Journal of Applied Sciences

ISSN 1812-5654

Journal of Applied Sciences 13 (13): 2366-2371, 2013
ISSN 1812-5654 / DOI: 10.3923/jas.2013.2366.2371

An Algebraic Inverse Method of 7-DoF Manipulator Redundant Robots

Shuai Guo, Cheng Hongjing and Chunsheng Xie
Department of Mechanical Engineering and Automation,
Shanghai University, 200072 SHU, Shanghai, China

Abstract: This study puts forward an algebraic method of 7-DoF manipulator redundant arms with strong real-time and a light amount of calculation. It also can map any posture in the workspace. The essence of the algorithm is that by increasing the amount of known quantities. The method establishes numerical mapping relations between the operation space and the joint space, optimizes the singular form and time overhead. Based on this algorithm, 7-Dofmanipulator redundant arm can be controlled in PTP motion control.

Key words: 7-DoF Redundant manipulator, Algebraic algorithm, kinematics

INTRODUCTION

Real-time kinematics algorithm is the key for humanoid robot motion performance. At present, the development of the kinematics algorithms which are below the sixth degree of freedom becomes more and more mature. However, most of the humanoid robots use the seventh degrees of freedom arms. Compared with the sixth degrees of freedom arm, the 7-Dof manipulator is more flexibility and useful. Because most postures have many kinematic solutions, it is very hard to solve its kinematics and it can avoid the situation such as some joint move too quickly. Therefore, it can satisfy the needs for the demands of more complex movement control.

There are several kinematics algorithms for 7-DoF redundant robot, such as iterative algorithm, least squares algorithm, the position and orientation separation algorithm and so on. Due to presence of redundant DOF, kinematics algorithm is complex, large computational, time-consuming and it affects the real-time. Paper (Han *et al*., 2011) submitted a pose separation method whose essence is to add condition-speed of terminal coordinate system but actually it's difficult to be given. Paper (Singh and Claasens, 2010) proposed a weighted least squares algorithm based on SVD decomposition to solve the kinematics problem. But the real-time of this method is poor and it needs the special requirements for arm structures. Paper (Kalra *et al*., 2006) proposed a twice calculation: firstly, use the gradient projection method; secondly, use fixed joint method to calculate. This method is convergent but still has a large error. Although it can cut 150ìs, it can't satisfy the needs of the calculation. Based on the separation velocity method, paper (Klanke *et al*., 2006) solves the problem of the humanoid arm self-motion. The essence of this method is using velocity integral to find the displacement but it has large calculation errors. Paper (Alqasemi and Dubey, 2007) proposed an analytical solution method. But it servers for WMA(whole arm manipulator) whose structure is very special. This method doesn't have universal applicability.

This thesis uses an algebraic algorithm to solve kinematics of 7-DoF redundant manipulator by adding an additional condition that Joint 2 must be near the target. It can solve the problem that how to make the additional condition sure. Besides by processing singular configuration and algorithm optimization, it can shorten the time and improve the accuracy. So that it can meet the requirement of the real-time.

PARAMETERS OF MANIPULATOR

This study uses 7-DOF WLA (Weight Light Arm) Humanoid robot arm (Fig. 1) of SCHUNK Company as Experimental Platform. The arm (Fig. 2) consists of two PRL120 rotating module, two PRL100 rotating module, two PRL 80 rotating module, one PRL 60 rotating module and connectors. Parameters of manipulator are shown in Table 1.

As the Fig. 3 shows, it simulates the work space of experimental platform. From the Fig. 3, it's easy to find that the experimental platform has a huge work space and it has several corresponding joints for a posture. Therefore, how to establish the mapping relationship between operation space and Working space is the key of solving inverse kinematics problem for7-DoF manipulator redundant robots.

Corresponding Author: Shuai Guo, Department of Mechanical Engineering and Automation, Shanghai University, 200072 SHU, Shanghai, China

Fig. 1: Plateform

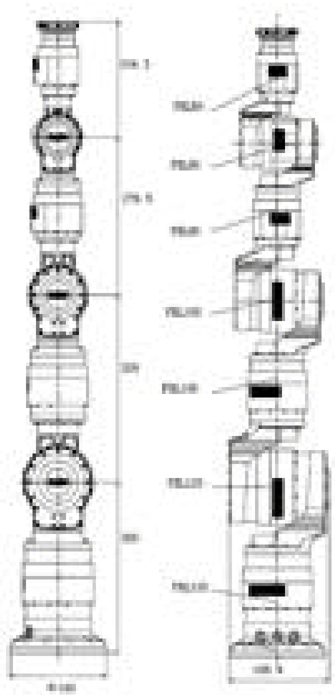

Fig. 2: Structure

PROCEDURE OF THE ALGEBRAIC ALGORITHM

Mathematics description: In fact, inverse kinematic of the manipulator is to establish the mapping relationship

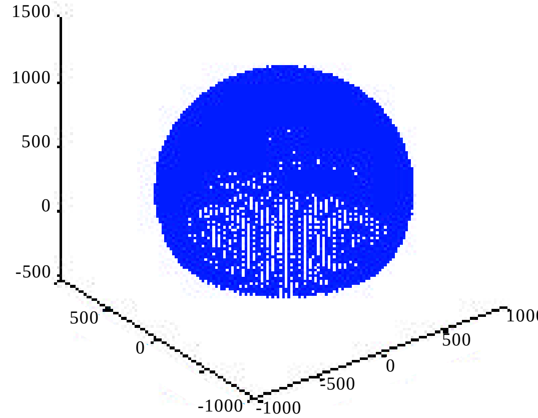

Fig. 3: Work space

Table 1: Joins Parameter

ID	Range	Max Vel (°/s)	Max Acc (°/s2)	P	I	D
1	-180~180°	40	160	32	150	4
2	-120~120°	40	160	32	150	4
3	-180~180°	40	160	48	150	2
4	-140~140°	40	160	48	150	2
5	-180~180°	40	160	56	180	6
6	-160~160°	40	160	56	180	6
7	-180~180°	40	160	32	160	3

between Operating space and Joint space. Operating space is usually expressed as six variable quantities, three euler angles (o, a, t) and three position value (x, y, z). While Joint space is expressed by each joint angle values, such as the value of joint i is θ_i.

As the rule (Youlun *et al*., 2008) shows, operating space of 7-DoF manipulator redundant arms consisted of six variable quantities while Operating space consist of seven joint angles. Therefore, this kind of equation which contains seven unknowns can't get the specific solution. It only can find the relationship between these unknowns.

Pieper standard (Youlun *et al*., 2008) points out with one of the following two sufficient conditions, the robot arm can have closed-form solution: (1) Three adjacent joint axes cross at one point, (2) Three adjacent joint axes parallel to each other. The paper used to meet condition 1, so it must have a closed form solution.

Idea of algorithm is by increasing the constraint of mechanical arm, such as speed of terminal velocity, direction of terminal velocity and so on to establish the mapping relationship between the joint space and the operation space:

$$\begin{cases} f_1(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7) = x \\ f_2(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7) = y \\ f_3(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7) = z \\ f_4(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7) = o \\ f_5(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7) = a \\ f_6(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7) = t \end{cases} \quad (1)$$

According to the current position of the joint 2, this thesis solve it near the current position of the Joint 2. It means that add variable at the velocity position posture $n\theta_2$ (the value is the value of joint 2 near the current location). In this way, t he number of variables in Joint space equal to the number of variables in working space,then it can continue solving (as the relationship shows in 2):

$$\begin{Bmatrix} \theta_2 = n\theta_2 \\ f_1(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7) = x \\ f_2(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7) = y \\ f_3(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7) = z \\ f_4(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7) = o \\ f_5(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7) = a \\ f_6(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7) = t \end{Bmatrix} \tag{2}$$

It also can be solve by adding other variables to establish the mapping between the operating space and joint space. For example, we can increase the variable and vector of the terminal velocity. As auxiliary conditions that increase the terminal velocity, a equation would be added during the solving process:

$$\dot{px}^2 + \dot{py}^2 + \dot{pz}^2 = v$$

(v is a known quantity). For the whole equation, it increases not only the integral item but also the squared, that makes the whole equation set become a more complicated transcendental equations and make the solve process become more complicated. By Increasing the direction of terminal velocity-vector P = [px, py, pz] as auxiliary condition, we can use the position method in paper (Han *et al*., 2011) to solve. But it is difficult for the auxiliary condition to be given. As auxiliary conditions that increase target location of joint 2 near the current location, actually, the solution procedure adds two linear equations: (1) abs $(\theta_2-\theta_{2c})\leq$max; (2)$f(\theta_2)\leq 1$. This auxiliary condition doesn't make the procedure complicated and the program is easy. But it is difficult to determine the second equation.

Solving procedure: Logical procedure of solving 7-DoF redundant manipulator as Fig. 4:

Solving θ_4: Firstly, build a coordinate system as Fig. 5. Secondly, use Matlab to calculate forward kinematic and find the mapping relationship. From Table 1, if $\cos(\theta)$ and $\sin(\theta)$ are known, θ is determined uniquely ; if only $\cos(\theta)$ is known, θ just has two possible solutions. This set will bring great convenience for the control and solution:

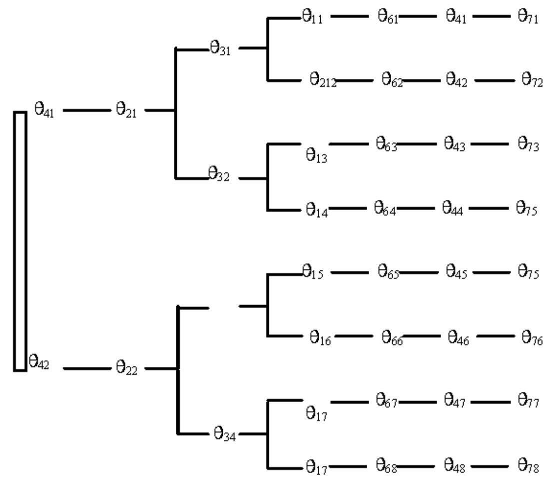

Fig. 4: Logical procedure

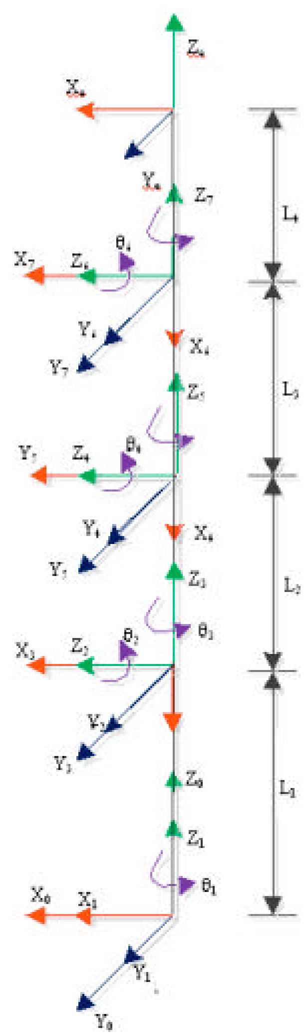

Fig. 5: Joint coordinate system

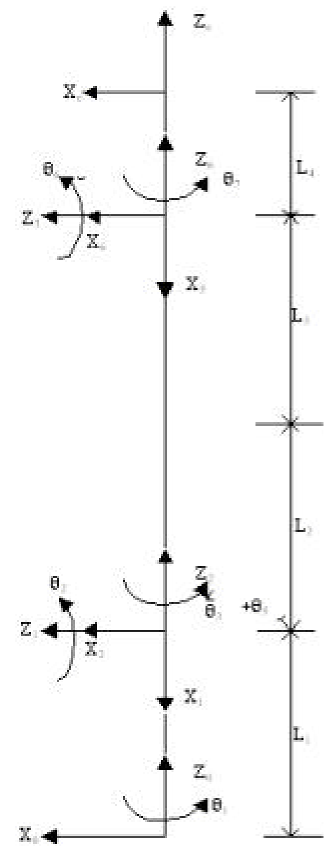

Fig. 6: $\theta_4 = 0$

$$
\begin{aligned}
T_{01} &= R_z(\theta_1)*T_z(L_1)*R_y(90)\\
T_{12} &= R_z(\theta_2)*R_y(-90)\\
T_{23} &= R_z(\theta_3)*T_z(L_2)*R_y(90)\\
T_{34} &= R_z(\theta_4)*R_y(-90)\\
T_{45} &= R_z(\theta_5)*T_z(L_3)*R_y(90)\\
T_{56} &= R_z(\theta_6)*R_y(-90)\\
T_{6e} &= R_z(\theta_7)*T_z(L_4)\\
T_{0e} &= T_{01}*T_{12}*T_{23}*T_{34}*T_{45}*T_{56}*T_{6e}\\
f(T_{0e}) &= [x \quad y \quad z \quad o \quad a \quad t]
\end{aligned}
\tag{3}
$$

According to the rules of homogeneous transformation, the mapping relationship between the joint space and operation space can be expressed: $Rz(\theta)$ rotates axis z of angleθ; $Ry(\theta)$ rotates axis y of angle θ; $Tz(L)$ transforms axis z of displacement L. Forward kinematic as follows:

$$(C_1 = \cos(\theta_1),\ S_1 = \sin(\theta_1))$$

$$a_x = T_{0e}(1,3) = C_6(S_4(C_1S_3+C_2C_3S_1)+C_4S_1S_2)+ S_6(C_5(C_4(C_1S_3+C_2C_3S_1)-S_1S_2S_4)+S_5(C_1C_3-C_2S_1S_3))$$

$$a_y = T_{0e}(2,3) = C_6(S_4(S_1S_3-C_1C_2C_3)-C_1C_4S_2)+ S_6(C_5(C_4(S_1S_3-C_1C_2C_3)+C_1S_2S_4)+S_5(C_3S_1+C_1C_2S_3))$$

$$a_z = T_{0e}(3,3) = C_6(C_2C_4-C_3S_2S_4)- S_6(C_5(C_2S_4+C_3C_4S_2)-S_2S_3S_5)$$

$$p_x = T_{0e}(1,4) = L_4(C_6(S_4(C_1S_3+C_2C_3S_1)+C_4S_1S_2)+ S_6(C_5(C_4(C_1S_3+C_2C_3S_1)-S_1S_2S_4)+ S_5(C_1C_3-C_2S_1S_3)))+L_3(S_4(C_1S_3+C_2C_3S_1)+C_4S_1S_2)+L_2S_1S_2$$

$$p_y = T_{0e}(2,4) = L_4(C_6(S_4(S_1S_3-C_1C_2C_3)-C_1C_4S_2)+ S_6(C_5(C_4(S_1S_3-C_1C_2C_3)+C_1S_2S_4)+S_5(C_3S_1+C_1C_2S_3)))+ L_3(S_4(S_1S_3-C_1C_2C_3)-C_1C_4S_2)-L_2C_1S_2$$

$$p_z = T_{0e}(3,4) = L_1+L_3(C_2C_4-C_3S_2S_4)- L_4(S_6(C_5(C_2S_4+C_3C_4S_2)-S_2S_3S_5)-C_6(C_2C_4-C_3S_2S_4))+L_2C_2$$

$$pA_x = p_x-L_4a_x = L_3(S_4(C_1S_3+C_2C_3S_1)+C_4S_1S_2)+L_2S_1S_2$$

$$pA_y = p_y-L_4a_y = L_3(S_4(S_1S_3-C_1C_2C_3)-C_1C_4S_2)-L_2C_1S_2$$

$$pA_z = p_z-L_1-L_4a_z = L_3(C_2C_4-C_3S_2S_4)+L_2C_2$$

Calculating Total = $pA_x^2+pA_y^2+pA_z^2$ could find:

$$\text{Total} = L_2^2+2\,L_2L_3C_4+L_3^2 \tag{4}$$

Using Eq. 4 solves θ_4.When θ_4 is equal to zero (Fig. 6), manipulator is singular. It only can calculate theta$\theta_3+\theta_5$. The coordinate system is shown in Fig. 6

Solving θ_2

Additional conditions: θ_2 is near joint 2 current location and search the entire space. It can find out whether 7-DoF manipulator redundant arm has the solution or out of the work place.

Find the solvability condition of θ_2, according to the following formula:

$$pA_x = p_x-L_4a_x = L_3(S_4(C_1S_3+C_2C_3S_1)+C_4S_1S_2)+L_2S_1S_2$$

$$pA_y = p_y-L_4a_y = L_3(S_4(S_1S_3-C_1C_2C_3)-C_1C_4S_2)-L_2C_1S_2$$

$$pA_z = p_z-L_1-L_4a_z = L_3(C_2C_4-C_3S_2S_4)+L_2C_2$$

From the formula above, we can get the solvability condition of θ_2:

$$C_3 = -(pA_z-(L_3C_2C_4+L_2C_2))/(L_3S_2S_4) \tag{5}$$

$$S_3 = \pm\sqrt{1-C_3^2} \tag{6}$$

$$S_1 = (L_3(pA_xC_4S_2+pA_yS_3S_4+pAxC_2C_3S_4)+pA_xL_2S_2)/ (-L_2^2C_2^2+L_2^2-2L_2L_3C_2^2C_4+2S_2S_4L_2L_3C_2C_3+2 L_2L_3C_4-L_3^2C_2^2C_3^2C_4^2+L_3^2C_2^2C_3^2-L_3^2C_2^2C_4^2+ 2S_2S_4L_3^2C_2C_3C_4+L_3^2C_3^2C_4^2-L_3^2C_3^2+L_3^2) \tag{7}$$

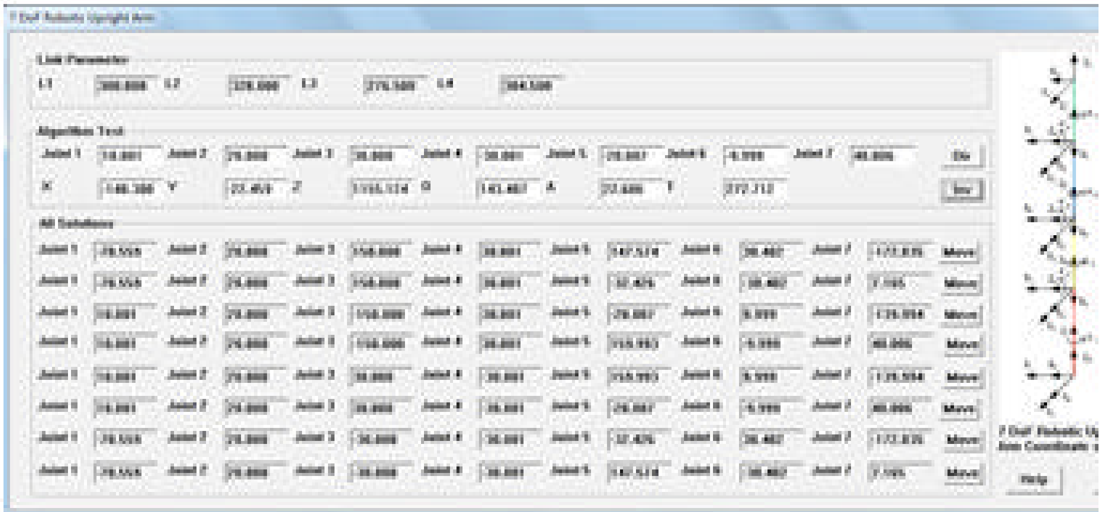

Fig. 7: Algorithm verification interface

Table 2: Singularity table

State of joints	Result
$\theta_2 = 0, \theta_4 = 0, \theta_6 = 0$	Result = $\theta_1+\theta_3+\theta_5+\theta_7$
$\theta_2 = 0, \theta_4 = 0, \theta_6 \neq 0$	Result = $\theta_1+\theta_3+\theta_5$
$\theta_2 = 0, \theta_4 \neq 0, \theta_6 \neq 0$	Result = $\theta_1+\theta_3$
$\theta_2 \neq 0, \theta_4 = 0, \theta_6 = 0$	Result = $\theta_3+\theta_5+\theta_7$
$\theta_2 \neq 0, \theta_4 \neq 0, \theta_6 = 0$	Result = $\theta_5+\theta_7$
$\theta_2 \neq 0, \theta_4 = 0, \theta_6 \neq 0$	Result = $\theta_3+\theta_5$

$$C_1 = -(L_3(pA_yC_4S_2-pA_zS_3S_4+pA_yC_2C_3S_4)+pA_yL_2S_2)/(-L_2^2C_2^2+L_2^2-2L_2L_3C_2^2C_4+2S_2S_4L_2L_3C_2C_3+2L_2L_3C_4-L_3^2C_2^2C_3^2C_4^2+L_3^2C_2^2C_3^2-L_3^2C_2^2C_4^2+2S_2S_4L_3^2C_2C_3C_4+L_3^2C_3^2C_4^2)-L_3^2C_3^2+L_3^2) \quad (8)$$

Because of the absolute value of the cosine is less than or equal to 1 and θ_1, θ_2, θ_3, θ_4 are confirmed, thenθ_4, θ_5, θ_6, can be solved. Therefore, the condition of the solution for θ_2 is that the right side of type (5), (7), (8) is less than or equal to 1. According to the solvable conditions of inequality and the current location, θ_2 can be determined.

Solving θ_3, θ_1: θ_3 Can be solved from equation 5; θ_1 can be solved from Eq. 7 and 8.

Solve θ_5, θ_4, θ_6: Use inverse transform method (Youlun *et al*., 2008) to solve θ_5, θ_4, θ_6.

Singularity-bit processing: From structural characteristics of 7-DOF manipulator, we can discover that when θ_2, θ_4 or θ_6 is zero, manipulator will be singular configuration. Singularities in this study as shown in Table 2:

After solving, we can get the corresponding values of joint. Then according to the current position of arm and certain principle, we can solve the corresponding joint angles. Considering the farther from the terminal movement joints the heavier load is required, the allocation principles of this study is that allocate movement angles to these joints which is near the terminal movement joints.

OPTIMIZATION ALGORITHM

Using classification to solve and inline function are main optimization methods in the algorithm. According to the classification, we can use return and break to end the program when it identified no solution or the completed solution. It can improve the efficiency of the program. Using the small functions frequently, it increases the inverse time greatly. But C^{++} built-in function could embed small functions into the main program compile time directly and decreasing time consume greatly.

After optimization on windows system, the procedure of kinematic even at a bad situation just takes less than 0.02is and meets real-time requirements.

Using Visual C^{++} 6.0 verifies correctness of algorithm (Fig. 7).

Use following code to census time consume of algorithm execution.

```
double cost;
LARGE_INTEGER BeginTime;
LARGE_INTEGER EndTime;
LARGE_INTEGER Frequency;
QueryPerformanceFrequency(&Frequency);
QueryPerformanceCounter(&BeginTime);
// Algorithm Program
QueryPerformanceCounter(&EndTime);
cost = (double)(EndTime.QuadPart-BeginTime.QuadPart)
/Frequency.QuadPart;//is
```

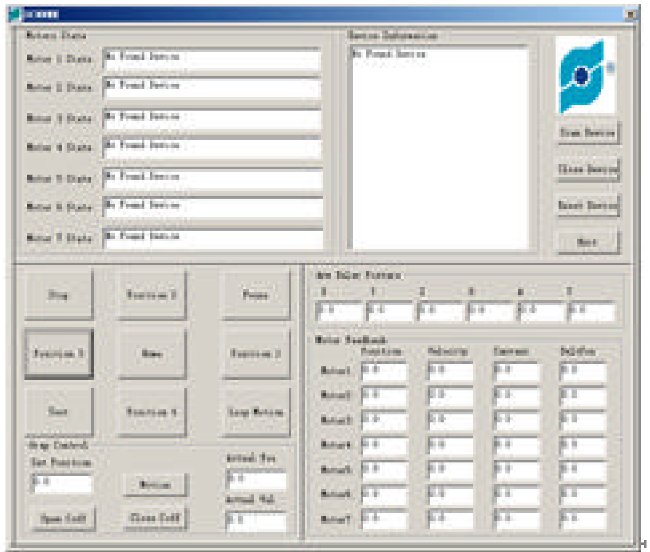

Fig. 8: Motion control

Table 3: Time consume

Class	Consume time (µs)
$\theta_2 = 0, \theta_4 = 0, \theta_6 = 0$	0.000082
$\theta_2 = 0, \theta_4 = 0, \theta_6 \neq 0$	0.000284
$\theta_2 \neq 0, \theta_4 = 0, \theta_6 = 0$	0.000062
$\theta_2 \neq 0, \theta_4 = 0, \theta_6 \neq 0$	0.000259
$\theta_2 = 0, \theta_4 \neq 0, \theta_6 = 0$	0.000286
$\theta_2 = 0, \theta_4 \neq 0, \theta_6 \neq 0$	0.000068
$\theta_2 = 0, \theta_4 \neq 0, \theta_6 = 0$	0.000313
$\theta_2 \neq 0, \theta_4 \neq 0, \theta_6 \neq 0$	0.016969

Statistics result of time shows in Table 3.

Through the test, the algebraic algorithm not only could solve kinematics of 7-DOF redundant manipulator and singularities but also has excellent real-time. Based on the algebraic algorithm, we finish PTP motion control program (Fig. 8) and could make the manipulator to grapan bottle (Fig. 1).

CONCLUSION

This study puts up an algebra algorithm with strong real-time, little computation and mapping all posture for solving kinematics of 7-DoF redundant manipulator and verifies it using experimental platform and makes PTP motion control.

REFERENCES

Alqasemi, R. and R. Dubey, 2007. Redundancy control and optimization of a 9-DoF wheelchair-mounted robotic arm system. Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation, June 20-23, 2007, Jacksonville, FI., pp: 504-509.

Han, Z.X., L. Yan and Y.H. Ren, 2011. 7 DOF humanoid robot inverse kinematics analysis. Sci. Technol. Eng., 11: 1733-1736.

Kalra, P., P.B. Mahapatra and D.K. Aggarwal, 2006. An evolutionary approach for solving the multimodal inverse kinematics problem of industrial robots. Mechanism Mach. Theory, 41: 1213-1229.

Klanke, S., D. Lebedev, R. Haschke, J. Steil and H. Ritter, 2006. Dynamic path planning for a 7-DoF robot arm. Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and System, October 9-15, 2006, Beijing, pp: 3879-3884.

Singh, G.K. and J. Claasens, 2010. An analytical solution for the inverse kinematics of a redundant 7 Dof manipulator with link offsets. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, October 18-22, 2010, Taipei, Taiwan.

Tan, H.Q., 2004. C++ Programming Designer. 4th Edn., Tsinghua University Press, BeiJing.

Youlun, X., L.X. Tang and H. Ding, 2008. The Basis of Robotics. 1st Edn., HuaZhong University of Science and Technology Press, WuHan.