Inverse Kinematic Solution for Robot Manipulator Based on Electromagnetism-like and Modified DFP Algorithms

YIN Feng[1] WANG Yao-Nan[1] WEI Shu-Ning[1]

Abstract A new method for computing numerical solutions to the inverse kinematics problem of robotic manipulators is developed in this paper. With the joint limitations, the electromagnetism-like method (EM) utilizes an attraction-repulsion mechanism to move the sample points towards the optimum solution rapidly. Based on this approximate solution given by EM, a modified Davidon-Fletcher-Powell (DFP) algorithm is developed to solve the problem at the desired precision. Unlike the traditional algorithms, this modified DFP (MDFP) algorithm randomly chooses the search step size between 0 and 1. Hence, the computational complexity is greatly reduced. The experimental results based on ten general test functions and PUMA 560 robot show that this new near-real time hybrid method can produce best performance.

Key words Robot manipulator, inverse kinematics (IK), electromagnetism-like method (EM), variable metric method

DOI 10.3724/SP.J.1004.2011.00074

The inverse kinematics (IK) problem for a serial-chain manipulator is to find the values of the joint positions given the position and orientation of the end-effector. The problem is important for robot trajectory planning, motion control, and workspace analysis.

The problem solving methods are generally divided into closed-form solution methods and numerical methods. Closed-form solution methods, including algebraic and geometric methods[1−2], are desirable because they are faster and easily identify all possible solutions. The major disadvantage of closed-form solutions is that they are not general, but robot dependent. The most effective methods for finding closed-form solutions are Ad Hoc techniques that take advantage of particular geometric features of specific mechanisms. So, these methods are available only for certain classes of industrial manipulators with simplified structures.

Contrarily, numerical methods are not robot dependent, so they can be applied to any kinematic structure. Raghavan et al.[3] used dialytic elimination to reduce the IK problem of a general six-revolute serial-chain manipulator to a polynomial of degree 16 and find all possible solutions. The roots provide solutions for one of the joint variables, while the other variables are computed by solving linear systems. Manocha et al.[4] improved the numerical properties of this technique by reformulating the problem as a generalized eigenvalue problem. Husty et al.[5] made use of classical multidimensional geometry to structure the IK problem and to use the geometric information before starting the elimination process. Then, the 6R-chain was broken up in the middle to form two open 3R-chains and the 16 solutions of the inverse kinematics were obtained. Qiao′s recent work[6] showed that homogeneous transform matrix in terms of double quaternion could lead to double kinematic equations of 6R robots; a 16th degree univariate polynomial was yielded from the resultant matrix via linear algebra and Dixon resultant formulation. Such methods, however, appear to be not suitable for solving the high degree of freedom (DOF) problem, as the computational complexity of these methods are highly dependent on the number of the objective equations. In addition, these methods may produce extraneous roots in solving some joint variables.

Furthermore, a number of different iterative methods were employed to solve the IK problem. Essentially, these methods make use of the Newton Raphson scheme[7−8], modified predictor-corrector algorithm[9], separation of variable method[10] or adopt various optimization algorithms in order to solve an equivalent minimization problem[11]. Instead of solving directly, most of them used gradient-based nonlinear programming algorithms to solve the equivalent minimization problem. In general, since the inverse Jacobian matrix was not used, these methods were numerically more stable. However, most of them converge to a single solution based on an initial guess, so the quality of that guess greatly impacts the convergence time. The Cyclic-coordinate descent (CCD) method developed by Wang et al.[11] is an iterative heuristic search method and its initial approximation of the solution vector can be arbitrary. However, due to the heuristic nature of this method, the rate of convergence is highly dependent on the structure of the manipulator.

Recently, a few attempts were made to apply artificial neural network (ANN)[12−14] for prediction of inverse kinematic solutions. Essentially, ANN approximates inverse kinematics relations of the robot in order to map the Cartesian configuration into corresponding joint angles. To obtain better learning performance, a large number of training patterns need to be given. However, generation of such large data set is very difficult in practice for a robot situated in a cluttered workspace. Furthermore, the training data obtained from derived inverse kinematic equations may contain mapping error due to nonlinear mapping between joint angle coordinates and Cartesian coordinates leading to inaccuracies in predicted inverse kinematic solutions[15]. In addition, certain hybrid techniques made use of ANN along with expert system[16], fuzzy logic[17], and genetic algorithm[18] for obtaining inverse kinematic solutions. Though these intelligence approaches can easily provide inverse kinematic solutions for two or three DOF planar robots, these methods demanded for high performance computing systems and complex computer programming for obtaining the solutions of more DOF robots[15].

In this paper, a novel hybrid algorithm based on electromagnetism-like method (EM)[19] and modified Davidon-Fletcher-Powell (MDFP) was developed to solve the IK problem. EM had been tested on available test problems in [19], and it showed that EM can converge to the optimal solution in equations evaluations of variable

Manuscript received May 25, 2010; accepted October 13, 2010

Supported by National High Technology Research and Development Program of China (863 Program) (2008AA04Z214) and National Natural Science Foundation of China (2008BAF36B01)

1. Department of Electrical and Information Engineering, University of Hunan, Changsha 410082, P. R. China

dimension less than 30 without any first or second order derivative information. A theoretical study of EM analysis for convergence to the optimal solution was presented in [20]. Recently, EM had been applied to solve various optimization problems successfully, such as scheduling optimization[21], multi-objective optimization[22], and NP optimizations problems[23] and so on. However, at present, there is very little research which uses EM to solve IK problems, and almost no papers can be found in this area in public. As we know, the IK problem can be transformed into an equivalent minimization problem[7−8, 11]. As the number of joints of most robot manipulators are much less than 30, the equivalent minimization problem belongs a low-dimensional optimization problem. According to conclusions in [19], this problem appears to be easily resolved by EM theoretically. Moreover, EM can be easily combined with other optimization methods. Thus, a hybrid approach that made use of EM and MDFP are proposed to resolve the IK problem in this paper. It first used EM to rapidly find a feasible point that is near to the true solution and then used the MDFP to obtain a solution at the desired degree of precision. Due to exploiting the strength of two methods, the hybrid method is more efficient and precise.

The rest of paper is organized as follows. The formulation of the problem and the objective function are presented in Section 1. Section 2 briefly introduced EM and a numerical example using EM to solve IK problem is given. In Section 3, an MDFP algorithm is developed and performance evaluation of the hybrid approach is shown through 10 general test functions. Further, a numerical example for solving the IK problem for the PUMA 560 robot is given. Finally, discussions and conclusions are given in Section 4.

1 Problem formulation

As shown in Fig. 1, the desired position vector and orientation matrix of a manipulator end-effector are denoted by: $\boldsymbol{P}_d$ and $[\boldsymbol{R}_d] = [\boldsymbol{d}_1, \boldsymbol{d}_2, \boldsymbol{d}_3]$, where $\boldsymbol{d}_j$ $(j = 1, 2, 3)$ are unit vectors along the x_d, y_d, z_d axes. $\boldsymbol{P}_h$ is the current position vector of the end-effector. The current orientation matrix is defined by: $[\boldsymbol{R}_h] = [\boldsymbol{h}_1, \boldsymbol{h}_2, \boldsymbol{h}_3]$, where $\boldsymbol{h}_j$ $(j = 1, 2, 3)$ are unit vectors along the x_h, y_h, z_h axes and the joint variables are denoted by the $n \times 1$ vector, $\boldsymbol{\theta} = [\theta_1, \theta_2, \cdots, \theta_n]^{\rm T}$.

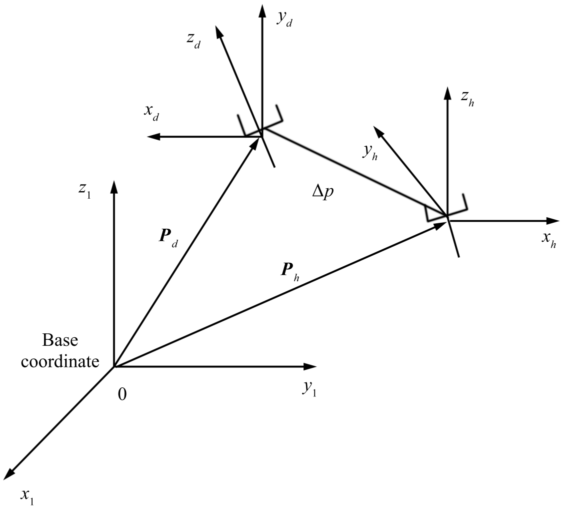

Fig. 1 The current and desired end-effector configurations

Therefore, the errors between the current and the desired locations of the end-effector can be described by the following functions[11]:

Position error:

$$\Delta p(\theta) = \|\boldsymbol{P}_d - \boldsymbol{P}_h(\theta)\| \tag{1}$$

Orientation error:

$$\Delta o(\theta) = \sum_{j=1}^{3}(\boldsymbol{d}_j \cdot \boldsymbol{h}_j(\theta) - 1)^2 \tag{2}$$

The total error:

$$e(\theta) = \Delta p(\theta) + \Delta o(\theta) \tag{3}$$

where $(\cdot)$ denotes the vector dot product. Now, the inverse kinematics problem is to find a solution θ^*, such that $e(\theta^*) \le \varepsilon$ $(\varepsilon \to 0)$. This problem can be transformed into the following equivalent minimization problem:

$$\min e(\theta) \quad \text{s.t.} \quad \theta \in \mathbf{R}^n | l_k \le \theta_k \le u_k,\ k = 1, 2, \cdots, n \tag{4}$$

where l_k and u_k are the lower and upper bounds of the i-th joint variable, respectively.

2 Review of EM

EM is a new stochastic search method similar to genetic algorithm (GA)[19]. To solve the problem in (4), the general scheme of EM is given by following procedure:

```
Initialize;
While termination criteria are not satisfied do;
    Local search;
    Calculation of charge and total force vector;
    Movement according to the total force;
End while
```

2.1 Initialization

The procedure initialization is used to sample m points, $\{\theta^1, \cdots, \theta^m\}$, randomly from the feasible domain of the joint variables, where $\theta^i = [\theta^i_1, \cdots, \theta^i_n]$ $(i = 1, \cdots, m)$. The procedure of uniform sampling can be determined by

$$\theta^i_k = l_k + \text{rand} \cdot (u_k - l_k), \quad k = 1, 2, \cdots, n \tag{5}$$

The procedure ends with m points identified, and the point that has the best function value is stored in $\theta^{\rm best}$.

2.2 Local search

The local search procedure is used to gather the local information and improve the current solutions. It can be applied to one or many points for local refinement at each iteration. The selection of these two procedures, does not affect the convergence result.

2.3 Calculation of charge and total force vector

The charges of the points are calculated according to their objective function values, and the charge of each point is not constant and changes from iteration to iteration. The charge of the i-th point, q^i, is evaluated as

$$q^i = \exp\left[-n\frac{(e(\theta^i) - e(\theta^{\rm best}))}{\sum\limits_{k=1}^{m}(e(\theta^k) - e(\theta^{\rm best}))}\right],\ i = 1, 2, \cdots, m \tag{6}$$

In this way, the points that have better objective values possess higher charges. Notice that, unlike electrical charges, no signs are attached to the charge of an individual point in (6). Instead, the direction of a particular force between two points is decided after comparing their objective function values. Hence, the total force $\boldsymbol{F}^i$ exerted on point i is computed by the following equation:

$$\boldsymbol{F}^i = \begin{cases} \sum\limits_{j\neq i}^{m}(\theta^j - \theta^i)\dfrac{q^j q^i}{\parallel \theta^j - \theta^i \parallel^2}, & \text{if } e(\theta^j) < e(\theta^i) \\ \sum\limits_{j\neq i}^{m}(\theta^i - \theta^j)\dfrac{q^j q^i}{\parallel \theta^j - \theta^i \parallel^2}, & \text{others} \end{cases} \tag{7}$$

According to (7), the point that has a better objective function value attracts the other one. Contrarily, the point with worse objective function value repels the other. Since θ^{best} has the minimum objective function value, it acts as an absolute point of attraction. Then, it attracts all other points in the population to better region. The procedure of calculation for total force vector is as following:

```
for $i = 1$ to $m$ do
    $q^i = \exp\left[-n\frac{e(\theta^i)-e(\theta^{\text{best}})}{\sum_{k=1}^{m}(e(\theta^k)-e(\theta^{\text{best}}))}\right]$, $\boldsymbol{F}^i = 0$;
end for
for $i = 1$ to $m$ do
    for $j = 1$ to $m$ do
        if $i \neq j$ then
            $F^i_j = (\theta^j - \theta^i)\frac{q^i q^j}{\|x^j - x^i\|^2}$;
            if $f(x^j) < f(x^i)$ then
                $F^i = F^i + F^i_j$ {Attraction};
            else
                $F^i = F^i - F^i_j$ {Repulsion};
            end if
        end if
    end for
end for
```

2.4 Movement according to the total force

After evaluating the total force vector $\boldsymbol{F}^i$, the point i is moved in the direction of the force by a random step length in (8). Here, the random step length λ is assumed to be uniformly distributed between 0 and 1.

$$\theta^i = \theta^i + \lambda\frac{\boldsymbol{F}^i}{\|\boldsymbol{F}^i\|}\boldsymbol{R}_{NG}, \quad i = 1, 2, \cdots, m \tag{8}$$

In (8), $\boldsymbol{R}_{NG}$ is a vector whose components denote the allowed feasible movement toward the upper bound u_k or the lower bound l_k of the joint variables. The procedure is as following:

```
for $i = 1$ to $m$ do
    if $i \neq$ best then
        $\lambda = U(0, 1)$;
        $\boldsymbol{F}^i = \frac{\boldsymbol{F}^i}{\|\boldsymbol{F}^i\|}$;
        for $k = 1$ to $n$ do
            if $F^i_k > 0$ then
                $x^i_k = x^i_k + \lambda F^i_k(u_k - x^i_k)$;
            else
                $x^i_k = x^i_k + \lambda F^i_k(x^i_k - l_k)$;
            end if
        end for
    end if
end for
```

After finishing the above procedures, the positions of points are updated and we have finished one iteration calculation of EM. Take Fig. 2 for an example. There are three particles and their own objective values are 15, 10 and 5, respectively. Because Particle 1 is worse than Particle 3 while Particle 2 is better than Particle 3, Particle 1 represents a repulsion force which is $\boldsymbol{F}_{13}$ and Particle 2 encourages Particle 3 that moves to the neighborhood region of Particle 2. Consequently, Particle 3 moves along with the total force $\boldsymbol{F}$.

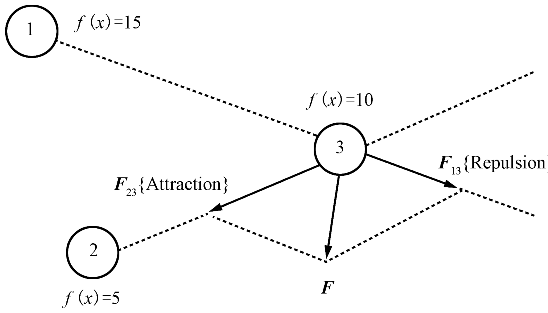
1
$f(x)$=15
$f(x)$=10
3
$\boldsymbol{F}_{13}${Repulsion}
$\boldsymbol{F}_{23}${Attraction}
2
$f(x)$=5
$\boldsymbol{F}$

Fig. 2 An example of attract-repulse effect on three particles

2.5 An inverse kinematic example of a general 3R manipulator

According to test results in [19], in general, 25 iterations per dimension are satisfactory for converging to the optimum point for moderately difficult functions. Therefore, an approximate solution vector which is near the true solution to the inverse kinematic problem of n-DOF robot manipulator can be obtained in finite iterations ($\leq 25n$). An example for solving the inverse kinematics problem of a general 3R manipulator has been used, as shown in Fig. 3. This example demonstrates a typical run of EM by presenting the positions of all particles in iterations.

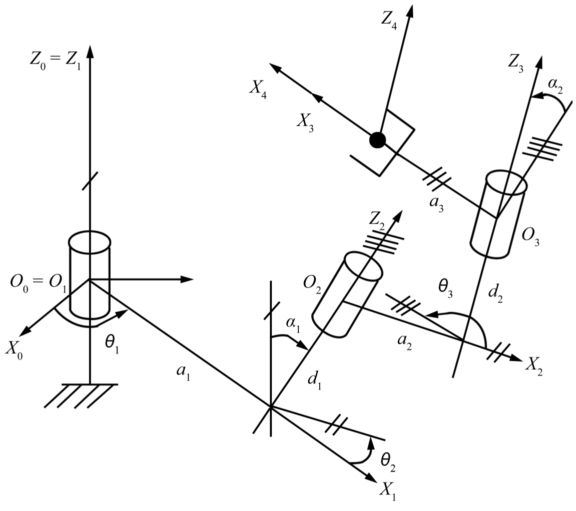

Fig. 3 A kinematic scheme for a general 3R manipulator

For this example, the Denavit-Hartenberg parameters are shown in Table 1. The joint limitations are: $\theta \in [-\frac{\pi}{2}, \pi]$. The desired configuration is: $\boldsymbol{P}_d$ = [0.0084, 2.2992, 1.2962]$^{\text{T}}$, $\boldsymbol{R}_d = [\boldsymbol{d}_1, \boldsymbol{d}_2, \boldsymbol{d}_3]$, where $\boldsymbol{d}_1$ = [−0.6863, −0.5636, 0.4598]$^{\text{T}}$, $\boldsymbol{d}_2$ = [0.4415, −0.8251, −0.3525]$^{\text{T}}$, and $\boldsymbol{d}_3$ = [0.578, −0.0389, 0.815]$^{\text{T}}$, which corresponds to an exact solution of $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3] = [60°, 70°, 80°]$. In this example we have selected the sample point m to be 20.

In the following figures, "$*$" represents the location of true solution and "$\diamondsuit$" shows the current best point. Fig. 4 shows the location of particles when the algorithm is started by randomly sampling points from the feasible region. The points in the population move towards the region around

Table 1 The link parameters of the 3R manipulator

Joint	Link length a (m)	Twist angle α (°)	Offset length d (m)	Rotation angle (°)
1	2.4354	27.0245	0.2513	θ_1
2	1.0826	15.7664	0.1509	θ_2
3	0.9748	0	0	θ_3

the current best point. Fig. 5 shows a new best point θ^{best} is observed and the points start to converge towards the new best point. Finally, the optimum solution is located as shown in Fig. 6. The detailed discussion about EM is given in Section 3.

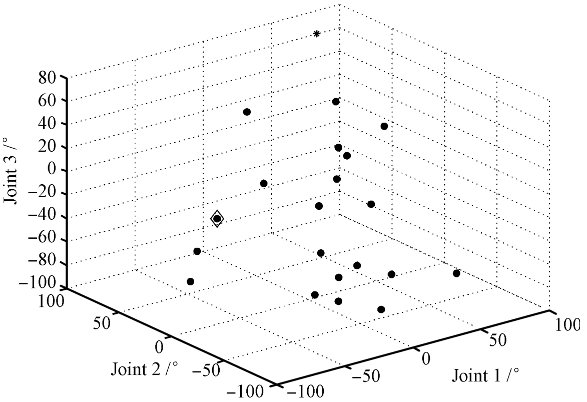

Fig. 4 The random initial positions of the particles

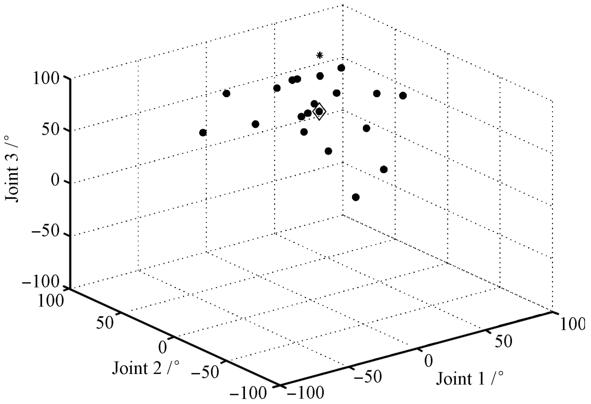

Fig. 5 Points start to attract and repel each other

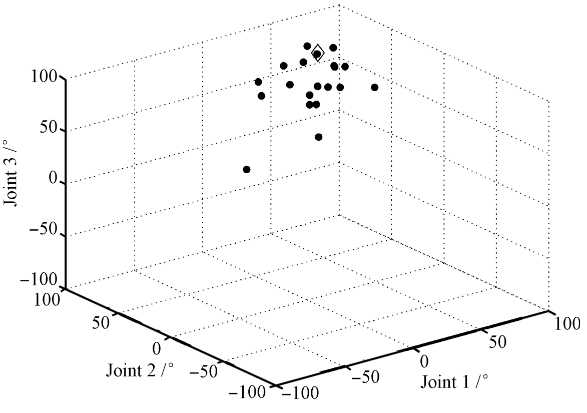

Fig. 6 One of the points finds the true solution

3 Hybrid methods using EM and modified DFP for solving the IK problem

Our experience shows that applying only a few iterations of EM will bring at least one sample point into the neighborhood of the true solution. In general, four results are usually obtained by EM:

Case 1. A true solution is obtained within a small number of iterations;

Case 2. A true solution is obtained within a desired number of iterations;

Case 3. A true solution is obtained when the number of iterations is large enough;

Case 4. An approximate solution close to the true solution is obtained within a desired number of iterations.

Without loss of generality, we will consider the most general Case 4. For Case 4, although the result can be improved by increasing the number of iterations, this may increase the running time. Notice that an approximate solution near to the true solution can be obtained rapidly by EM. On the other hand, when the solution vector is near the true solution, the DFP method is generally expected to give a rapid convergence rate[24]. Therefore, if we combine the above two optimization methods, this hybrid approach is more efficient and precise.

The DFP method finds the solution to the secant equation that is closest to the current estimate and satisfies the curvature condition (see below). It was the first quasi-Newton method which generalizes the secant method to a multidimensional problem. This update maintains the symmetry and positive definiteness of the Hessian matrix. Given a function $f(x)$, its gradient ∇f, and positive definite Hessian matrix B, the Taylor series is:

$$f(x_k + s_k) = f(x_k) + \nabla f(x_k)^{\mathrm{T}} s_k + \frac{1}{2} s_k^{\mathrm{T}} B s_k \tag{9}$$

And the Taylor series of the gradient itself is:

$$\nabla f(x_k + s_k) = \nabla f(x_k) + B s_k \tag{10}$$

which is used to update B. The DFP method finds a solution that is symmetric, positive, and closest to the current approximation of B_k:

$$B_{k+1} = (I - \gamma_k y_k s_k^{\mathrm{T}}) B_k (I - \gamma_k s_k y_k^{\mathrm{T}}) + \gamma_k y_k y_k^{\mathrm{T}} \tag{11}$$

where

$$y_k = \nabla f(x_k + s_k) - \nabla f(x_k), \quad \gamma_k = \frac{1}{y_k^{\mathrm{T}} s_k} \tag{12}$$

And B_k is a symmetric and positive definite matrix. The corresponding update to the inverse Hessian approximation $H_k = B_k^{-1}$ is given by:

$$H_{k+1} = H_k - \frac{H_k y_k y_k^{\mathrm{T}} H_k}{y_k^{\mathrm{T}} H_k y_k} + \frac{s_k s_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k} \tag{13}$$

B is assumed to be positive definite and the vectors s_k^{T} and y must satisfy the curvature condition:

$$s_k^{\mathrm{T}} y_k = s_k^{\mathrm{T}} B s_k > 0 \tag{14}$$

The DFP method is quite effective. However, due to the highly nonlinear nature of the manipulator kinematic equations, the gradient vector of the objective function and the optimum step size in the search direction are difficult to obtain in DFP. To solve these two problems, the analytical form of the gradient vector and an approximation to the optimum search step size had been developed in [11]. However, since the Jacobian matrix is used in [11], the computational process is more complex. To reduce the computational complexity, a modified DFP algorithm is developed in this paper. Unlike the traditional DFP algorithm, this modified DFP algorithm randomly chooses the search step size between 0 and 1. Thus, the computational complexity is greatly reduced, and more importantly, this method is not only computationally efficient but also numerically stable. In order to ensure that the modified DFP algorithm converges to the optimum solution, the following assumptions are made.

Assumption 1. The pre-specified initial value of the solution is close to the optimum solution and the search procedure is restricted to a small area near to the true solution in modified algorithm.

Assumption 2. The optimum search direction is known per iteration.

From the foregoing discussions, the initial guess value can be given by EM. The search direction in Assumption 2 can be obtained using an approximated Hessian matrix and the gradient vector of the objective function[24]. Based on the above assumptions, the true solution can be approached gradually using a small random search step size between 0 and 1. Since the search domain is small, the optimum solution can be efficiently calculated. The complete solution procedure of MDFP algorithm is summarized below:

```
Step 1. Given the initial value $x^{(1)} \in \mathbf{R}^n$ and termination criteria $\varepsilon > 0$;
Step 2. $H_1 = I_n$ (unit matrix);
Step 3. Given a random search step size: $\lambda = \text{rand}\,(0,1)$;
Step 4. Calculate the gradient at $x^{(1)} : g_1 = \nabla f(x^{(1)})$;
Step 5. $k = 1$;
Step 6. Calculate the search direction: $d^{(k)} = -H_k g_k$;
Step 7. $err = \|g_k\|$;
Step 8. While $err \geq \varepsilon$ do;
    Step 9. $x^{(k+1)} = x^{(k)} + \lambda d^{(k)}$;
    Step 10. $g_{k+1} = \nabla f(x^{(k+1)})$;
    Step 11. $q^{(k)} = g_{k+1} - g_k$;
    Step 12. $p^k = x^{k+1} - x^k$;
    Step 13. $H_{k+1} = H_k + \frac{p^{(k)} p^{(k)\mathrm{T}}}{p^{(k)\mathrm{T}} q^{(k)}} - \frac{H_k q^{(k)} q^{(k)\mathrm{T}} H_k}{q^{(k)\mathrm{T}} H_k q^{(k)}}$;
    Step 14. $d^{k+1} = -H_{k+1} g_{k+1}$;
    Step 15. $err = \|g_{k+1}\|$;
    Step 16. $k = k + 1$;
Step 17. End while
```

Lemma 1[24]**.** If the gradient $g_i \neq 0$, the matrix H_i (Step 13 in MDFP algorithm) is symmetric positive definite.

Lemma 2[24]**.** For a symmetric positive definite matrix H_i, the search direction vector:

$$\boldsymbol{d}^{(k)} = -H_k \nabla f(x^{(k)}) \tag{15}$$

is a descent direction. It shows that a small step along $\boldsymbol{d}^{(k)}$ can guarantee that the objective function is reduced. Thus, the global optimum solution can be approached by an appropriate small step in the direction of descent with limited iteration.

3.1 Testing the convergence properties of EM-MDFP algorithm

In order to examine the basic convergence properties of EM-MDFP algorithm, general 10 test functions taken from the web site, http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedarfiles/go.htm, are calculated by EM-MDFP method.

First, we make use of EM to obtain an approximate solution near to the true solution. Though better results can be achieved by EM when a large number of iterations and sample points (m) are used, we have selected a small number of sample points (m) and maximum iterations aim to reduce the running time and examine whether the MDPF algorithm can converge to the optimum solution when the quality of initial value given by EM is not good enough. In this paper, we have selected the sample points (m) to be 10 and the maximum iterations to be 30, except function $Trid$ and *Michalewics* (5). The number of 30 iterations seems to be not large enough to adequately explore a feasible solution for these two problems. Then, we have selected the maximum iterations to be 100 to calculate function $Trid$ and $Michalewics$ (5). The running time, approximate solution ($x(1)$) and the best objective function values (Best $f\,(x)$) are recorded. The testing procedure continuously runs 10 times. The average running time (Avg. time) and objective function values (Avg. $f(x)$) are also reported. Secondly, the MDFP method starts from the initial point given by EM to search the optimum solution. The stopping criterion is set to be 10^{-6}. For the comparison, number of iterations, running time, and objective function, including their average values, are recorded.

All the computations are conducted on a Celeron (R) CPU 2.80 GHz PC. The algorithm is coded in Matlab 6.5. Table 2 shows that the test results of the test function *booth*. The known optimum solution and minimum value of test function *booth* are, respectively, $x^* = (1, 3)$ and $f(x^*) = 0.0$. In this example, the search step size λ is assumed to be uniformly distributed between 0 and 1. More functions test results are given in Table 3 and the process of testing is similar to Table 2. Without loss of generality, the search step size λ is randomly set to be a small positive value between 0 and 1. Our results show that EM is able to approximate the optimum. Moreover, although the search step size λ is not optimal, the convergence is extremely rapid and the optimum solutions in all cases can be obtained by MDFP. Function $Michalewics$ in Table 3 appears to be a special case of taking more iterations by MDFP. It is well known that the global optimum of function $Michalewics$ is hard to be obtained. To get a better convergence property, a given optimal search step size is used instead of choosing randomly in MDFP. The global optimum is obtained successfully and the running times of MDFP are, respectively, only about 0.08 and 0.6 s. Notice that the running time is inevitable to be disturbed by other programs running in PC, the actual time consumption of MDFP method is almost close to 0. This is also why some iterative procedures with less number of iterations need more running time in Tables 2 and 3.

3.2 Inverse kinematic problems of PUMA 560

The structure of the manipulator used in this example is based on the PUMA 560 robot. This robot is chosen beca-

Table 2 Results of EM-MDFP for test function *booth*

Solution	EM			MDFP			
number	$x^{(1)}$	$f(x^{(1)})$	Time (s)	$f(x^*)$	Time (s)	Steps	λ
1	(1.3843, 2.5867)	0.3219	0.3130	0.0000	0.0780	130	0.1
2	(1.2745, 2.8634)	0.1701	0.3130	0.0000	0.0160	70	0.2
3	(1.2524, 2.7525)	0.1251	0.3130	0.0000	0.0160	42	0.3
4	(1.0923, 2.9118)	0.0164	0.3430	0.0000	0.0150	28	0.4
5	(1.1791, 2.8806)	0.0606	0.3280	0.0000	0.0320	24	0.5
6	(1.1484, 2.7962)	0.0758	0.3440	0.0000	0.0000	18	0.6
7	(0.9709, 3.1807)	0.1254	0.3120	0.0000	0.0000	15	0.7
8	(0.6474, 3.4850)	0.4297	0.3130	0.0000	0.0160	12	0.8
9	(1.4567, 2.6599)	0.3786	0.3280	0.0000	0.0310	9	0.9
10	(1.1880, 2.9396)	0.1042	0.3440	0.0000	0.0150	5	1.0
Avg.		0.1808	0.3250	0.0000	0.0219	35	

Table 3 Results of EM-MDFP for all test functions

Function	Known	EM			MDFP			
name	optimum	Best $f(x)$	Avg. $f(x)$	Avg.time (s)	Avg. $f(x)$	Avg.time (s)	Avg.steps	λ
Beale	0.0	0.0121	0.0650	0.3141	0.0000	0.0108	52	random
Hump	0.0	0.0000	0.0403	0.5311	0.0000	0.0281	34	random
Matyas	0.0	0.0150	0.1917	0.3263	0.0000	0.0295	60	random
Sum squares (5)	0.0	0.3935	1.3336	0.3281	0.0000	0.0138	23	random
Zakharov (2)	0.0	0.0030	0.0686	0.3166	0.0000	0.0266	26	random
Levy (2)	0.0	0.0004	0.1783	0.3391	0.0000	0.0231	22	random
Trid (6)	−50.0	−47.2770	−43.7232	0.9302	−50.00	0.0201	28	random
Michalewics (2)	−1.8013	−1.8003	−1.7571	0.3469	−1.8013	0.0876	289	0.05
Michalewics (5)	−4.6876	−4.2910	−3.7141	1.0453	−4.6863	0.5967	2 776	0.008

use a closed-form solution can be obtained, which can be used for checking the accuracy of the numerical solution. The link parameters and the joint limits of this robot are listed in Table 4.

To handle the IK problem of this robot with MDFP method, the search direction must be calculated firstly, which can be obtained by using an approximated Hessian matrix and the gradient vector of the objective function. This Hessian matrix can be computed using the formula (Step 13 in MDFP algorithm). For (3), the elements of the gradient vector are given by following equation[11]:

If the joint is a translational joint, then

$$\frac{\partial e(\theta)}{\partial \theta_i} = 2\boldsymbol{z_i} \cdot (\boldsymbol{P_h}(\theta) - \boldsymbol{P_d}) \tag{16}$$

If the joint is a rotational joint, then

$$\frac{\partial e(\theta)}{\partial \theta_i} = 2\boldsymbol{z_i} \cdot \Big\{[\boldsymbol{P_d} - \boldsymbol{P_h}] \times \boldsymbol{P_{ih}} + \sum_{j=1}^{3}(\boldsymbol{d}_j \cdot \boldsymbol{h}_j - 1)(\boldsymbol{h}_j \times \boldsymbol{d}_j)\Big\} \tag{17}$$

Given θ, $\boldsymbol{z_i}$, $\boldsymbol{P_h}$, and $\boldsymbol{h}_j$ $(j = 1, 2, 3)$ can be recursively computed by using the forward recursion formulas in [25]. Consequently, the search direction vector can be efficiently evaluated.

Example 1. The convergence criterion of EM in this example is defined by $\varepsilon = 0.1$ instead of the maximum iterations. The convergence of MDFP method is defined by $\varepsilon = 10^{-k}$, where k varies from 1 to 8. For each value of k, 10 problems were given to EM-MDFP method to solve. Both the initial and the desired configurations of the problems were randomly generated within the working space of the robot. The average running times of the solved problems are plotted in Fig. 7. From this figure, the EM-MDFP method is able to solve all of the given problems at the desired precision. Moreover, even though the required degree of precision is high, the EM-MDFP is also computationally efficient. It should be noted that a C implementation would be much faster than Matlab. Then, we have reasons to believe that the EM-MDFP is a near real time algorithm if the evaluations are carried out in C/C ++ environment.

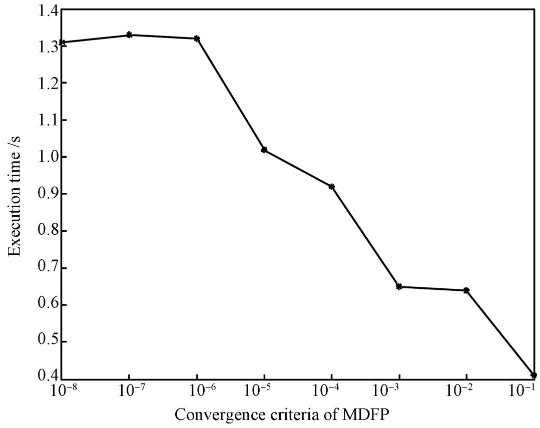

Fig. 7 Average execution time for Example 1

Example 2. This example is used to demonstrate that the EM-MDFP method has the potential of finding all of the multiple solutions. As we know that the PUMA 560 has at most eight solutions when there are no joint limits. In order to find all solutions, the joint limitations were released in this example. The desired configuration of the end effector was given by: $\boldsymbol{P}_d = [0.732, 0.386, 0.682]^{\mathrm{T}}$, $\boldsymbol{d}_1 = [-0.727, 0.646, 0.233]^{\mathrm{T}}$, $\boldsymbol{d}_2 = [0.317, 0.014, 0.948]^{\mathrm{T}}$, and $\boldsymbol{d}_3 = [0.610, 0.763, -0.215]^{\mathrm{T}}$, which corresponds to an exact solution of $\boldsymbol{\theta} = [15°, 25°, 35°, 45°, 55°, 65°]^{\mathrm{T}}$. The convergence criteria of EM and MDFP are, respectively, equal to 0.1 and 10^{-6}. And the number of sample points in EM are 10. For convenience, the algorithm had been modified to run iteratively. This process is very easy to implement by using a loop command. Given a maximum number of evaluations, the program will run continuously and the results of each iteration are recorded. In this example, all eight solutions were found successfully after 75 iterations, as shown in Table 5.

The following plots (Figs. 8 ~ 11) summarize the performance of EM-MDFP algorithm with different search steps to solve the inverse kinematics of PUMA robot. In this example, four representative search step sizes, $\lambda = 0.01, 0.1, 0.5, 1.2$, respectively, are chosen to represent the convergence properties of EM-MDFP. Our results show that the EM-MDFP is able to approximate the optimum solution at the desired precision in all cases. As seen in the plots, the error plot is smoother for MDFP with smaller search step size. But the convergence properties of MDFP with smaller search size, in general, are worse than the one with large search step size. However, if we set the search step size $\lambda > 2$, the MDFP do not converge. The performance of the MDFP developed in this paper is very similar to the performance of the steepest descent method. Similarly, the parameter λ in MDFP can be called as learning speed in this paper.

4 Discussions and conclusions

In this paper, we have developed a novel hybrid algorithm, called EM-MDFP, which is a powerful and easy algorithm for solving the inverse kinematics problem of robot manipulators. The major advantages of EM are that its complexity is independent on the characteristics of the ki-

Table 4 The link parameters of the PUMA 560 robot

Joint	Link length (m)	Twist angle (°)	Offset length (m)	Joint limitations (°)
1	0	−90	0.6604	[−160, 160]
2	0.4320	0	0.2000	[−225, 45]
3	0	90	−0.0505	[−45, 225]
4	0	−90	0.4320	[−110, 170]
5	0	90	0.0	[−100, 100]
6	0	0	0.0565	[−266, 266]

Table 5 The multiple solutions of Example 2

Solution number	θ_1 (°)	θ_2 (°)	θ_3 (°)	θ_4 (°)	θ_5 (°)	θ_6 (°)
1	14.9981	25.0043	34.9991	45.0663	54.9701	65.0246
2	14.9943	25.0045	35.0040	−134.8443	−54.9786	−115.0963
3	14.9991	−29.9968	144.9999	103.3375	36.5176	−11.5898
4	14.9989	−29.9990	144.9972	−76.6554	−36.5255	168.3681
5	−142.5264	−149.4014	34.3634	130.9456	−15.9070	148.1551
6	−142.9029	−149.9997	35.0004	−49.5765	18.4455	−31.2045
7	−142.9026	155.0002	145.0002	20.0620	−44.5842	−93.8781
8	−142.9034	155.0008	144.9990	−159.9225	44.5867	86.1114

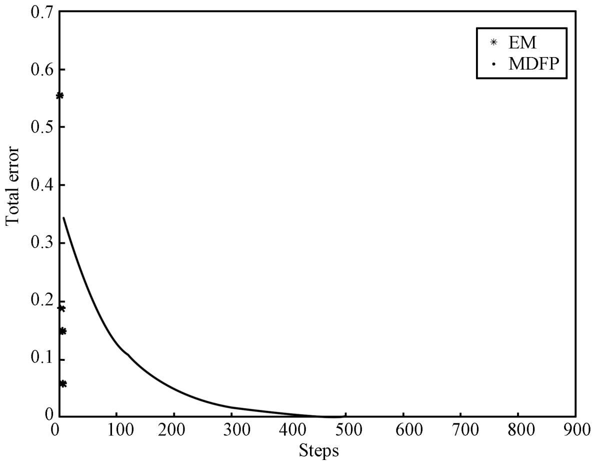

Fig. 8 Performance of the EM-MDFP when $\lambda = 0.01$

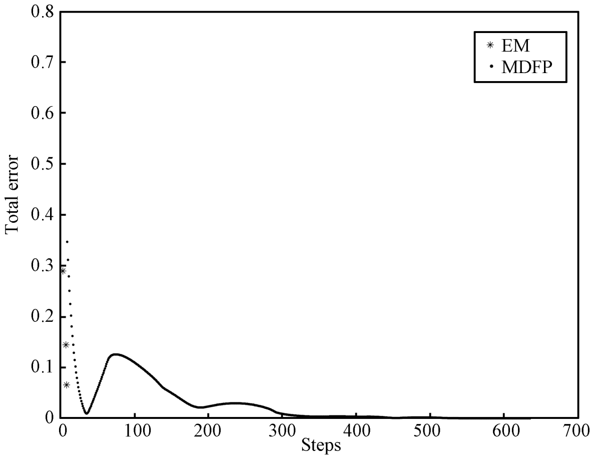

Fig. 9 Performance of the EM-MDFP when $\lambda = 0.1$

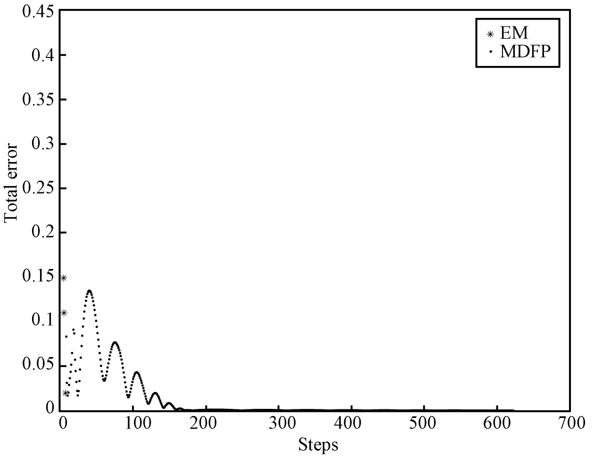

Fig. 10 Performance of the EM-MDFP when $\lambda = 0.5$

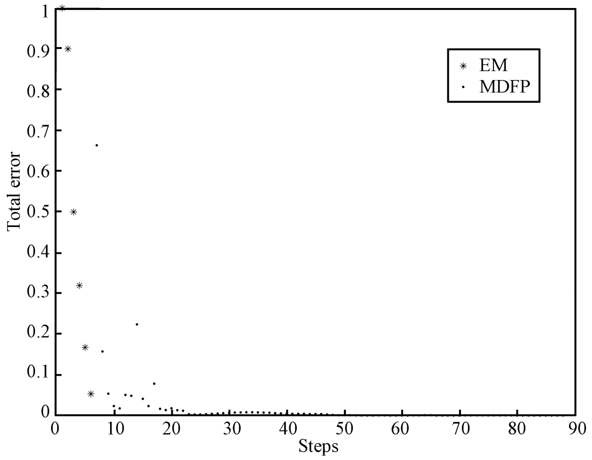

Fig. 11 Performance of the EM-MDFP when $\lambda = 1.2$

nematic equations involving dimensionality and the degree of nonlinearity, and it is not sensitive to the initial and singular configurations of the manipulator. Without using the first or second order information, EM is able to converge rapidly to an approximate solution. Starting from the initial point given by EM, a modified variable metric method denoted by MDFP is used to search the optimum solution at the desired precision. Since the size step in search can be randomly chosen, the computational complexity is greatly reduced. The experimental results show that this hybrid method is not only numerically stable but also computationally efficient. According to the test results, some additional comments are:

1) EM is efficient enough to obtain IK solutions independent of robot geometry and the number of degrees of freedom. However, when the required degree of precision is high, it appears to be more suitable for off-line calculation, rather than on-line calculation;

2) Since the EM-MDFP fully exploits the strength of both the EM and the MDFP method, it is computationally efficient and very suitable for on-line calculation;

3) Though the search step sizes can be randomly chosen in MDFP, it should be noted that performance for different step size varies, especially for some difficult functions. In order to get a better convergence performance for a particular function, a pre-specified optimal search step size is used, which is easily determined by using experimental method.

References

1 Manseur R, Doty K L. Structural kinematics of 6-revolute-axis robot manipulators. *Mechanism and Machine Theory*, 1996, **31**(5): 647−657

2 Raghavan M, Roth B. Inverse kinematics of the general 6R manipulator and related linkages. *Journal of Mechanical Design*, 1993, **115**(3): 502−508

3 Raghavan M, Roth B. Kinematic analysis of the 6R manipulator of general geometry. In: Proceedings of the 5th International Symposium on Robotics Research. Tokyo, Japan: MIT Press, 1991. 263-269

4 Manocha D, Canny J F. Efficient inverse kinematics for general 6R manipulators. *IEEE Transaction on Robotics and Automation*, 1994, **10**(5): 648−657

5 Husty M L, Pfurner M, Schrocker H P. A new and efficient algorithm for the inverse kinematics of a general serial 6R manipulator. *Mechanism and Machine Theory*, 2007, **42**(1): 66−81

6 Qiao S G, Liao Q Z, Wei S M, Su H J. Inverse kinematic analysis of the general 6R serial manipulators based on double quaternions. *Mechanism and Machine Theory*, 2010, **45**(2): 193−199

7 Manseur R, Doty K L. Fast inverse kinematics of 5-revolute-axis robot manipulators. *Mechanism and Machine Theory*, 1992, **27**(5): 587−597

8 Thomopoulos S C A, Tam R Y J. An iterative solution to the inverse kinematics of robotic manipulators. *Mechanism and Machine Theory*, 1991, **26**(4): 359−373

9 Cheng H, Gupta K C. A study of robot inverse kinematics based upon the solution of differential equations. *Journal of Robotic Systems*, 2007, **8**(2): 159−175

10 Chen I M, Yang G L, Kang I G. Numerical inverse kinematics for modular reconfigurable robots. *Journal of Robotic Systems*, 1999, **16**(4): 213−225

11 Wang L C T, Chen C C. A combined optimization method for solving the inverse kinematics problem of mechanical manipulators. *IEEE Transactions on Robotics and Automation*, 1991, **7**(4): 489−499

12 Hasan A T, Hamouda A M S, Ismail N, Al-Assadi H M A A. An adaptive-learning algorithm to solve the inverse kinematics problem of a 6 DOF serial robot. *Advances in Engineering Software*, 2006, **37**(7): 432−438

13 AI-Faiz M Z. Inverse kinematics solution for robot manipulator based on neural network. *MASAUM Journal of Basic and Applied Sciences*, 2009, **1**(2): 147−154

14 Koker R, Oz C, Cakar T, Ekiz H. A study of neural network based inverse kinematics solution for a three-joint robot. *Robotics and Autonomous Systems*, 2004, **49**(3−4): 227−234

15 Chiddarwar S S, Babu N R. Comparison of RBF and MLP neural networks to solve inverse kinematic problem for 6R serial robot by a fusion approach. *Engineering Applications of Artificial Intelligence*, 2010, **23**(7): 1083−1092

16 Oyama E, Agah A, MacDorman K F, Maeda T, Tachi S. A modular neural network architecture for inverse kinematics model learning. *Neurocomputing*, 2001, **38−40**: 797−805

17 Alavandar S, Nigam M J. Neuro-fuzzy based approach for inverse kinematics solution of industrial robot manipulators. *International Journal of Computers, Communications & Control*, 2008, **3**(3): 224−234

18 Karlra P, Prakash N R. A neuro-genetic algorithm approach for solving the inverse kinematics of robotic manipulators. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics. Washington D. C., USA: IEEE, 2003. 1979−1984

19 Birbil S I, Fang S C. An electromagnetism-like mechanism for global optimization. *Journal of Global Optimization*, 2003, **25**(3): 263−282

20 Birbil S I, Fang S C, Sheu R L. On the convergence of a population-based global optimization algorithm. *Journal of Global Optimization*, 2004, **30**(2−3): 301−318

21 Yurtkuran A, Emel E. A new hybrid electromagnetism-like algorithm for capacitated vehicle routing problems. *Expert Systems with Applications*, 2010, **37**(4): 3427−3433

22 Tsou C S, Kao C H. An electromagnetism-like meta-heuristic for multi-objective optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation. Vancouver, Canada: IEEE, 2006.1172−1178

23 Wu P, Yang K J, Fang H C. A revised EM-like algorithm +K-OPT method for solving the traveling salesman problem. In: Proceedings of the 1st International Conference on Innovative Computing, Information and Control. Beijing, China: IEEE, 2006. 546−549

24 Chen Bao-Lin. *Optimization Theory and Algorithms (Second Edition)*. Beijing, China: Tsinghua University Press, 2005. 375−376 (in Chinese)

25 Wang L, Ravani B. Recursive computations of kinematic and dynamic equations for mechanical manipulators. *IEEE Journal of Robotics and Automation*, 1985, **1**(3): 124−131

YIN Feng Ph. D. candidate at the College of Electrical and Information Engineering, Hunan University. He received his master degree from Hunan University in 2008. His research interest covers intelligent control and robotics. Corresponding author of this paper.
E-mail: yinfeng83@126.com

WANG Yao-Nan Professor at the College of Electrical and Information Engineering, Hunan University. He received his bachelor degree in computer engineering from East China Technology Institute (ECTI) in 1981, and master and Ph. D. degrees in control engineering from Hunan University in 1990 and 1994, respectively. From 1981 to 1994, he was with ECTI. Since 1995, he has been a professor at the College of Electrical and Information Engineering, Hunan University. His research interest covers industrial process control, intelligent control theory and applications, and image processing. E-mail: yaonan@hnu.cn

WEI Shu-Ning Ph. D. candidate at the College of Electrical and Information Engineering, Hunan University. Her research interest covers robot control technology, neural network control, and machine learning. E-mail: weishuning@sina.com