2009 International Conference on Artificial Intelligence and Computational Intelligence

Efficient Algorithms for the Kinematics and Path Planning of Manipulator

Dequan Guo[1,2], Hui Ju[1], Yuqin Yao[1], Feng Ling[1], and Tianxiang Li[2]

[1]Department of Control Engineering
Chengdu University of Information Technology
Chengdu, Sichuan Province, 610225, China
Email: wwwguosky61@yahoo.com.cn

[2]Department of Machinery Engineering
Neijiang Vocational and Technical College
Neijiang, Sichuan Province, 641100, China
Email: gdq2008@gmail.com

***Abstract*—The two basic problems of the automatic control of robotic manipulators are the kinematics and the path planning, which are the fundamental for computer controlled robots. The article presented fast and efficient algorithms for the inverse kinematics and path planning of manipulator consisting of six revolute joints. Through the control, we cause the end-effector of the manipulator to the maximum possible nearby of the expecting position and orientation with some small deviations in the permissible scope. In the solving, the problem of inverse kinematics was reduced to a system of algebraic equations. In the paper, we used a series of algebraic and numeric transformations to reduce the difficult problem to compute the values of the matrix. The resulting algorithm computed all the solutions of manipulators with six revolute joints. In the path planning, we made the end-effector of the manipulator smoothly in the initial and final position and a larger speed in the middle process using the improved method combined of fourth-order cubic and B-spline curve. The above mentioned approaches have been applied in our hand-eye coordinate system.**

Key word: kinematics, solution, interpolation, path planning

I. INTRODUCTION

Kinematics is the science of motion which treats the subject without regarding to the forces that cause it. In the science of kinematics, we often study the position, the velocity, the acceleration, and all higher order derivatives of the position variables. Hence, the study of the kinematics of manipulators refers to all the geometrical and time-based properties of the motion. A manipulator may be thought as a set of bodies connected in a chain by joints. These bodies are called links. Joints form a connection between a neighboring pair of links [1]. In three dimension space, in order to position an end-effector to anywhere, we require six joints at least, because the description of an object in space requires six parameters—three for position and three for orientation.

In general, Kinematics studies the motion without considerations of the forces that produce it, like the flexion of a muscle. To understand different types of Kinematics, it is necessary to describe Forward and Inverse Kinematics. As its name indicates, Forward Kinematics (FK) refers to the direct manipulation of the structure through rotations and translations until it reaches the desired final position. Its hierarchy consists in the assignation of an object as father of another. Say, the structure is basically transformed in a top-down transversal until the wanted effect is reached. FK is conceptually very easy to understand and it is many times very useful [3].Inverse Kinematics (IK) solves the problem of the final position not through direct manipulation of the structure, but through the articulation of the structure necessary to reach the goal. For some time linked to robotics and to off-line animations packages later, the power of today's machines plus the development of different methods allows us to think of IK used in real-time [3].

For instance, automate robots of an assembly-line in a factory, using the angles formed by their joints to reach the desired point [3]. The node that is wanted to reach certain point is called end-effector, a mechanical hand in the example. The first node in a parent-child chain (manipulator) is referred as base. This involves the concept of skeleton. A skeleton can be manipulated in a top-down approach using FK. However, the main advantage of the use of skeletons is the fact that enables to transfer the control of the hierarchy in a bottom-up direction. The use of IK considerably reduces the time consumed to generate, for instance, animations compared to FK. As seen, the advantage of IK is that transformations in the end-effector produce changes in the rest of the nodes without necessity that the user had to manually calculate their transformations. For example, if it is wanted that a character reached an object, instead of calculating each transformation, the user can indicate the goal and the rest of the structure will be re-located in a way that the end-effector reached this point.

After many years of research by many scholars'efforts, we have gotten some progresses. In the general case, the problem reduces to computing all the solutions of a multivariate algebraic system. The main interest has been in establishing tight bounds on the number of solutions and computing them efficiently and robustly. The robustness of the algorithm refers to the correct computation of all the solutions. Among all manipulator configurations, inverse kinematics of 6R manipulators seems to have received the most attention [4].One approach is the algebraic formulations, which can be used along with algorithms for finding roots of a univariate polynomial. Because of numerical problems ,this way is slow in practice, and difficult to generalize to manipulators with special geometries [4];As opposed to

978-0-7695-3816-7/09 $26.00 © 2009 IEEE
DOI 10.1109/AICI.2009.241

eliminating variables from a system of equations, an approach is based upon homology methods is offered. In this way, they compute the solutions of the algebraic system by following paths in the complex space [4].In the literature [7], six degrees of freedom manipulator is decomposed into the structure of the location and gesture, then using the conversion matrix structure of the two inverse kinematics analysis, and established the operator of the inverse kinematics, then obtain the angle values. We can see an improved genetic algorithm that is used to solve the inverse kinematics in literature [8].In the simulation; the way solved the inverse kinematics more precisely, more quickly and more stably than the simple genetic algorithm.

II. ROBOTIC MANIPULATOR KINEMATICS

A. Establish the Link Coordinates

Assuming the link of robotic manipulator, this is consisted of any number of joints and can be constituted in any form. Figure.1 shows the order of three and two-link joints. Although the joint is not necessarily the actual robot with any joint or link is similar to, but they are very common to see and can easily be expressed any joint real robot [1].

a_i: the distance from $\hat{Z}_i$ to $\hat{Z}_{i+1}$ measured along $\hat{X}_i$;

α_i: the angle from $\hat{Z}_i$ to $\hat{Z}_{i+1}$ measured along $\hat{X}_i$;

d_i :the distance from $\hat{X}_{i-1}$ to $\hat{X}_i$ measured along $\hat{Z}_i$;

θ_i :the distance from $\hat{X}_{i-1}$ to $\hat{X}_i$ measured along $\hat{Z}_i$.

We usually choose a_i >0, because it corresponds to a distance; however, α_i, d_i ,and θ_i are signed quantities.

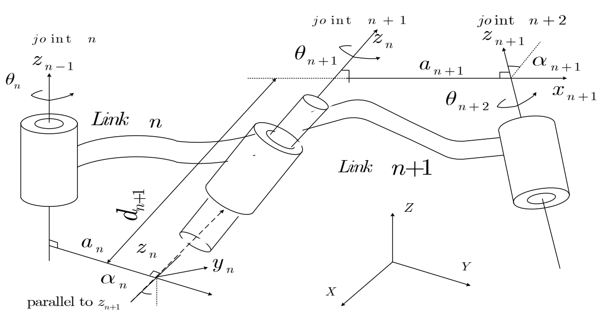

Figure 1. Location of intermediate frames

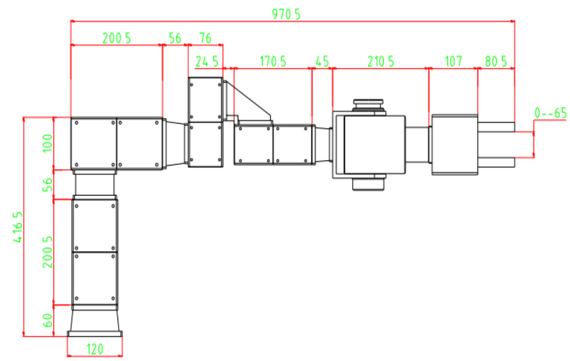

Figure 2. Configure of manipulator

TABLE I. LINK PARAMETERS OF THE MANIPULATOR

link	a_i	α_i	d_i	θ_i	Range of join (degree)
1	0	0	366.5	0	-180～180
2	0	-90	242	0	-120～120
3	0	90	0	0	-120～120
4	0	-90	356	0	-120～120
5	0	90	0	0	-180～180
6	0	-90	280	0	-180～180

B. Forward Kinematics

If we know the each angle of the joint, we can easily calculate position and orientation of the end-effector. From Figure .1,we obtain the general form ri-1,i[1]:

$$ri-1,i=\begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i c\alpha_{i-1} & c\theta_1 c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Using the link parameters shown in Table 1, for the robot of Figure .2, compute the individual transformations for each link.

Substituting the parameters into (1), we obtain

$$r01=\begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 366.5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad r12=\begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 242 \\ -s\theta_2 & -c\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$r23=\begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad r34=\begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 356 \\ -s\theta_4 & -c\theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$r45=\begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_5 & c\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad r56=\begin{bmatrix} c\theta_6 & -s\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 280 \\ -s\theta_6 & -c\theta_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Multiplying the link transformation matrices, we obtain the following expression:

$$r06=r01(\theta_1)r12(\theta_2)r23(\theta_3)r34(\theta_4)r45(\theta_5)r56(\theta_6) \quad (3)$$

C. Inverse Kinematics

The problem of solving the inverse kinematics is fundamental in the design of robot manipulators, and the equations of a manipulator is a nonlinear one[1].Given the numerical value of ${}_{N}^{0}T$, we attempt to find values of $\theta_1, \theta_2, \ldots, \theta_n$. Given the manipulator geometry and the pose of the end-effector, we are interested in computing all the joint positions corresponding to that pose. In this paper, we restrict ourselves to revolute manipulators with up to six joints. For special geometries such as those characterized by intersecting or parallel joint axes, the solution can be expressed as a closed form function. However, no such formulation is known for the general case and no good practical solutions are available to solve those using iterative methods [4]. For the case of an arm with six degrees of freedom, we have 12 equations and six unknowns. However, among the equations arising from the rotation-matrix portion of ${}_{6}^{0}T$, only 3 are independent. These, added to the 3 equations from the position-vector portion of ${}_{6}^{0}T$, given 6 equations with six unknowns. These equations are nonlinear, transcendental equations, which are difficult to solve. As with any nonlinear set of equations, we must concern ourselves with the existence of solutions, with multiple solutions, and with the method of solution [1].

Multiplying solutions are another possible problem encountered in solving kinematical equations. Generally speaking, the more nonzero link parameters there are the more methods there will be to reach a demand goal. We often see some rotary-jointed manipulator with six degrees of freedom (DOF), there will be up to sixteen possible solutions [1].Considering possible solutions, it will be wise to define what constitutes the "solution" of a given manipulator. The solution strategies are split into two broad classes: closed-form solutions and numerical solutions, the latter are much slower. The former are departed into algebraic and geometric by the methods of obtaining the solution. A major recent result in kinematics is that, according to our definition of solvability, all systems with revolute and prismatic joints having a total of six degrees of freedom in a single series chain are solvable [1].In the geometric approach to finding a manipulators' solution, we try to decompose the spatial geometry of the manipulator into several plane-geometry problems. In this way, joint angles can be solved for through using the tools of plane geometry [6].The solution is often used to find a solution requiring the arm in plane.

Supposing the expect equation is TX=r06, and it is known at initial time.

We wish to solve:

$$TX = \begin{bmatrix} nx & ox & ax & px \\ ny & oy & ay & py \\ nz & oz & oz & pz \\ 0 & 0 & 0 & 1 \end{bmatrix} = r06 \tag{4}$$

Where θ_i is the i th joint rotation angle, S_{n+1} is shorthand for $\sin\theta_{n+1}$, C_{n+1} is shorthand for $\cos\theta_{n+1}$, after, we use this expression to express trigonometric functions of angle.

A restatement of (4) that puts the dependence on θ_1 on the left-hand side of the equation is:

$$[r01(\theta_1)]^{-1} r06 = r12(\theta_2) r23(\theta_3) r34(\theta_4) r45(\theta_5) r56(\theta_6) \tag{5}$$

Inverting r01, we write (5) as

$$T16 = \begin{bmatrix} c_1 & s_1 & 0 & 0 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & -366.5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} nx & ox & ax & px \\ ny & oy & ay & py \\ nz & oz & oz & pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6}$$

$$r16 = r12(\theta_2) r23(\theta_3) r34(\theta_4) r45(\theta_5) r56(\theta_6)$$

The simple technique of multiplying each side of a transform equation by an inverse is often employed to advantage in separating out variables in the search for a solvable equation. From the matrices, Equating the (1,3),(1,4),(2,3),(2,4),(3,3),(3,4)elements from both sides of equation T16=r16. We have:

$$c_1 ax + s_1 ay = -c_2 s_3 c_5 - c_2 c_3 c_4 s_5 + s_2 s_4 s_5 \tag{7}$$

$$c_1 px + s_1 py = -280(c_2 c_3 c_4 - s_2 s_4) s_5 - 280 c_2 s_3 c_5 - 356 c_2 s_3 \tag{8}$$

$$-s_1 ax + c_1 ay = c_3 c_5 - s_3 c_4 s_5 \tag{9}$$

$$-s_1 px + c_1 py = 242 - 280 s_3 c_4 s_5 + 280 c_3 c_5 + 356 c_3 \tag{10}$$

$$az = s_2 c_3 c_4 s_5 + c_2 s_4 s_5 + s_2 s_3 c_5 \tag{11}$$

$$-366.5 + pz = 280(s_2 c_3 c_4 + c_2 s_4) s_5 + 280 s_2 s_3 c_5 + 356 s_2 s_3 \tag{12}$$

Multiplying 280 to (9) then subtract to (10) we obtain:

$$280 c_1 ay - 280 s_1 ax - c_1 py + s_1 px = -242 - 356 c_3 \tag{13}$$

In the same way, we obtain

$$280 c_1 ax + 280 s_1 ay - c_1 px - s_1 py = 356 c_2 s_3 \tag{14}$$

$$280 az + 366.5 - pz = -356 s_2 s_3 \tag{15}$$

From (10) and (11), we obtain:

$$(280ax - px)^2 + (280ay - py)^2 = 356^2 c_2^2 s_3^2 + (-242 - 356 c_3)^2 \tag{16}$$

Substituting (15) into (16), we obtain

$$(280ax - px)^2 + (280ay - py)^2 = 356^2 s_3^2 - (280az + 366.5 - pz)^2 + 242^2 + 2*242*356 c_3 \tag{18}$$

So we solve for the possible solution as

$$\theta_3 = \arccos \frac{(280ax - px)^2 + (280ay - py)^2 + (280az + 366.5 - pz)^2 - 242^2 - 356^2}{2*242*356} \tag{19}$$

Then substitute (19) into (15) we obtain

$$\theta_2 = \arcsin \frac{pz - 280az - 366.5}{s_3} \tag{20}$$

Then substitute (19) into (14) we obtain

$$\theta_1 = \arctan\left(\frac{280ay - py}{280ax - px}\right) - \arctan\left(\frac{-242 - 356 c_3}{\pm\sqrt{(280ay - py)^2 + (280ax - px)^2 - (242 + 356 c_3)^2}}\right) \tag{21}$$

Inverting r01, r12 and r23, we write (3) as

$$[r01(\theta_1)]^{-1} [r12(\theta_2)]^{-1} [r23(\theta_3)]^{-1} r06 = r34(\theta_4) r45(\theta_5) r56(\theta_6) \tag{22}$$

$$T36=\begin{bmatrix} c_1c_2c_3-s_1s_2c_3 & c_1s_2+s_1c_2 & c_1c_2s_3-s_1s_2s_3 & 0 \\ -c_1s_2c_3-s_1c_2c_3 & c_1c_2-s_1s_2 & -c_1s_2s_3-s_1c_2s_3 & 0 \\ -s_3 & 0 & c_3 & -608.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} nx & ox & ax & px \\ ny & oy & ay & py \\ nz & oz & oz & pz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$r36 = r34(\theta_4)r45(\theta_5)r56(\theta_6) \qquad (23)$$

Because T36=r36, both sides (2,4) elements are equate.

$$356+280c_5 = -c_1s_2 - s_1c_2c_3px + c_1c_2 - s_1s_2py + (-c_1s_2 - s_1c_2)s_3pz \qquad (24)$$

$$\theta_5 = \arccos\frac{-c_1s_2 - s_1c_2c_3px + c_1c_2 - s_1s_2py + (-c_1s_2 - s_1c_2)s_3pz - 356}{280} \qquad (25)$$

or (2,3) we obtain:

$$c_5 = (-c_1s_2 - s_1c_2)c_3ax + c_1c_2 - s_1s_2ay + (-c_1s_2 - s_1c_2)s_3az \qquad (26)$$

Using (26) we can check (25)

From (1, 3) and (3, 3) elements we obtain:

$$-c_4s_5 = (c_1c_2 - s_1s_2)c_3ax + (c_1s_2 + s_1c_2)ay + (c_1c_2 - s_1s_2)s_3az \qquad (27)$$

$$s_4s_5 = -s_3ax + c_3az$$

$$\theta_4 = \arctan\frac{s_3ax - c_3az}{(c_1c_2 - s_1s_2)c_3ax + (c_1s_2 + s_1c_2)ay + (c_1c_2 - s_1s_2)s_3az} \qquad (28)$$

At the same time, we can check θ_4 by (1,4) and (3,4) elements; From (2,1)and (2,2) elements, we obtain:

$$s_5c_6 = (c_1c_2 - s_1s_2)c_3ax + (c_1s_2 + s_1c_2)ay + (c_1c_2 - s_1s_2)s_3az$$

$$-s_5s_6 = (-c_1s_2 - s_1c_2)c_3nx + (c_1c_2 - s_1s_2)ny + (-c_1s_2 - s_1c_2)s_3nz \qquad (29)$$

$$\theta_6 = -\arctan\frac{(-c_1s_2 - s_1c_2)c_3nx + (c_1c_2 - s_1s_2)ny + (-c_1s_2 - s_1c_2)s_3nz}{(c_1c_2 - s_1s_2)c_3ax + (c_1s_2 + s_1c_2)ay + (c_1c_2 - s_1s_2)s_3nz} \qquad (30)$$

Until now, we obtain all the solutions; the next step is how to select the available or optimistic solutions, according to the ranges of joint.

D. Optimize Solution

For the robot of six degrees of freedom, there is available closed solution to the inverse kinematics in only two types of structure [9].that are:

- The last three joint axes intersect at one point;
- There are three joint axes which are parallel to each other;

Other type's structure of robots only can be obtained closed inverse solutions through the numerical method. This paper studied the modular robot to meet the first condition. We can draw the following methods of the joints of the robot and the optimal solution feasible solution:

a) According to the value of the angle of the trigonometric functions, we derive the value of all possible angles, and then in accordance with the scope of the joint movement restrictions are not removed within the scope of the movement angle.

b) As a result of the continuity of movement, can not be a major point of the beating. Therefore, the actual calculation can be used in the previous sample value and the angle of movement at all times compared to the value, the value of similar value is obtained by calculating the reasonable value.

c) In accordance with the principle of the optimal time, check to make changes in the joints of the volume and the smallest absolute values of the expectations of the joint movement values.

E. Experiments

For checking the effect, we choose a group data.

At first, supposing: Angle1=0; Angle2=0; Angle3=30; Angle4=30; Angle5=30; Angle6=0;

We obtain:

$$\begin{bmatrix} 0.3995 & -0.4330 & -0.8080 & -404.2436 \\ 0.8080 & -0.2500 & 0.5335 & 699.6833 \\ -0.4330 & -0.8660 & 0.2500 & 436.500 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Then, we substitute the matrices into TX, and using the angle obtained by our algorithm, we can get the angles, as follows: Angle1=1.000; Angle2=0; Angle3=30.004; Angle4=16.3524; Angle5=35.9501; Angle6=1.1968; in the experiments, the angle4 derivates more to the correct value. The reason might be the character of mechanic. The issues will to be further enhanced in the next work.

III. PATH PLANNING

During moving the manipulator from an initial position to a desired final position, the path planning is very important in the process, sometimes, the manipulator may reach the undesired position, or don't move to the available position non-smoothly. So it is necessary to specify the motion in much more detail than by simply stating the desired final configuration. One way to include more detail in a path description is to give a sequence of desired via points(intermediate points between the initial and final positions).Thus, in completing the motion ,the tool frame must pass through a set of intermediate positions and orientations as described by the via points. Each of these via points is actually a frame that specifies both the position and orientation of the tool relative to the station. The name path points includes all the via points plus the initial and final points [1].points are actually frames, which give both position and orientation. We need the motion of the manipulator to be smooth. For this purpose, the function of motion is continuous and has a continuous first derivative, and a continuous second derivative is also desirable in particular space.

Robot trajectory planning mainly involves the following three questions [10]:

- To describe the task of robot and manipulator path and trajectory of the movement;
- To compute trajectory required internal computer, according to the trajectory parameters;
- To calculate the practical trajectory, that is, generating the trajectory of motion including position, velocity and acceleration.

A. Cubic Polynomials Interpolation

If there are strict requirements of the operation, path for the robot movement and gesture is necessary to path planning in space of rectangular coordinates, that is, Cartesian space. Sometimes, there is no rigid request and the shape of the path is complex if described in Cartesian space. We emphasize importance of the position accurately and quickly and with enough space for the manipulator. Therefore, we only use variable joint space trajectory planning.

In the joint-space schemes, we usually employ cubic polynomials, or higher-order polynomials. For obtaining a single smooth motion, at least four constraints on $\theta(t)$ are evident. Two constraints on the function's value come from the selection of initial and final values:

$$\theta(0)=\theta_0 \quad ,\theta(t_f)=\theta_f \quad ,\dot{\theta}(0)=0 \ ,\dot{\theta}(t_f)=0 \ . \quad (31)$$

In the practice, requiring the initial and final velocity is zero. Only the four constraints are satisfied by a polynomial of at least third degree [1].These constraints uniquely specify a particular cubic. This is:

$$\theta(t)=a_0+a_1t+a_2t^2+a_3t^3 \quad (32)$$

$$a_0=\theta_0 \ , \ a_1=0 \ , \ a_2=\frac{3}{t_f^2}(\theta_f-\theta_0) \ , \ a_3=\frac{2}{t_f^3}(\theta_f-\theta_0) \quad (33)$$

Using（33）, then calculating the cubic polynomial that connects any initial joint-angle position with any desired final position is easy and making the joint starts and finishes at zero velocity. For example the initial rotary joint is motionless at θ=10 degrees, the final joint is θ=70, in 3 seconds.

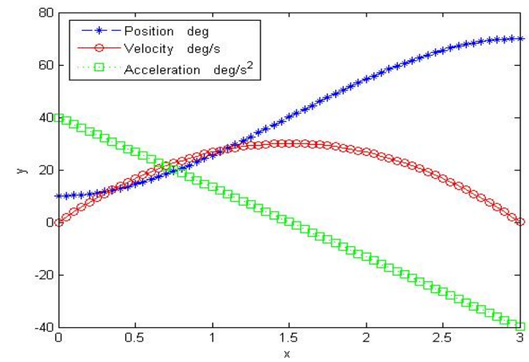

Figure.3 Position, velocity and acceleration profiles

B. Cubic Polynomials for a Path with Via Points

There are several ways in which the desired velocity at the via points might be specified [1]:

1) The user specifies the desired velocity at each via point in terms of a Cartesian linear and angular velocity of the tool frame at that instant.

2) The system automatically chooses the velocities at the via points by applying a suitable heuristic in either Cartesian space or joint space.

3) The system automatically chooses the velocities at the via points in such a way as to cause the acceleration at the via points to be continuous.

Sometimes, we use linear function with parabolic blends. There might be some questions in the Cartesian paths: intermediate points unreachable; high joint rates near singularity; start and goal reachable in different solutions.

C. B-Spline Method

Because the B-spline is based on the expression vector, which has a unified expression, and all the coefficients are constant with a small amount of calculation [11]; At the same time, all the control points of B-spline is calculated in one-time, and the path is formed in depart of the sub-structure, Therefore, control points can be obtained through off-line, the track could be online with real-time characters; Moreover, B-spline has convexity and local support, which are important to employ to design the B-spline curves of the path and having great significance. Based on the B-spline curve interpolation algorithm controls the robot's joints' movement to rotate in the way of speeding up, average speed, deceleration; While based on cubic polynomial interpolation algorithm controls robot joints' movement in the way of speed up, slow down ,which is not conducive to the robot movement.

In reference [12], the paper exploited the similarity between the movements of a humanoid robot and human movement to generate joint trajectories for such robots. In particular, they show how to transform human motion information captured by an optical tracking device into a high dimensional trajectory of a humanoid robot. They utilized B-spline wavelets to efficiently represent the joint trajectories and to automatically select the density of the basis functions on the time axis. They applied their method to the task of teaching a humanoid robot how to make a dance movement.

The literature [13] presented a general solution to this problem that involves joint-space tessellation, a dynamic time-scaling algorithm, and graph search. The solution incorporated full dynamics of movement and actuator constraints, and could be easily extended for joint limits and workspace obstacles, but was subject to the particular tessellation scheme used. In their results showed that, in general, the optimal paths were not straight lines, but rather curves in joint-space that utilized the dynamics of the arm and gravity to help in moving the arm faster to its destination.

D. The Improved Method

Through the above analysis, in view of B-spline with a fast start and end but smooth in the middle of the process; while fourth-order cubic with a slower start and steep middle process. We use fourth-order cubic (or multitude polynomials interpolation) at the initial process when the value up to a setting standard, changing it to B-spline method, then, transferring to multitude polynomials interpolation at the nearby final position. At the same time, the method has the merits of both above mentioned approaches.

IV. CONCLUSION

In this paper, we presented an efficient and quick algorithm for inverse kinematics of a 6R manipulator. Starting from the robot kinematics, using D-H expression to establish modular robot kinematics equations and obtain the solutions of robot inverse kinematics equation based on the algebraic method then analysis possible solutions or no solution of the situation in algebraic method. The algorithm only solving the numerical solutions, we can obtain the closed-form solutions. This avoids the complexity of solving geometry. However, there is still some improvement to select the solutions in the real-time system. For the purpose of the solution are more practical in the robot path planning, furthermore, introduce the trajectory planning methods based on joint space and Cartesian space, and the B-spline curve, then carry out an analysis of the advantages and disadvantages. At last, combining the both merits, we obtain an improved method and have better effect.

ACKNOWLEDGEMENT

This work has been supported by the Shanghai Ingenious Automation Co.Ltd. Without their help, we can’t do the above work.

REFERENCES

[1] John J,Craig.“Introduction to Robotics Mechanisc and Contronl”.BeiJing:China Machine Press.2005 5.

[2] Saeed B.Niku.”Introduction to Robotics Analysis,Systems,Applications”.BeiJing:China Pulishing House of Electronics Industry.2004.1.

[3] Diego Park.” Inverse Kinematics”. Computer Graphics, Department of Computer Science, University of Buenos Aires, Argentina.

[4] Dinesh Manocha, “Yunshan Zhu. A fast algorithm and system for inverse kinematics of general serial manipulators”. In Proceedings of the IEEE Conference on Robotics and Automation,pp.1-6,1994.

[5] WATTA., POLICARPO F.: 3D Games: Animation and Advanced Real-time Rendering, vol. 2. Harlow: Addison-Wesley,2002.

[6] C.S.G.Lee and M.Xiegler,”Geometric Approach in Solving Inverse kinematics of PUMA Robots,”IEEE Transactions on Aerospace and electronic Systems ,Vol.AES-20,No.6,November 1984.

[7] Bi Z M, Cai H G. Inverse Kinematics Problem of six Degrees of Freedom Manipulator. Robot, 1994, 16 (2): 92-97.

[8] LI M L , CHEN L S.” A genetic algorithm applying to robot manipulator control” Journal of Fuzhou University(Natural Science) Vol.31(3).2003,6.

[9] Zhao W. “Research and Design of Modular Robot Motion Control System”.College of Information Science and Engineering,Wuhan University of Science and Technology,Wuhan,August,7,2007.

[10] Xiong Y L. Robot Technology Infrastructure. Wuhan: Huazhong University of Science and Technology Press, 1996:46-48

[11] WANG Y M,XU W H. Robot joint space trajectory optimization of B-spline design. Electrical and Mechanical Engineering, Vol. 17, No. 4, 57-60, 2000.

[12] Ales Ude, Christopher G. Atkeson, Marcia Riley. Planning of Joint Trajectories for Humanoid Robots Using B-Spline Wavelets. In Proc. of the IEEE Int. Conf. on Robotics and Automation.2000.

[13] Gideon Sahar, John M., John M. Holierbach. Planning of Minimum-Time Trajectories for Robot Arms. In Proc. of the IEEE Int. Conf. on Robotics and Automation.November,1984