{X}_0 \\ \mathbf{x}_x' \\ \mathbf{x}_y' \end{bmatrix}. \tag{12}$$

The covariance matrix is

$$\mathbf{M}_1 = \mathbf{X}_1{}'\mathbf{X}_1 \tag{13}$$

$$= \begin{bmatrix} \mathbf{X}_0{}' & \mathbf{x}_x & \mathbf{x}_y \end{bmatrix} \begin{bmatrix} \mathbf{X}_0 \\ \mathbf{x}_x' \\ \mathbf{x}_y' \end{bmatrix} \tag{14}$$

$$= \mathbf{X}_0{}'\mathbf{X}_0 + \mathbf{x}_x\mathbf{x}_x' + \mathbf{x}_y\mathbf{x}_y' \tag{15}$$

$$= \mathbf{M}_0 + \mathbf{x}_x\mathbf{x}_x' + \mathbf{x}_y\mathbf{x}_y'. \tag{16}$$

$$
\begin{aligned}
|\mathbf{M}_1| &= |\mathbf{M}_0 + \mathbf{x}_x\mathbf{x}_x' + \mathbf{x}_y\mathbf{x}_y'| \\
&= |\mathbf{M}_x + \mathbf{x}_y\mathbf{x}_y'| \\
&= |\mathbf{M}_\mathbf{x}|(1 + \mathbf{x}_y'\mathbf{M}_x^{-1}\mathbf{x}_y) \\
&= |\mathbf{M_0}|(1 + \mathbf{x}_x'\mathbf{M}_0^{-1}\mathbf{x}_x)(1 + \mathbf{x}_y'\mathbf{M}_x^{-1}\mathbf{x}_y) \quad (17) \\
&= |\mathbf{M_0}|(1 + \Delta(\mathbf{x}_x, \mathbf{x}_y)). \quad (18)
\end{aligned}
$$

where $\mathbf{M}_x = \mathbf{M_0} + \mathbf{x}_x\mathbf{x}_x'$. So,

$$
\begin{aligned}
\Delta(\mathbf{x}_x, \mathbf{x}_y) &= \mathbf{x}_x'\mathbf{M_0}^{-1}\mathbf{x}_x + \mathbf{x}_y'\mathbf{M_0}^{-1}\mathbf{x}_y(1 + \mathbf{x}_x'\mathbf{M_0}^{-1}\mathbf{x}_x) \\
&\quad -(\mathbf{x}_x'\mathbf{M_0}^{-1}\mathbf{x}_y)^2. \quad (19)
\end{aligned}
$$

For cases with more than 2 rows in a Jacobian matrix, the formulas gets more complex. Instead of using the explicit expression as in Equation (19), we use the cascade formula 17. For a Jacobian matrix with t new rows , the cascade formula can be written as

$$
\begin{aligned}
|\mathbf{M}_1| &= |\mathbf{M_0}|(1 + \mathbf{x}_1'\mathbf{M}_0^{-1}\mathbf{x}_1)(1 + \mathbf{x}_2'\mathbf{M}_{0,1}^{-1}\mathbf{x}_{0,2}) \ldots \\
&\quad (1 + \mathbf{x}_t'\mathbf{M}_{0,t-1}^{-1}\mathbf{x}_t) \quad (20)
\end{aligned}
$$

where $\mathbf{x}_i$ is the ith row in the Jacobian matrix and $\mathbf{M}_{0,i}$ is the covariance matrix that is added with the first i rows. The inverse of the $\mathbf{M_i}$ for $i > 0$ can be computed with equation (6) sequentially.

To remove two rows from a Jacobian matrix, the updating formula for the determinant of the covariance matrix is

$$
\begin{aligned}
\Delta(\mathbf{x}_x, \mathbf{x}_y) &= -\mathbf{x}_x'\mathbf{M_0}^{-1}\mathbf{x}_x - \mathbf{x}_y'\mathbf{M_0}^{-1}\mathbf{x}_y(1 - \mathbf{x}_x'\mathbf{M_0}^{-1}\mathbf{x}_x) \\
&\quad -(\mathbf{x}_x'\mathbf{M_0}^{-1}\mathbf{x}_y)^2. \quad (21)
\end{aligned}
$$

To remove more than 2 rows, the formula is

$$
\begin{aligned}
|\mathbf{M}_1| &= |\mathbf{M_0}|(1 - \mathbf{x}_1'\mathbf{M}_0^{-1}\mathbf{x}_1)(1 - \mathbf{x}_2'\mathbf{M}_{0,1}^{-1}\mathbf{x}_{0,2}) \ldots \\
&\quad (1 - \mathbf{x}_t'\mathbf{M}_{0,t-1}^{-1}\mathbf{x}_t) \quad (22)
\end{aligned}
$$

The inverse of the $\mathbf{M_i}$ and $\mathbf{M_{i,j}}$for $i > 0$ can be computed with formula (11) sequentially.

IV. Active Calibration Algorithm

The calibration algorithm can be carried out to satisfy two conditions. It stops when either the maximum number of poses is reached, or some optimal metrics are met such as the determinant of the covariance matrix reaches a sufficient level.

There are two basic elements in the calibration algorithm, add-a-pose and remove-a-pose:

- Add-a-pose:
  For a design $\boldsymbol{\xi}_k$, to add a pose to the design is to find a new pose p_j from the pose pool $\boldsymbol{\Omega}$, such that

$$
O(+(p_j, \boldsymbol{\xi}_k(n))) = max(+(p, \boldsymbol{\xi}_k(n))), \quad \forall p \in \boldsymbol{\Omega} - \boldsymbol{\xi}_k) \quad (23)
$$

  $+(p, \boldsymbol{\xi})$ is the operation that adds a pose p into the design $\boldsymbol{\xi}_k$. It computes with Equation 19 if each pose has two rows in its Jacobian matrix or with Equation 20 if each pose has more than three rows in its Jacobian matrix. The $\mathbf{M}^{-1}$ is updated with Equation 6.
- remove-a-pose
  For a design $\boldsymbol{\xi}_k$, to remove a pose to design is to find a pose p^* in the current design $\boldsymbol{\xi}_k$, such that

$$
O(-(p_j, \boldsymbol{\xi}_k(n))) = max(-(p, \boldsymbol{\xi}_k(n))), \quad \forall p \in \boldsymbol{\xi}) \quad (24)
$$

  $-(p, \boldsymbol{\xi})$ is the operation that remove a pose p from the design $\boldsymbol{\xi}$. It computes with Equation 21 if each pose has two rows in its Jacobian matrix and with Equation 22 if each pose has more than three rows in its Jacobian matrix. The $\mathbf{M}^{-1}$ is updated with Equation 11.

Some poses are required to be included or can be measured without cost. They are included in the initial pose set and are not allowed to be removed. There are poses that are in the pose space but are not viewable from measurement sensors, such as laser or stereo cameras. Those poses can not be included for calibration. In our algorithm, a preprocess is applied to eliminate the unobservable poses.

The following is the algorithm of selecting optimal poses for robot calibration.

1) Initialize: A number N of poses are selected as candidate poses. The candidate pose set is called the pose pool $\boldsymbol{\Omega}$. n initial poses are randomly selected from the pose pool $\boldsymbol{\Omega}$. The initial Jacobian matrix, its covariance matrix $\mathbf{M_0}$, determinant of the covariance matrix $|\mathbf{M_0}|$ and its inverse $\mathbf{M_0}^{-1}$ are computed. The initial design is denoted as $\boldsymbol{\xi}_0$.
2) Exchange Step: A pose p is removed from the current design $\boldsymbol{\xi}_i$ with remove-a-pose element. The current OI_D of the design without such a pose is computed. A new pose q from the candidate pose pool is selected with add-a-pose to form a new design. The OI of the new design is computed. If the OI of the new design is larger than the OI_D of the current design, pose p is replaced with pose q to form a new design $\boldsymbol{\xi}_{i+1}$.
3) The step 2 stops either reaching a predefined iteration number or it does not increase the OI_D anymore.
4) Add step: To expand the design with more poses, a new pose in the candidate pose pool is selected with add-a-pose element.
5) Repeat step 4 until either the number of poses reaches a predefined maximum number n_1, or adding a new pose does not increase the OI_D anymore.
6) Exchange Step: It is the same as 2. The design after an exchange-step and an add-step goes through an exchange-step again.

The algorithm repeats several times to prevent getting stuck at a local optimum.

V. Puma robot calibration

The robot calibration design algorithm is demonstrated with a Puma 560 with simulation. The robot has 6-DOF and 27 parameters. The end-effector position (x, y, z) is

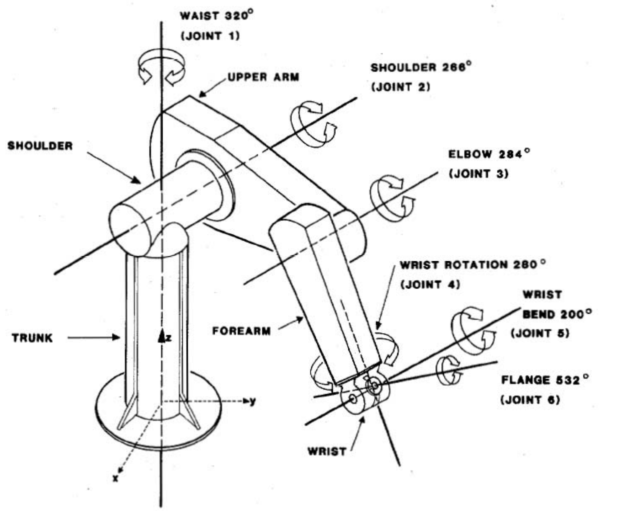

Fig. 1. Kinematic model of a PUMA 560 robot. It shows the degrees of joint rotation and ranges.

expressed in coordinate zero. We suppose it is tracked by an external measuring device with an 1 cm normally distribution error. The joint angle errors are assumed negligible.

A. Kinematic Model

The six revolute joints of the robot are shown in Figure 1. The ranges of each joints are indicated and listed as follows

$$\begin{aligned} -250^o \leq \theta_1 \leq 70^o \quad & -110^o \leq \theta_2 \leq 170^o \\ -133^o \leq \theta_3 \leq 133^o \quad & -100^o \leq \theta_4 \leq 100^o \\ -142^o \leq \theta_5 \leq 142^o \quad & -176^o \leq \theta_6 \leq 356^o \end{aligned} \tag{25}$$

The unknown parameters are DH parameters and the offsets and gains of each joint. The real values of the parameters are in Table I.

Frame	γ	α	a	d	β
a	$\frac{\pi}{2}$	$\frac{\pi}{100}$	20	0	$\frac{\pi}{100}$
0	0	-$\frac{\pi}{2}$	0	0.6858	10
1	0	$\frac{\pi}{100}$	0.4318	0	$\frac{\pi}{100}$
2	0	$\frac{\pi}{2}$	-0.0203	0.1491	10
3	0	-$\frac{\pi}{2}$	0	0.4331	10
4	0	$\frac{\pi}{2}$	0	0	10
5	0	0	0.2	0.1655	10
6	0	0	0	0	10

TABLE I
CALIBRATION PARAMETERS

Twenty-seven nominal parameters are randomly generated from the real parameters by adding normally distributed noise with variance of 0.1. Some parameters are not changed due to being constant. Table II displays the nominal parameters.

A mixed Hayati and DH parameterization is used for calibration since there are joint axes that are nearly parallel to their previous axis. For each pose, there are accurate measurements of 6 joints angles $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$ and noisy

Frame	γ	α	a	d	β
a	1.6360	0.1572	19.8820	0	-0.0123
0	-0.0367	-1.4524	0.0603	0.6713	10
1	-0.1455	-0.2622	0.3411	0	-0.1963
2	0.0128	1.6374	-0.1214	0.0074	10
3	0.0024	-1.3948	-0.1029	0.3002	10
4	0.1207	1.6151	-0.0468	-0.0747	10
5	0.0938	0	0.1322	0.3083	10
6	0	0	0	0	10

TABLE II
NOMINAL PARAMETERS

measurements of the location of the end-effector (x, y, z). The Jacobian matrix is calculated with the joints angles and the nominal parameters. Column-scaling is not used since the determinant-based observability index is invariant to column scaling [19].

B. Calibration Approaches

Usually the observability index increases with the pose number in a design. Since in this paper we are concentrating on pose selection, we define the the pose number in a design as 30.

To better understanding the pose selection, the pose space is latticed with 5 grids on each joint angle to generate 15,625 (5^6) poses for the candidate pool $\mathbf{\Omega}$. All candidate poses are indexed from 1 to 15,625. The index is assigned according to the combination of joint angles. For each joint, the angles are assigned to $0-5$. For example, a joint angle combination 030412 is indexed as $5^4 \times 3 + 5^2 \times 4 + 5^1 \times 1 + 2 = 1982$.

There are two practical approaches to select an optimal design. One is randomly composing a number of designs and pick the best. The other is actively selecting new designs according to a criteria. Our actively pose selection uses an exchange-add-exchange routine to update a randomly composed design. Our criteria is the determinant of the Jacobian matrix.

C. Random Design Selection

5,000 random designs are selected from the 15,625 poses. The observability index OI of each design is calculated and displayed in Figure 2. The highest OI of the 5,000 designs is 0.096284.

As described in Equation 1, the OI can be converted to the OI_D with

$$OI_D = \prod_{i=1}^{L} (\sigma_i)^2 = ((OI \times L)^L)^2.$$

With this formula, the determinant of the covariance matrix of the best design is calculated as 2.5414×10^{22}.

D. Exchange-Add-Exchange Active Pose Selection

The pose selection algorithm in Section IV randomly selected 12 poses as the initial design (Figure 3) and updated it with the exchange-a-pose, and then added 18 more poses with the add-a-pose. After the add-a-pose, the exchange-a-pose was called again. But no exchange was made since

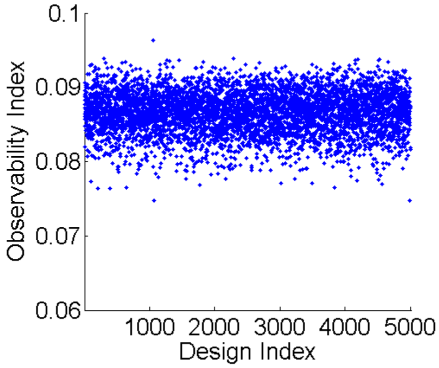

Fig. 2. The observability indices of 5000 randomly selected designs.

no better pose was found. Figure 4 shows the OI_D's of pose selection with our exchange-add-exchange routine from the same candidate pool. The OI_D of the final design is 1.4818×10^{27}.

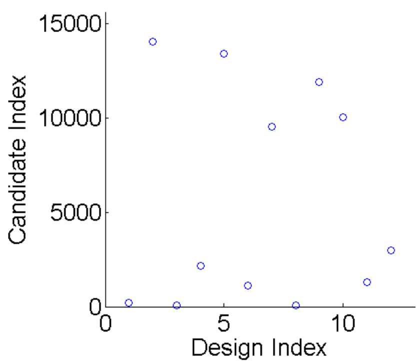

Fig. 3. The 12 initial poses.

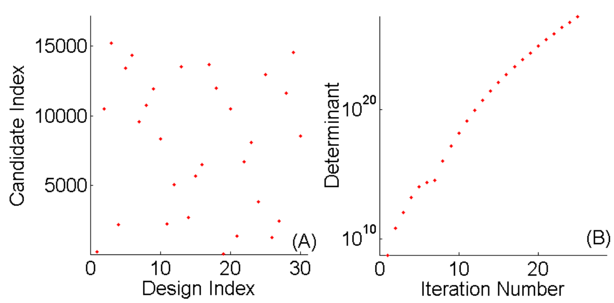

Fig. 4. (A) The selected poses; (B) The determinant of the variance of the covariance matrix for each pose selection iteration.

Starting with the same initial poses, the same exchange-add-exchange routine was carried again with original observability index OI as the criterion. The selected poses and the OI's for each iteration are displayed in Figure 5. The optimal design obtained with our exchange-add-exchange routine has OI at 0.11798. Comparing with Figure 4 (A), we can see that the poses selected according to the eigenvalue-based OI are the same as the poses selected according to the updating determinant-based observability index. It is indeed that alternative determinant-based observability index provides the same criterion as eigenvalue-based observability index. For each iteration, to compute the observability index, the computation complexity is $O(n^3+(n\log^2 n)\log b+m^{2.376})$, where n is the number of unknown parameters, m is the number of measurements, and b decides the accuracy of the eigenvalue computation. For this example, $n=27$, $m=900$, and $b=10$. The computation complexity for observability index is at the level of 10^7. To update the determinant in each iteration, the computation complexity is $O(n^2)$ that in this example is at the level of 10^3. For designs with large m, the advantage of using determinant-based criteria would be much more obvious.

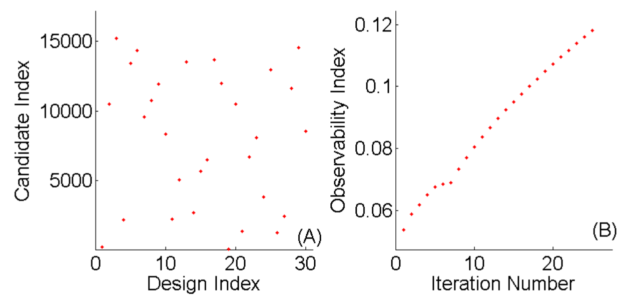

Fig. 5. (A) The selected poses for the optimal design based on the OI. (B) The observability index (OI) for each pose selection iteration.

The inverse of the determinant of the covariance matrix represents the volume of the confidence hyper-ellipsoid for the parameters. Figure 6 shows the inverse of the determinants in the pose selection. We can see that the overall determinant of covariance of the parameters decreases from level 10^{-1} to level 10^{-27}. It is 50,000 times smaller comparing with the pose set selected in Section V-C.

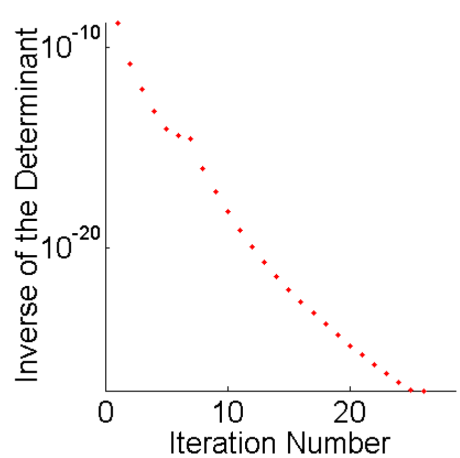

Fig. 6. The determinant of the variance of the estimated parameters.

E. Results

Based on the selected poses in Section V-D a robot calibration simulation is carried out. Table III lists the estimated parameters with the simulation. Table IV lists the residuals of the estimated parameters.

Frame	γ	α	a	d	β
a	1.570881	0.031672	20.001537	0	0.030290
0	0.003324	-1.568910	0.004117	0.685666	10
1	0.011386	0.030969	0.431692	0	0.027905
2	0.003694	1.566960	-0.028746	0.149672	10
3	-0.000672	-1.581108	-0.002910	0.433253	10
4	0.006226	1.577711	-0.001479	0.000302	10
5	-0.017604	0	0.199881	0.165315	10

TABLE III

THE ROBOT CALIBRATION RESULTS.

Frame	γ	α	a	d	β
a	0.000084	0.000256	0.001537	0	0.001126
0	0.003324	0.001886	0.004117	0.000134	0
1	0.011386	0.000447	-0.000108	0	0.003511
2	0.003694	0.003836	-0.008446	0.000572	0
3	0.000672	0.010311	-0.002910	0.000153	0
4	0.006226	0.006914	0.001479	0.000302	0
5	0.017604	0	0.000119	0.000185	0

TABLE IV

THE RESIDUALS OF THE ESTIMATED PARAMETERS AFTER ROBOT CALIBRATION.

VI. CONCLUSION

A determinant-based updating observability index is proposed to replace the eigenvalue-based observability index. The determinant-based observability index can be updated with vector-matrix multiplications. Its computation complexity is much lower than computing eigenvalues. With the new criterion, more candidate pose sets is afforded to be included in the optimal pose set search. An active robot calibration algorithm based on the determinant-based updating observability index is developed and demonstrated with a 6-DOF PUMA 560 robot calibration simulation. Based on two observability indexes, the same set of poses are selected. It proved that the two observability indexes are equivalent. The performance of our active robot calibration algorithm is compared with the exhaustive searching approach. In terms of the confidence hype-ellipsoid volume for the estimated parameters, our algorithm out-performs the exhaustive searching approach more than 50,000 times.

ACKNOWLEDGMENTS

This research is supported by NIH Grant 1R21EB004600-01A2.

REFERENCES

[1] A. Atkinson and A. Donev. *Optimum Experimental Designs*. Clarendon Press, Oxford, 1992. it is a listing of a fortran program BLKL blocking k l exchange algorithm.

[2] D. Bini and V. Y. Pan. *Polynomial and Matrix Computations, Volume1: Fundamental Algorithms*. Birkhauser, Boston, 1994.

[3] R. Cook and C. Nachtsheim. A comparison of algorithms for constructing exact d-optimal designs. *Technometrics*, 22(3):315–324, 1980.

[4] D. Daney. Optimal measurement configurations for gough platform calibration. In *Proc. IEEE International Conference on Robotics and Automation*, pages 147–152, 2002.

[5] D. Daney. Kinematic calibration of the gough platform. *Robotica*, 21(4):677–690, 2003.

[6] D. Daney, B. Madeline, and Y. Papegay. Choosing measurement poses for robot calibration with local convergence method and tabu search. *International Journal of Robotic Research*, 24(6):501–518, 2005.

[7] M. R. Driels and U. S. Pathre. Significance of observation strategy on the design of robot calibration experiments. *Journal of Robotic Systems*, 7(2):197–223, 1990.

[8] W. DuMouchel and B. Jones. A simple bayesian modification of d-optimal designs to reduce dependence on an assumed model. *Technometrics*, 36(1):37–47, 1994.

[9] J. M. E. and N. C. D-optimal designs on convex design spaces. *Technometrics*, pages 271–277, 1983.

[10] V. V. Fedorov. *Theory of optimal experiments*. Moscow State University, Moscow, preprint 7 lsm edition, 1969.

[11] V. V. Fedorov. *Theory of optimal experiments*. Academic Press, New York, 1972.

[12] Z. Galil and J. Kiefer. Time- and space-saving computer methods, related to Mitchell's DETMAX, for finding d-optimum designs. *Technometrics*, 22(3):301–313, 1980.

[13] W. Khalil, M. Gautier, and C. Enguehard. Identifiable parameters and optimum configurations for robots calibration. *Robotica*, 9:63–70, 1991.

[14] P. Kim and Y. Ding. Optimal design of fixture layout in multistation assembly processes. *IEEE Transactions on Automation Science and Engineering*, 1(2):133–145, 2004.

[15] A. J. Miller and N. K. Nguyen. Algorithm AS195: A fedorov exchange algorithm for d-optimal design. *Applied Statistics*, 43(4):669–678, 1994.

[16] T. J. Mitchell. An algorithm for the construction of d-optimal experimental designs. *Technometrics*, 16(2):203–210, 1974.

[17] V. Y. Pan and Z. Q. Chen. The complexity of the matrix eigenproblem. In *Proc. 31st Annual ACM Symposium on Theory of Computing*, pages 507–516. ACM Press, 1999.

[18] P. Renaud, N. Andreff, F. Marquet, and P. Martinet. Vision-based kinernatic calibration of a H4 parallel mechanism. In *Proc. IEEE International Conference on Robotics and Automation*, pages 1191–1196, 2003.

[19] Y. Sun and J. M. Hollerbach. Observability indices selection for robot calibration. In *IEEE International Conference Robotics and Automation*, 2008. under review.

[20] Y. Takeda, G. Shen, and H. Funabashi. A DBB-based kinematic calibration method for in-parallel actuated mechanisms using a fourier series. *Journal of Mechanical Design*, 126(5):856–865, 2004.

[21] W. J. Welch. Branch-and-bound search for experimental designs based on d optimality and other criteria. *Technometrics*, 24(1):41, 1982.

[22] W. J. Welch. Computer-aided design for experiments for response estimation. *Technometrics*, 26(3):217, 1984.