Position and orientation accuracy of the end-effector is affected by the precision of kinematic parameters elements of the robot. Thus, good precision requires good knowledge of robot physical parameters values. However, this condition can be difficult to meet in practice. Hence, calibration techniques can be devised in order to improve the robot accuracy through estimation of those particular parameters. In this paper, the Genetic Algorithm is used to calibrate the robot kinematic accuracy. A kinematic model is formulated and conducted as an optimization problem for serial robot manipulators. The objective is to analyze and evaluate the performance of the GA in optimizing such robot kinematic accuracy. In this algorithm, the errors in the robot parameters represent the parents and offspring population and the error matrix norms represent the cost functions. The convergence and effectiveness of the presented model are demonstrated by a numerical example.

1. INTRODUCTION

Genetic Algorithm (GA) is a rather new search tool in robotics which exhibit high efficiency in certain multi-modal and multi-dimensional domains. The algorithm has been implemented to handle the optimization problem for a number of different application areas such as robotics, composite materials design, scheduling and also salesman problems. Recently, it is a gate for the automatic design which is the best way to develop artificial intelligence. It can be easily adapted to the planning problems for many types of assembly machines. The GA uses population as parents to represent possible solutions. The fitness of all of the individuals in the population is evaluated by its cost. The genetic operators (such as crossover, inversion, rotation and mutation) are applied to generate new population called offspring in an iterative procedure to obtain nearly optimal solutions.

The position and orientation of the end-effector would have no errors other than those caused by imperfection of the repeatability and dynamic effects. However, in more sophisticated applications, errors in the position and orientation of the end-effector result from the kinematic parameters errors as well, which are mostly due to manufacturing and/or measurement errors.

The calibration of robot manipulators has attracted many researchers. Veitschegger and Wu have presented a result comparison between two models. The first model has ignored the higher order terms and did not address the special case of two consecutive parallel joints while the second model has considered both cases. Wu has used a new technique to correct the kinematic errors of robot manipulator. Vukobratović and Borovac have investigated the influence of the deviations of the links nominal measures (due to manufacturing tolerances) on the accuracy of positioning the manipulator tip for various mechanism configurations Bruyninckx et al. have developed a systematic and fully general model-based approach to compliant robot motion, taking into account uncertainties in the geometry of the manipulated object and the environment with which it is in contact. Samak et al. have studied the effect of kinematic perturbations on robot precision. Kazerowian and Qian have presented a kinematic calibration model for position and orientation of serial manipulator end-effector errors due to repeatability imperfections.

This paper introduces a new perspective proposal in order to analysis, implement and evaluate the performance of the GA in optimizing the robot kinematic accuracy. The kinematic relationships have been described by using the zero-reference-position (ZRP) method. The prescribed analysis showed improvement in the robot accuracy precision.

2. ZERO POSITION ANALYSIS METHOD

The zero position analysis method was introduced by Gupta. It has the advantages that it is not prone to the discontinuity difficulties as those in the Denavit Hartenberg notation. Due to the nature of this method, small changes in the structure inherently correspond to small changes in the structure parameters. It has also proven its effectiveness and versatility in many works on both kinematic and dynamic analysis of robot manipulators. The joint coordinate systems in this method are not used. Instead, a convenient reference position of the robot is chosen and the following vectors are defined in the world coordinate system.

- $\mathbf{u}_i$ a unit vector along joint axis $i$.
- $\mathbf{b}_i$ a body vector which connects a point on joint $(i-1)$ to a point on joint $i$.
- $\mathbf{w}_i$ and $\mathbf{u}_i$ two perpendicular vectors fixed on the end-effector.

All the above-mentioned parameters are given in their ZRP (with zero subscript). They are converted to the current position...
as the manipulator moves to new positions. The current vector are derived from their ZRP vectors as follows,

\[ u_i = R_i u_{bi} \quad (1) \]

\[ b_{i+1} = R_i b_{j_i} \quad (2) \]

\[ u_a = R_a u_{ba} \quad \text{and} \quad u_i = R_i u_{bi} \quad (3) \]

Where \( i = 1, 2, \ldots, n \), and the 3 by 3 rotation matrix \( R_i \) for a revolute joint, is defined as

\[ R_i = R(q_i, u_{bi}) R(q_{i+1}, u_{bi+1}) \ldots R(q_n, u_{bn}) \]

\[ = \prod_{j=1}^{n} R(q_j, u_{bj_i}) \quad (4) \]

Hence, for \( n \)-revolute joints manipulators, the above equation represents the hand orientations \( R_h \) as

\[ R_h = \prod_{i=1}^{n} R(q_i, u_{bi}) \quad (5) \]

The matrix \( R(q_i, u_{bi}) \) represents a rotation by \( q_i \) about a screw axis \( u_{bi} \). It can be written as

\[ R(q_i, u_{bi}) = \begin{bmatrix} u_{bi} & -v_{bi} & \frac{1}{2} ||u_{bi}||^2 - 1 \end{bmatrix} \begin{bmatrix} 0 & -v_{bi} & u_{bi} \\ v_{bi} & 0 & -u_{bi} \\ -u_{bi} & v_{bi} & 0 \end{bmatrix} \begin{bmatrix} 1 - \cos(q_i) & 0 & \sin(q_i) \\ 0 & 1 & 0 \\ \sin(q_i) & 0 & 1 - \cos(q_i) \end{bmatrix} \begin{bmatrix} 0 & -u_{bi} & v_{bi} \\ u_{bi} & 0 & -v_{bi} \\ v_{bi} & u_{bi} & 0 \end{bmatrix} + \begin{bmatrix} 0 & -u_{bi} & v_{bi} \\ u_{bi} & 0 & -v_{bi} \\ v_{bi} & u_{bi} & 0 \end{bmatrix} \]

\[ \text{where,} \quad V_i = 1 - \cos(q_i) \text{ and } S_i = \sin(q_i), \text{ and } u_{bi}, v_{bi} \text{ and } u_{bi} \text{ are components of the unit vector } u_{bi}. \text{ If the } i^{th} \text{ joint is prismatic, then } R(q_i, u_{bi}) \text{ is replaced with a 3 by 3 identity matrix.} \]

Equation (6) can be decomposed as follows,

\[ R(q_i, u_{bi}) = V_i A + S_i B + I \quad (7) \]

where

\[ A = \begin{bmatrix} u_{bi}^2 - 1 & u_{bi} v_{bi} & u_{bi} v_{bi} \\ u_{bi} v_{bi} & u_{bi}^2 - 1 & u_{bi} v_{bi} \\ u_{bi} v_{bi} & u_{bi} v_{bi} & u_{bi}^2 - 1 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 0 & -u_{bi} & u_{bi} \\ u_{bi} & 0 & -u_{bi} \\ -u_{bi} & u_{bi} & 0 \end{bmatrix} \quad (8) \]

3. ROTATIONAL ERROR MODEL

For small changes in \( u_a, u_i \) and \( u_{bi} \), the corresponding small change in the rotation matrix \( R(q_i, u_{bi}) \) can be shown to be

\[ \delta R(q_i, u_{bi}) = R_{ma} \delta u_{bi,x} + R_{mi} \delta u_{bi,y} + R_{mi} \delta u_{bi,z} \quad (9) \]

where

\[ R_{ma} = \begin{bmatrix} 2X_i & Y_i & Z_i \\ Y_i & 0 & -S_i \\ Z_i & S_i & 0 \end{bmatrix} \]

\[ R_{mi} = \begin{bmatrix} 0 & X_i & S_i \\ -X_i & 2Y_i & Z_i \\ -S_i & Z_i & 0 \end{bmatrix} \]

\[ R_{mi} = \begin{bmatrix} 0 & -S_i & X_i \\ S_i & 0 & Y_i \\ X_i & Y_i & 2Z_i \end{bmatrix} \quad (10) \]

and \( X_i = u_{bi} V_i, \quad Y_i = u_{bi} V_i \) and \( Z_i = u_{bi} V_i \)

For small changes in \( R(q_i, u_{bi}) \), the hand orientation, Eq. (5), becomes

\[ R_h + \delta R_h = \prod_{i=1}^{n} [R(q_i, u_{bi}) + \delta R(q_i, u_{bi})] \quad (11) \]

Ignoring second and higher orders of variations, the above equation leads to

\[ \delta R_h = \sum_{i=1}^{n} \left[ R_{ma} \delta u_{bi,x} + R_{mi} \delta u_{bi,y} + R_{mi} \delta u_{bi,z} \right] \quad (12) \]

The left-hand side of Eq. (12) is a skew symmetric matrix. It has only three significant elements namely (3,2), (1,3) and (2,1) or \( \delta_{r_2}, \delta_{r_3} \) and \( \delta_{r_1} \). Therefore, \( \delta R_h R_h^{-1} \) could be converted to a vector \( \delta_h \), which shows the errors in the end-effector orientation. Substituting \( \delta R(q_i, u_{bi}) \) from Eq. (9) into Eq. (12) yields

\[ \delta_h = \sum_{i=1}^{n} \left[ R_{ma} \delta u_{bi,x} + R_{mi} \delta u_{bi,y} + R_{mi} \delta u_{bi,z} \right] \quad (13) \]

4. POSITIONAL ERROR MODEL

The position vector \( P_p \) of a reference point \( p \) at the hand

\[ P_p = \sum_{i=1}^{n} b_{i+1} = \sum_{i=1}^{n} R_i b_{j_i} \]

\[ = \sum_{i=1}^{n} \left( \prod_{j=i}^{n} R(q_j, u_{bj_i}) \right) b_{j_i+1} \quad (14) \]

For small changes in \( P_p \), the position vector becomes

\[ P_p + \delta P_p = \sum_{i=1}^{n} \left[ \prod_{j=i}^{n} R(q_j, u_{bj_i}) \right] b_{j_i+1} + \delta b_{j_i+1} \]

(15)

Ignoring second and higher orders of variations, the above equation leads to

\[ \delta P_p = \sum_{i=1}^{n} \left[ R_{ma} \delta u_{bi,x} + R_{mi} \delta u_{bi,y} + R_{mi} \delta u_{bi,z} \right] \quad (16) \]

Eq. (16) shows the errors in the end-effector position. Substituting \( \delta R(q_i, u_{bi}) \) from Eq. (9), the above equation becomes

\[ \delta P_p = \sum_{i=1}^{n} \left[ R_{ma} \delta u_{bi,x} + R_{mi} \delta u_{bi,y} + R_{mi} \delta u_{bi,z} \right] \quad (17) \]
where
\[ E = b_{0,i} + \sum_{j=i}^{n} (R(q_j, u_{0j}) b_{0,j+1}) \]  
(18)

In order to insure the length constraint of the unit vectors, the following constraint is to be satisfied,
\[ |u_{0j}| = \sqrt{u_{0j,x}^2 + u_{0j,y}^2 + u_{0j,z}^2} = 1 \]  
(19)

For small changes,
\[ u_{0j,x} \delta u_{0j,x} + u_{0j,y} \delta u_{0j,y} + u_{0j,z} \delta u_{0j,z} = 0 \]
or
\[ u_{0j}^T \delta u_{0j} = 0 \]  
(20)

The above constraint is also implemented as an extension to the error Jacobian matrix.

5. ERROR JACOBIAN MATRIX

From Eqs. (13,17,20), the following equation can be constructed,
\[ J \begin{bmatrix} \delta u_0 \\ \delta b_0 \end{bmatrix} = \begin{bmatrix} \delta q_h \\ \delta p_p \end{bmatrix} \]  
(21)

The above equation describes a linear relationship between the errors in the robot kinematic parameter elements (\(\delta u_i\) and \(\delta b_i\)) which can be defined as follows
\[ \delta u_i = (\delta u_{0i,x} \ \delta u_{0i,y} \ \delta u_{0i,z} \ \cdots \ \delta u_{0i,n})^T \]  
(22)
\[ \delta b_i = (\delta b_{0i,x} \ \delta b_{0i,y} \ \delta b_{0i,z} \ \cdots \ \delta b_{0i,n})^T \]  
(23)

and the error in the position and orientation of the end-effector (\(\delta r_h\) and \(\delta p_p\)).

The matrix \(J\) represents the error Jacobian matrix, which can be expressed as
\[ J = \begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \\ J_5 & J_6 \end{bmatrix} \]  
(24)

The \(i^{th}\) elements of the submatrix \(J_1\) are defined as
\[ J_{1i} = [R_{i-1} R_{i} R_{i}^T R_{i-1} R_{i} R_{i}^T R_{i-1} R_{i} R_{i}^T] \]  
(25)
while \(J_2 = 0\); and the \(i^{th}\) elements of \(J_3\) are
\[ J_{3i} = [R_{i-1} R_{i} E \ E R_{i-1} R_{i} E R_{i-1} R_{i} E] \]  
(26)

where \(E\) is given by Eq. (18), and
\[ J_4 = -R_i, \quad J_5 = -u_i, \quad \text{and} \quad J_6 = 0 \]  
(27)

\(J_1\) to \(J_4\) are \(3 \times 6\); while \(J_5\) and \(J_6\) are \(n \times 6\). Therefore, the error matrix \(J\) is then \((6 + n) \times 6\). The errors in the kinematic parameter elements (\(\delta u_i\) and \(\delta b_i\)).

6. CALIBRATION ALGORITHM

In the calibration algorithm, the kinematic parameter errors are used to represent the GA\(^*\) population and their Spectral norms represent the GA\(^*\) cost functions. The algorithm could be proposed as follows,
1. The nominal link parameters \((u_0\) and \(b_0\)) and the joint variables \(q\) are used for an arbitrary configuration.
2. The nominal hand orientation and position \(R_h\) and \(P_p\) are computed by using Eqs. (5,14).
3. The error Jacobian matrix \(J\) is constructed by using Eqs. (25 - 27).
4. The actual joint variables \(q_a\) are calculated by adding a range of random error values.
5. The GA iterations is started by generating some initial populations for the link parameter errors \(\delta u_i\) and \(\delta b_i\). These populations are used as parents from which the genetic operators are applied to produce new offspring population.
6. The actual hand orientation and position \(R_h^a\) and \(P_p^a\) are computed as in step 2 by utilizing the actual joint variables.
7. The right hand side of Eq. (21) is obtained as
\[ \delta r = \delta R_h R_h^a \quad \text{and} \quad \delta P_p = P_p^a - P_p \]  
(28)

where
\[ \delta R_h = R_h^a - R_h \]  
(29)

8. The offspring populations are computed from the following relationship (the Pseudo inverse is used).
\[ \begin{bmatrix} \delta u_0 \\ \delta b_0 \end{bmatrix} = J^{-1} \begin{bmatrix} \delta q_h \\ \delta p_p \end{bmatrix} \]  
(30)

Then, their population costs are evaluated by computing their Spectral norms.
9. The offspring together with their parents are evaluated by their costs. The most fit population are those with the lowest costs.
10. This process is iterated until a certain criterion (such as a certain number of iterations) is met.

7. AN EXAMPLE

A Numerical example is presented in an arbitrary configuration for a six-degrees of freedom PUMA-type manipulator (Figure 1). Its joint variables are chosen as
\[ q = (-2.741 \ 4.501 \ 2.609 \ 2.044 \ 0.389 \ 2.285)^T \]
Figure 1: PUMA-type robot in its ZRP configuration

The nominal kinematic parameters are listed in Table 1. The joint value errors are randomly taken from a range of ± 0.001 radians; while the kinematic parameter errors are randomly taken from a range of ± 0.01 for screw axes $\mathbf{u}^i$ and ± 0.5 for body vectors $b^i$.

<table>
<thead>
<tr>
<th>$\delta u_{01} : \delta u_{06}$ ($\times 10^{-2}$)</th>
<th>$\delta b_{02} : \delta b_{07}$ (mm)</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.29, 0.64, 0.57, -0.12, 0.95, -0.29</td>
<td>0.15, 0.09, -0.26, 0.23, 0.31, -0.48</td>
<td>0.0112</td>
</tr>
<tr>
<td>0.76, -0.51, 0.53, 0.00, -0.99, 0.77</td>
<td>0.29, 0.07, -0.37, 0.27, 0.47, -0.08</td>
<td>0.0127</td>
</tr>
<tr>
<td>0.01, 0.63, 0.25, 0.92, 0.77, -0.26</td>
<td>0.09, -0.47, -0.07, -0.36, 0.16, -0.00</td>
<td>0.0127</td>
</tr>
</tbody>
</table>

Table 1: Nominal kinematic parameters

Initial populations are generated to represent the parents from which an initial set of errors along with their initial cost are computed and listed in Table 2. The GA operators are then applied to generate the offspring. This algorithm is converged after 11 iterations. The optimal kinematic parameter errors and their cost are given in Table 3. Whereas, the algorithm convergence are shown in Figure 2.

From the foregoing example, it has been shown that the initial cost is 0.0112 and the final optimal cost is 0.0027. Therefore, the optimal kinematic parameter errors are reduced by 24.1%.

8. CONCLUSIONS

The Genetic Algorithm procedure is used to calibrate the robot kinematic errors based on the zero position analysis method. The effectiveness of the algorithm and its convergence in the presence of small joint errors and measurement errors is demonstrated through a numerical experiment. The kinematic parameter errors are reduced by 24.1%.

REFERENCES


