POSES SELECTION USING GENETIC ALGORITHM TO IMPROVE THE LOCAL POE KINEMATICS CALIBRATION

Ahmad Suryo Arifin[1], Marcelo Ang Jr.[2], Ma Chongyou[3], and Lim Chee Wang[4]

[1,2] Department of Mechanical Engineering, National University Singapore, Singapore,
e-mail: [1]g0800239@nus.edu.sg, [2]mpeangh@nus.edu.sg
[3,4]Singapore Institute of Manufacturing Technology, Singapore,
e-mail: [3]cwlim@SIMTech.a-star.edu.sg, [4]cyma@SIMTech.a-star.edu.sg

Received Date: March 25, 2011

Abstract

This paper investigates the use of genetic algorithm to optimize poses selection to improve kinematic calibration for manipulator. Genetic algorithm is used to determine the optimal poses while iterative least square algorithm is used to calibrate the kinematics model of the manipulator. Observability index are used to evaluate the optimality of the set of poses. The fitness function of genetic algorithm is chosen from the observability index. In addition, local POE (Product of Exponential) method is used to model the manipulator kinematics. The objective of this paper is to design an algorithm which optimizes the number of poses while improving the calibration performance. The experiments utilize 7-DOF Mitsubishi PA-10 manipulator as the platform and a LEICA laser tracker as the measurement tool. The experiment shows that genetic algorithm can optimize the number of poses and improve the calibration performance

Keyword: Genetic algorithm, Kinematics calibration, Local POE, Pose optimization

Introduction

Precise position control requires an accurate model of the manipulator. However, in practice, the accuracy of the kinematics model is reduced due to several external errors such as: manufacturing errors, link misalignment, and assembly errors at the manipulator. Kinematics calibration is presented as a solution to improve the accuracy of the manipulator. In general, kinematic calibration is influenced by the calibration algorithm, the modeling method, the quality of the poses and the measurement device.

There are several methods for kinematics calibration in literature. C.B. Wang, et al [1] made use of a forward calibration method. The forward calibration identifies the actual parameter of the manipulator based on the measurement at the workspace of the manipulator. Even though the method shows promising improvement, it becomes a problem if the inverse model of the manipulator is needed. A. Doria, et al [2] introduced inverse kinematics using B-splines and multivariate parametric approximating splines functions as tools to do calibration. I.M. Chen, et al [3] proposed a least square method to calibrate the manipulator. Local product of exponential is used to model the kinematics. Several works related to this method are also proposed in [4], [5], and [6]. Although the local POE calibration method provides significant improvement, this technique requires a lot of poses.

To optimize the poses for calibration, several techniques are introduced. H. Chernoff, [7] minimized the trace of non-zero singular values of the Jacobian matrices of the manipulator. This method is called A-Optimality. Wald [8] maximized the determinant of the non-zero singular values of the Jacobian matrices of the manipulator. This method is introduced as D-Optimality. Smith [9] proposed G-Optimality, it minimizes the maximum prediction variance of the non-zero singular values of the Jacobian matrices. Ehrenfeld [10] introduced E-Optimality. E-Optimality maximizes the minimum value of the non-zero singular value of the Jacobian matrices. Y. Sun, et al [11] compared the above observability indices. It is shown that G-Optimality and E-Optimality are the best observability indices to minimize the uncertainty at the end effector position of the manipulator, and D-Optimality is the best index to minimize the variance of the parameter.

The objective of this paper is to optimize the poses selection so that it can improve the kinematics calibration performance. Genetic algorithm is used to optimize poses selection while the local product of exponential (POE) is the tool for kinematics calibration. Genetic algorithm (GA) is chosen due to its ability to avoid the local minima, while local POE is selected due to its simplicity. The GA will obtain the optimal poses; this pose will be used as the pose for the local POE calibration. The results show that pose selection using genetic algorithm could provide the same result but with the less poses.

Kinematics Calibration Using Product of Exponential

The product of exponential (POE) formula is used to express the forward kinematics equation of the open chain manipulator. The local POE is determined by the configuration of the local initial frame, joint displacement and local twist of the robot. The local initial frame consists of the initial position and orientation information.

The forward kinematic $_{0,n}(q_1, q_2, \ldots, q_n)$ denoted as T to simplify the equation can be written as [3]:

$$T = f(T(0), s, q) \tag{1}$$

where, $T(0) = [T_{0,1}(0) \;\; T_{1,2}(0) \;\; \ldots \;\; T_{n-1,n}(0)]^T$ is defined as the matrix of initial poses of the manipulator, $s = [s_1 \;\; s_2 \;\; \ldots \;\; s_n]^T$ is defined as the twist and $q = [q_1 \;\; q_2 \;\; \ldots \;\; q_n]^T$ is defined as the joint displacement.

Linearizing the forward kinematic model yields,

$$(\delta T)T^{-1} = \left[\frac{\partial f}{\partial T(0)}\delta T(0) + \frac{\partial f}{\partial s}\delta s + \frac{\partial f}{\partial q}\delta q\right]T^{-1} \tag{2}$$

$(\delta T)T^{-1}$ is described as the pose (position and orientation) error at the end effectors with respect to the base frame as a summation of partial derivative of initial pose, twist and the joint displacement. On the other hand, this pose error can be defined as the deviation between computed pose from the kinematic model and measurement pose

Two assumptions are used to simplify the calibration strategy [5]:

- Kinematics error only exists in the initial poses on the local frames $T(0)$
- The joint displacements q and the joint twist $\hat{s}$ retain their nominal values for the entire calibration analysis.

Based on this assumption, the linearization of the forward kinematic model is simplified as,

$$(\delta T)T^{-1} = \left[\frac{\partial f}{\partial T(0)}\delta T(0)\right]T^{-1} \tag{3}$$

$(\delta T)T^{-1}$ describes the pose (position and orientation) deviation at the end effector with respect to the base frame as a partial derivative of the initial pose only. This is due to the simplification mentioned above. On the other hand, the pose error can be defined as the deviation between the computed kinematic model and measurement of the pose. Using the two assumptions mentioned above, the solution with the minimal deviation can be described as:

$$Min\left(\sum \left\|(\delta T)T^{-1} - \left[\frac{\partial f}{\partial T(0)}\delta T(0)\right]T^{-1}\right\|\right) \tag{4}$$

The formulation of matrix logarithms $(\delta T)T^{-1}$ could be written as [3]:

$$(\delta T)T^{-1} = \log(T^a_{0,n+1} T^{-1}_{0,n+1}) \quad (5)$$

The kinematic calibration can be linearized as:

$$l = Ko \quad (6)$$

where,

$$l = \log\left(T^a_{0,n+1} T^{-1}_{0,n+1}\right)^V$$

$$K = [Ad_{T0,1(0)} \;\; Ad_{T0,1} Ad_{T1,2(0)} \;\; \dots \;\; Ad_{T0,n} Ad_{Tn,n+1(0)}]$$

$$o = [\delta p_1 \;\; \delta p_2 \;\; \dots \;\; \delta p_{n+1}]^T$$

l is a 6×1 vector defined as the gross kinematic errors which can be calculated from the actual measurement and the nominal calculation of the pose, K is a $6 \times 6(\mathrm{n}+1)$ matrices which can be calculated from the nominal pose of the manipulator, o is the $6(\mathrm{n}+1) \times \mathrm{n}$ matrices which is the kinematic error in the manipulator. n is the number of joints in the manipulator. o can be computed using pseudo inverse formula, it can be described as:

$$o = (K^T K)^{-1} K^T l \quad (7)$$

From the calibration formula above, the kinematics error can be identified and the initial pose can be updated by:

$$T^c_{i-1,i}(0) = T_{i-1,i}(0) e^{\delta \widehat{p_i}} \quad (8)$$

The deviation between the actual and calibrated pose can be used to evaluate the result of this kinematic calibration method. It can be written as:

$$\Delta T = \frac{1}{mp} \sum_{i=1}^{mp} \left\| \log\left(T_i^{-1(a)} T_i^c\right)^V \right\| \quad (9)$$

Specifically, the deviation between position and orientation of the actual and calibrated pose can be written as:

$$\Delta R = \frac{1}{mp} \sum_{i=1}^{mp} \left\| \log\left(R_i^{-1(a)} R_i^c\right)^V \right\| \quad (10)$$

$$\Delta P = \frac{1}{mp} \sum_{i=1}^{mp} \| \log(P_i^a - P_i^c) \| \quad (11)$$

ΔR and ΔP are the orientation and position deviation respectively.

The performance of Local POE calibration method is depending on the number of poses. More number of poses could give better calibration accuracy. However, more number of poses could influence the computation time when performing the calibration. In order to reduce the number of poses for the local POE calibration, genetic algorithm (GA) is presented. The input of GA will be the random pose and the result of the GA will be the optimal pose. This optimal pose will be used as the pose for the local POE calibration. The advantages of this combination are the efficiency improvement of the computation time of the calibration due to the selection of the optimal poses to do the calibration. The explanation of how to obtain the optimal pose is discussed in the next chapter.

Genetic Algorithm to Optimize the Number of Poses for local POE Calibration

In the kinematics calibration, the end effector position and orientation are measured in many different poses. This is due to the number of unknown parameters of the manipulator. In addition, the configuration of the poses and the number of poses could affect the result of kinematics calibration. Therefore, to achieve a better calibration performance, a large number of set of poses should be chosen. However, large number of set of poses will influence the computation time and memory. In order to save computation time and memory, a small but well conditioned set of poses are introduced. One way to choose a good set of poses is by using genetic algorithm. Genetic Algorithm (GA) is an evolutionary technique for determining the optimum solution to a complex problem [12]. There are five steps to perform GA to optimize the calibration. Firstly, choose a number of initial individuals. The individual in this Genetic Algorithm are the sets of poses. Each individual consists of 20 poses. Secondly, determine the number of individuals in the populations. In order to get the best result, a large number of individuals in population are required. However, large number of individuals will influence the computation time and memory. Therefore, an optimal number of individuals in the population should be determined carefully. Thirdly, evaluate the fitness function of each individual in the population. The fitness function utilizes the observability indices as the criteria for the GA. The observability indices are determined by the singular value of the identification matrix.

Based on kinematic calibration formula [3], the identification matrices could be defined as:

$$K = [Ad_{T0,1(0)} \;\; Ad_{T0,1}Ad_{T1,2(0)} \;\; \dots \;\; Ad_{T0,n}Ad_{Tn,n+1(0)}] \tag{12}$$

Where Ad_T is defined as the adjoint representation of the transformation matrix T, n is the number of joints in the manipulator. Using Equation 1, the singular value decomposition can be written as:

$$K = U\Sigma V' \tag{13}$$

Where U and V' are orthonormal matrices, Σ can be described as:

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ \vdots & \dots & 0 & \sigma_m \\ \vdots & \dots & \ddots & 0 \\ 0 & \dots & \dots & 0 \end{bmatrix} \tag{14}$$

Where $\sigma_1, \sigma_2, \dots, \sigma_m$ is defined as the singular value of the matrices K.

There are several observability indices methods to optimize the poses calibration by utilizing the singular value of the K matrices. In this paper, D-Optimality, G-Optimality, E-Optimality, and A-Optimality are introduced as methods to optimize the set of poses.

D-optimality or observability index O_1 is the root of the product of singular values of the matrices K. It can be expressed as:

$$O_1 = \frac{(\sigma_1 \sigma_2 \dots \sigma_{m_{mm}})^{\frac{1}{m_{mm}}}}{\sqrt{m_{mm}}} \tag{15}$$

G-Optimality or observability index O_2 is the ratio between the minimum singular values of the matrices K and the maximum singular value. It can be written as:

$$O_2 = \frac{\sigma_{MIN}}{\sigma_{MAX}} \tag{16}$$

E-Optimality or observability index O_3 is the minimum value of singular values of the matrices K. It can be written as:

$$O_3 = \sigma_{MIN} \tag{17}$$

A-Optimality or observability index O_4 is the ratio between square of the minimum singular value and the maximum singular values of the matrices K. It can be written as:

$$O_4 = \frac{{\sigma_{MIN}}^2}{\sigma_{MAX}} \tag{18}$$

Evaluating the fitness function value at each of generation using Equation 15, 16, 17 or 18 will keep the individuals with the best fitness values and will replace the least fit individuals with these fitter individuals. Fitter individuals are generated by performing crossover and mutation operations. Mutation is introduced to maintain the diversity of the population. By crossover, a new population is formed through some combination of data from the selected best-fit individuals. The crossover method, in the present work, is shown in Figure 1.

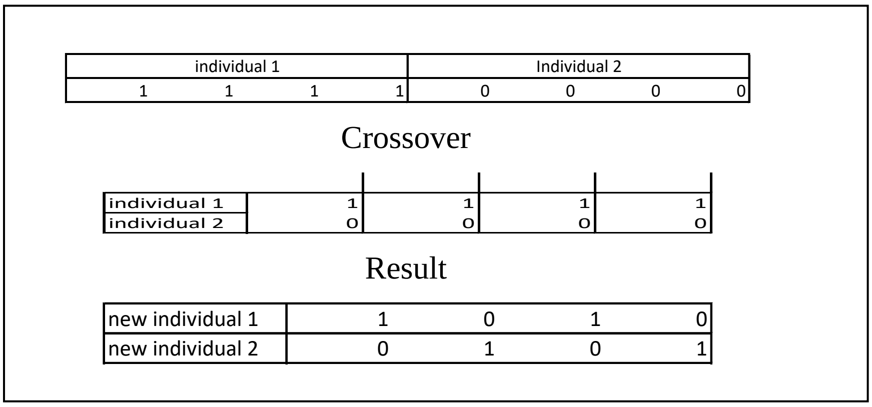

Figure 1. Crossover

Finally, the new population will replace the previous population for new generation. The result of genetic algorithm is an optimal set of poses. These optimal set of poses is used as the pose for the local POE calibration. These optimal poses selection could reduce the number of poses so that the calibration can be optimized. The full schematic of the calibration can be seen in Figure 2.

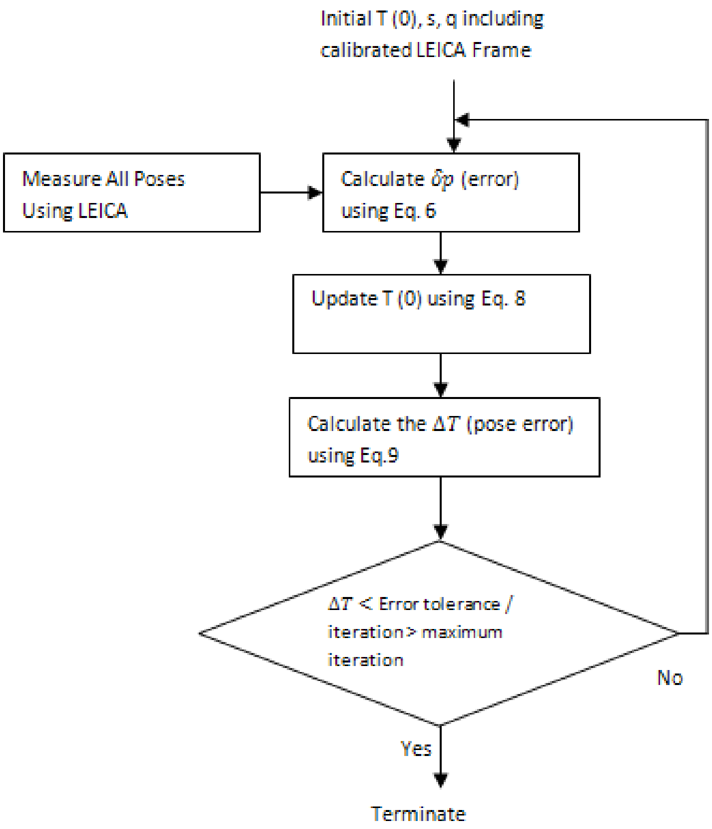

Figure 2. Algorithm flow chart

As can be seen from Figure 2, the actual pose can be measured using laser tracker. These poses are selected from the GA result. The gross kinematic error between actual and calculated pose can be calculated using Equation 6. Gross kinematic error can be projected to the pose error using pseudo inverse formula in Equation 7. The pose error is used to update the initial model of the manipulator using Eqn.8. The calibration will be terminated if the pose error satisfies the error tolerance, or when the algorithm reaches the maximum iteration. The error tolerance and the maximum iteration are defined by the user.

Experimental Results of Kinematics Calibration

In this research, the Mitsubishi PA-10 manipulator is calibrated with the proposed algorithm. The Mitsubishi PA-10 is a seven degree-of-freedom Industrial manipulator. To measure the position and the orientation of the end effector, we make use of the LEICA laser tracker and Tracker-Machine control sensor (T-Mac). LEICA laser tracker is a laser measurement device while T-Mac is the receiver of the laser from the laser tracker. The Laser tracker follows the position of T-Mac and sends the information of position and orientation of the T-Mac to the computer. The LEICA laser tracker is placed in front of the manipulator while a T-Mac is attached at the end effector of PA-10. The set up configuration of the experiment can be seen in Figure 3.

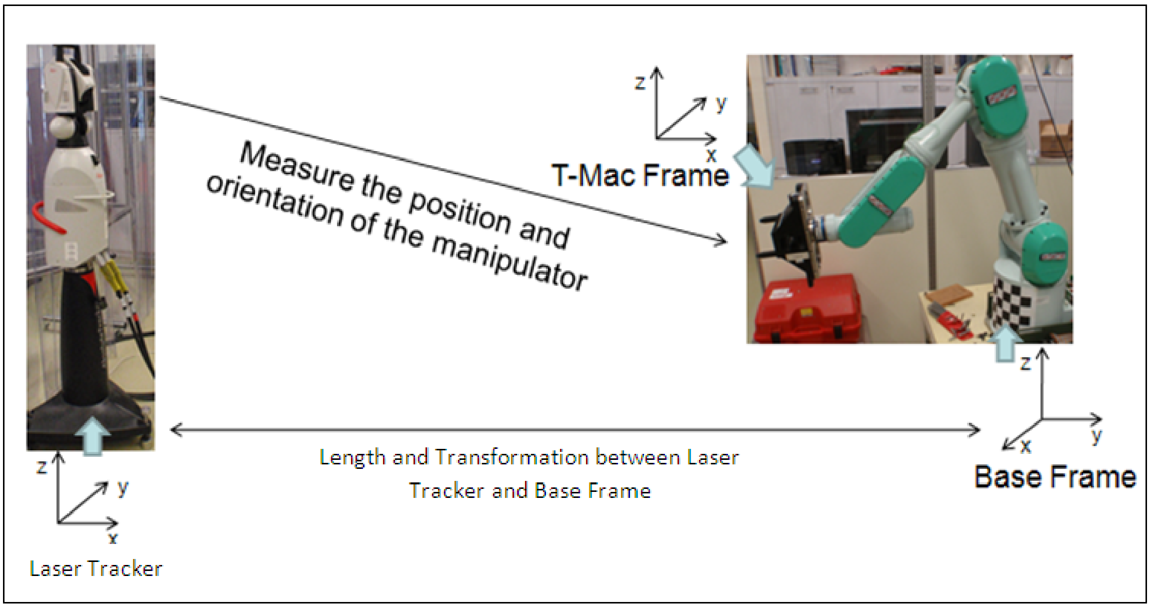

Figure 3. Experiment set up

As can be seen from Figure 3, the laser tracker will follow the end effector movement. However, the laser tracker has a limited workspace range. In order to measure the pose of manipulator, the end effector needs to be within the workspace range of the laser tracker. The angle of each joint is limited to ± 15 degrees to satisfy the workspace of laser tracker. In order to show the calibration result between an optimal set of poses and a random set of poses, several experiments are conducted. The first experiment utilizes 50 random poses. The experiment is using conventional local POE that proposed by I.M. Chen, et al [3]. This number of random poses is chosen because the local POE calibration method needs more poses to give a better result. From the experiment, it is shown that 50 random poses could give an acceptable calibration result. In the second experiments, we decrease the number of poses for calibration gradually from 50 poses until 20 poses. The goal of the second experiment is to evaluate the effect of reducing the number of poses on the calibration performance. The third experiment uses 20 poses which are obtained from the genetic algorithm to do the calibration. It is expected that the calibration could gives a similar calibration performance but using less number of poses compared to other local POE calibration [3][5][6].

Experiment 1: 50 Random Poses

In the first experiment, the maximum iteration of the calibration is ten times. Using Equation 10 and 11 the position and orientation deviation can be calculated. Figure 4. shows the position and orientation deviation from the actual and calibrated initial poses at each iteration.

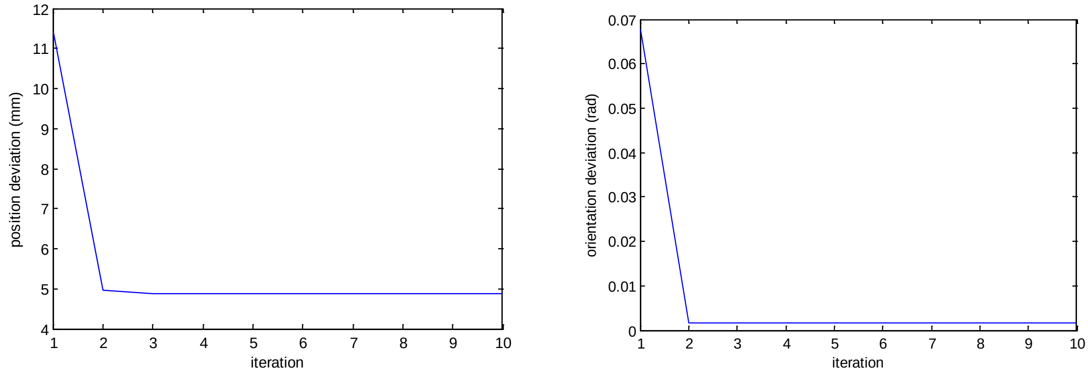

Figure 4. Position and orientation error at each iteration

Figure 4 shows that the calibration could reduce the position and orientation deviation of the manipulator after ten iterations. The position and orientation deviation converges after three iterations. Using Equation 17 and Equation 18, the final average position deviation is around 4.8 mm and the final average orientation deviation is around 0.0016 rad. The position and orientation deviation is reduced; before calibration, the average position deviation is around 11.4 mm, and the average orientation deviation is around 0.067 rad.

Experiment 2: Towards Minimum Number of Poses

As mentioned in the previous chapters, different number of poses and configurations will lead to the different calibration results. A higher number of poses will give better calibration results but at the expense of the time required for calibration. On the other hand, a fewer number of poses could give a poorer calibration result. In order to see the effect of decreasing the number of poses for the calibration, several (seven) different sets of poses is used to do the calibration. At the beginning, each set of calibration consists of 50 random poses, later the number of poses for calibration is gradually decreased from 50 poses until 20 poses with 5 poses intervals. The effect of the number of poses can be seen in Figure 5.

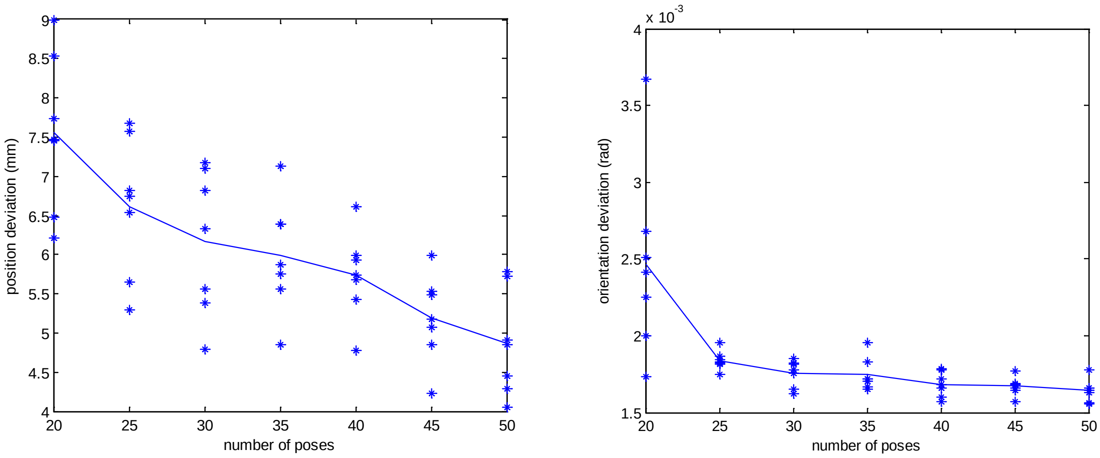

Figure 5. Position and orientation deviation for different set of number of poses (* is the result for different set of data, line is the average position/orientation error)

As can be seen from figure above, increasing the number of poses to do the calibration could improve the result of calibration. In order to improve the performance of the calibration using fewer poses, genetic algorithm is introduced. The details are discussed in the next chapter.

Experiment 3: GA Optimized Poses

In this experiment, the genetic algorithm is used to evaluate the poses as genes and the set of poses as an individual. The set of poses consist of 20 poses. The population is consisted of 200 individuals, and number of generation set at 200. The fitness function used is the observability index from the set of poses. The mutation rate is set as 15 % of the population. The offspring of the genetic algorithm is 70% from the total population.

Each set of poses contains 20 poses. This value is chosen, to give faster calculation and measurement time. Using genetic algorithm which utilizes several observability indices as its fitness function, the optimal set of 20 poses can be obtained. Several fitness functions values which utilize G, D, A and E Optimality for each generation can be seen in Figure 6.

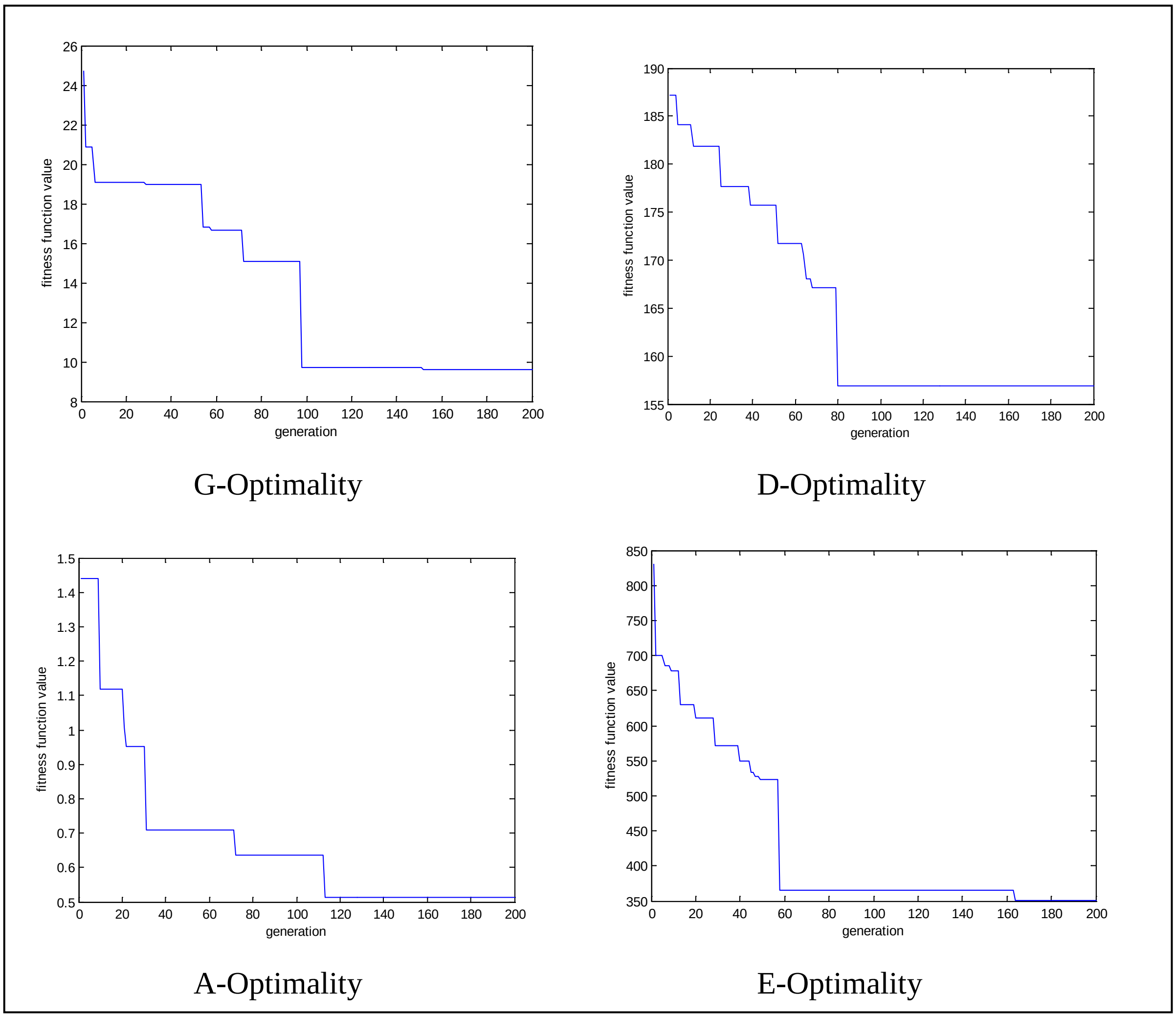

Figure 6. Fitness function value for each observability indices

Figure 6 shows that the genetic algorithm continuously seeks the minimum value of the fitness function. In this research, a smaller fitness function implies better performance of the calibration. After the GA determines the best set of poses for each observability index, this set of poses then can be calibrated using the least square method. The position and orientation deviation information of each set will be compared for each observability indices formula. To check the validity of the calibrated model for each observability index, 60 random poses are used to verify the result. The result can be seen in Table 1.

Table 1. Calibration Testing and Verification for Each Observability Index

Error With Calibration Set			Testing With Verification Pose		
Type	**Position Deviation (mm)**	**Orientation Deviation (rad)**	**Type**	**Position Deviation (mm)**	**Orientation Deviation (rad)**
A-Optimality	3.9298	0.0015	A-Optimality	4.885	0.0016
D-Optimality	3.6661	0.0014	D-Optimality	5.2687	0.0019
G-Optimality	3.6444	0.0016	G-Optimality	3.3743	0.0018
E-Optimality	3.2917	0.0014	E-Optimality	5.2834	0.0019
50 Random Pose	4.8640	0.0015	50 Random Pose	4.9626	0.0016

As can be seen from Table 1, the set of poses obtained from genetic algorithm can perform the calibration with less poses than the previous experiment. The result of the calibration using 50 random poses and the result using 20 poses with genetic algorithm gives similar position and orientation deviation. It is shows that the optimization is working well.

Conclusions and Future Works

The experiment shows that the optimal poses selection using genetic algorithm could reduce the number of poses and improve the calibration performance. It is shown that this method could be an alternative solution to improve the calibration performance especially in the calibration time. From the experiment, it could be seen that in this case, G-optimality could perform better performance compared to other optimality indices and to random poses.

Although the work shows some improvements, it is possible to further improve the current performance of the calibration. One way is to constrain the movement of the end effector at the task space level instead of at the joint space level. The task space constraint could eliminate the joint constraint while maintaining the end effector within the operational range of laser tracker.

References

[1] C.B. Wang, J. Chen, and J.C.S Yang, “Robot positioning accuracy improvement through kinematic parameter identification,” Paper presented at the *Third Canadian CAD/CAM and Robotics Conference*, Toronto, Canada, 1984.

[2] A. Doria, F. Angrilli, and S. De Marchi, “Inverse kinematics robot calibration by spline functions,” *Applied Math Modelling*, Vol. 17, pp. 492-498, 1993.

[3] I.M. Chen, G. Yang, C.T. Tan, and S.H. Yeo, “Local POE model for robot kinematic calibration,” *Mechanism and Machine Theory*, Vol. 36, pp. 1215-1239, 2001.

[4] F.C. Park, and K. Okamura, *Kinematic Calibration and the Product of Exponential Formula in: Advances in Robot Kinematics and Computational Geometry*, Cambridge, MIT Press, pp. 119-128, 1994.

[5] G. Yang, and I.M. Chen, "A novel kinematic calibration algorithm for reconfigurable robots," In: *Proceeding of IEEE Conference on Robotics and Automation*, pp. 3197-3202, 1997.

[6] S.K. Mustafa, G. Yang, S.H. Yeo, and W. Lin, "Kinematic calibration of a 7-DOF self calibrated modular cable driven robotic arm," *IEEE International Conference on Robotics and Automation,* Pasadena, United States, May 19-23, pp.1288-1293, 2008.

[7] H. Chernoff, "*Locally optimal design for estimating parameters*," Annals of Mathematical Statistic, Vol. 24, No. 4, pp. 586-602, 1953.

[8] A. Wald, "*On the efficient design of statistical investigations*," Annals of Mathematical Statistic, Vol. 14, No. 2, pp. 134-140,1943.

[9] K. Smith, "On the standard deviation of adjusted and interpolated values of an observed polynomial function and its constant and the guidance they give towards a proper choice of the distribution of observations," *BioMetrika*, Vol. 12, No. 1-2, pp. 1-85, 1918.

[10] S. Ehrenfeld, "*On the efficiencies of experimental design*," Annals of Mathematical Statistic, Vol. 26, No. 2, pp. 247-255, 1955.

[11] Y. Sun, and J.M. Hollerbach, "Observability index selection for robot calibration," Paper presented at the *IEEE International Conference on Robotic and Automation*, United States, May, 2008.

[12] D. Goldberg, *Genetic Algorithm in Search, Optimization, and Machine Learning Vision*, Addison Wesley, 1989.

[13] R. Brockett, "Robotic manipulators and the product of exponential formula," In *Mathematical Theory of Network and System*, P.A. Fuhrman, ed.: Springer, New York, United States, pp.120-129, 1984.