因此在驱动伺服电机时，在向D/A送数的同时必须选择保持电路通道号（确定哪一路S/H)。对于软件来说，驱动伺服电机的过程也就是向D/A送数及选择S/H通道的过程。向D/A送数的流程示于图10。

8. 故障处理

焊接机器人在工作过程中有可能出现各种不正常情况，如机器人本体可能会出现超速、越限、控制系统出错等，焊机系统可能会出现沾丝、断弧、气压不足等问题。无论是何种异常现象一旦出现，控制软件就立即清除D/A输出数据缓冲区并且制动电机运转，如果正在焊接，则关闭与焊机有关的所有设备。待查明异常原因且处理完毕，方可恢复工作。

四、结　束　语

经实践证明，该控制软件运行安全可靠，定位精度高，机器人运动平稳，达到了满意的效果。

由于本系统没有采用容错冗余技术，一旦CPU出现故障则机器人就无法正常工作，唯一的办法是人工紧急停机。

最后说明一点，HRSJ-1机器人专用计算机系统不仅可以用于点、弧焊机器人系统，而且也可以用于其它用途的机器人系统，如搬运、装配等机器人。

参　考　文　献

[1] 《中日机器人技术交流会会议资料》，1986年11月。
[2] "Robotics Today", April 1986, Vol. 8, No. 2.
[3] "The International Journal of Robotics Research", Vol. 5, No. 2, Summer 1986.
[4] "The International Journal of Robotics Research", Vol. 5, No. 4, Winter 1987.

机器人的运动控制方式及插补算法

航天工业部二院十七所　姜方荣

摘要　本文对机器人的两种控制方式，即点位(PTP)控制和连续轨迹(CP)控制进行了叙述，并以一种实用机器人系统为例介绍了数值插补算法在这两种控制中的应用。

主题词　机器人，控制，伺服系统，算法。

一、引　言

多关节工业机器人的运动控制一般是通过伺服系统和计算机来实现的。计算机根据机器人几何尺寸和关节角在基本坐标系中建立起运动方程，经过变换产生机器人的各关节角坐标，从而确定机器人的空间位置和姿态。伺服系统则完成机器人手的空间运动，使其精确地到达期望位置。机器人的运动控

收稿日期：1987年6月27日

制可分为点位控制和连续轨迹控制两类。数值采样插补算法是实现这两种控制的有效而简单的算法之一。

二、控制方式描述

1. 点位控制

点位控制能够确保机器人手的定位精度，但对相邻两个点的运动轨迹则不加控制。点位控制按其复杂程度可分为三种：（1）固定位置的点位控制。比如在两点之间，搬运工件的机器人就可采用此法。它无需伺服回路系统，只需确定各关节在运动空间两个固定点处的关节坐标，通过驱动源实现往复运动即可。这种系统不能实现人工示教，是一种最原始的点位控制，即所谓“固定位置的取放系统”。（2）可编程点位控制。在这种系统中每个关节均采用电动或液压伺服系统来实现机器人的运动，因而可以在运动空间随意设定许多个停留定位点，但它并不进行轨迹控制。它可以通过示教盒或计算机终端显示器进行示教。显然它比上法灵活，常用在装配、点焊等方面。（3）对轨迹进行粗略控制的点位控制，这是对第二种控制的改进。它除了能进行点的精确定位以外，还能使机器人手在相邻两定位点之间按逼近直线的轨迹运动。它可以通过示教设定空间运动的若干定位点，同时还可设定各关节的运动速度及其它辅助参量。计算机对这些参数进行大量的在线或离线运算，即可保证机器人手既能实现精确的机械定位又可获得良好的接近直线的运动轨迹。所有这些计算一般是在机器人关节坐标系下完成。由于它比连续轨迹控制相对简单一些，因而在轨迹精度要求不高的场合最为适用，如打毛刺、装配、点焊等。

2. 连续轨迹控制

连续轨迹控制不但要保证定位点的位置精度，而且要按相邻两定位点间规定的轨迹运动。连续轨迹运动控制可分为实时控制和离线控制。（1）实时连续轨迹控制是计算机在机器人运动过程中每隔一段时间（5～100ms）采集一次机器人手当前的实际位置，根据规定的定位点位置和设定速度，按特定算法实时地计算出机器人手单步长后的前向位置，使机器人按规定轨迹运动。事实上连续轨迹控制是通过若干插值点来实现的。这种控制方法由于需要大量实时计算，因而计算机必须有较强的运算能力和较快的运算速度。（2）离线连续轨迹控制与上法不同，它是将机器人若干定位点和轨迹事先精确计算好（比如在示教中计算好）并贮存于存贮器中。当机器人实际运动时将这些事先计算好的轨迹调出来执行即可。显然离线计算比实时计算简单一些，但是它内存开销很大并且要求机器人有很高的重复精度，否则再现时会影响轨迹精度。

三、数值采样插补在运动控制中的应用

实现关节机器人在两定位点间的过渡有多种插值逼近方法。数值采样插补算法是简单而有效的方法之一。下面介绍一种实用焊接机器人所采用的数值插补算法。这种机器人同时兼有点位控制和连续轨迹控制，采用示教再现方法工作。在该系统中，微机每隔5ms采集一次焊枪的空间位置以形成位置反馈控制回路。每采样一次，微机便根据设定速度计算出机器人下一步长各关节角的位置，供伺服系统使用。

1. 点位控制关节角插补

示教时取得定位点的关节角坐标位置，再现时微机实时计算相邻两定位点之间的若干关节角插值位置，经变换后由伺服系统使机器人运动，以此实现点的定位和近似直线的弧形轨迹。设第i个关节在关节坐标系中的当前位置是Q_{ci}，单步长后关节角位移增

量是 ΔQ_i，则单步长后焊枪末端的位置为

$$Q_{ni}=\Delta Q_i+Q_{ci} \quad (1)$$

设该关节运动速度为ω_i，则（1）式变为

$$Q_{ni}=\omega_i\Delta t+Q_{ci} \quad (2)$$

Δt为步长时间，恒取100ms。值得注意的是式（2）中角速度ω_i并不是示教时设定的值ω_{ti}。设 Q_{di} 为第 i 个关节目标定位点的位置，则各关节从当前位置运动到目标定位点所需时间为

$$t_i=\frac{Q_{di}-Q_{ci}}{\omega_{ti}} \quad (3)$$

由于各关节设定的速度 ω_{ti} 可能不同，位移也不一样，因而各关节运动时间 t_i 也就不同，从而造成各关节不能同时到位。为了保证各关节同时到达定位点，必须对各关节设定速度进行修正。现取 t_i 中最大值为各关节共同的运动时间，设为 T，则各关节的实际运动速度即为

$$\omega_i=\frac{Q_{di}-Q_{ci}}{T} \quad (4)$$

代入式（2）得

$$Q_{ni}=\frac{Q_{di}-Q_{ci}}{T}\cdot\Delta t+Q_{ci} \quad (5)$$

其运动轨迹见图1。

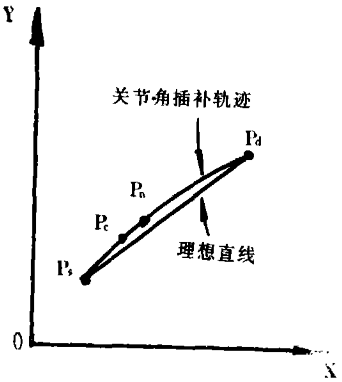

图1

2. 连续轨迹插补

因为任何空间曲线均可由空间直线和圆弧来拟合，故可以采用直线插补和圆弧插补相结合的方法来实现空间曲线。

（1）直线插补　此算法适用于直线轨迹。在基本坐标系中已知当前点和定位点的直角坐标（见图2）可求得这两点间的直线距离L为

$$L=\sqrt{(x_d-x_c)^2+(y_d-y_c)^2+(z_d-z_c)^2} \quad (6)$$

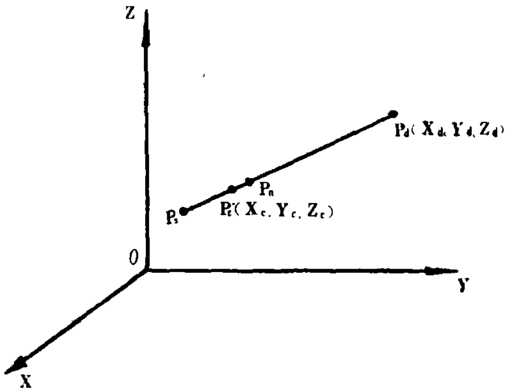

图2

设运动速度为V(恒速)，则运动时间为

$$T=\frac{L}{V} \quad (7)$$

故坐标系中速度分量为

$$V_X=\frac{X_d-X_c}{T}=\frac{(X_d-X_c)V}{L}$$

$$V_Y=\frac{Y_d-Y_c}{T}=\frac{(Y_d-Y_c)V}{L}$$

$$V_Z=\frac{Z_d-Z_c}{T}=\frac{(Z_d-Z_c)V}{L} \quad (8)$$

因此单步长后的直角坐标位置即为

$$X_n=V_X\Delta t+X_c$$
$$Y_n=V_Y\Delta t+Y_c$$
$$Z_n=V_Z\Delta t+Z_c \quad (9)$$

式中Δt 恒为100ms。根据式（9）及焊枪端点当前位置的各关节角位置由机器人逆几何变换方程即可求得插值点 P_n 所对应的关节角。经过若干步插值，焊枪端点即可按直线轨迹到达目标定位点 P_d。按此算法焊枪运

动轨迹同理想直线几乎一致，精度极高。

（2）空间圆弧插补　任何空间圆弧均可通过一系列坐标变换归结为新坐标系下的平面圆弧问题，因而简单和易于处理。由于空间不共线的三个点即可构成一个圆周，因此示教时只要设定期望圆弧上三个点即可通过本算法实现圆弧轨迹。

圆弧插补是通过对圆心角进行数值插补来实现的。

设机器人基本坐标系为 $OXYZ$，圆弧插补坐标系为 $OX'Y'Z'$（见图3）。其中 OZ' 为平面 M 的法线，P 为交点。OP 在 OXY 平面上的投影 OA 与 X 轴夹角为 θ_1，Z 与 Z' 夹角为 θ_2。OA 在 $OY'Z'$ 平面上。设圆弧上三点坐标为 $P_1(X_1,Y_1,Z_1)$、$P_2(X_2,Y_2,Z_2)$ 及 $P_3(X_3,Y_3,Z_3)$，该圆弧平面方程即为

$$AX+BY+CZ+D=0 \tag{10}$$

其中

$$A=\begin{vmatrix} Y_1 & Z_1 & 1\\ Y_2 & Z_2 & 1\\ Y_3 & Z_3 & 1\end{vmatrix}\quad B=-\begin{vmatrix} X_1 & Z_1 & 1\\ X_2 & Z_2 & 1\\ X_3 & Z_3 & 1\end{vmatrix}$$

$$C=\begin{vmatrix} X_1 & Y_1 & 1\\ X_2 & Y_2 & 1\\ X_3 & Y_3 & 1\end{vmatrix}\quad D=-\begin{vmatrix} X_1 & Y_1 & Z_1\\ X_2 & Y_2 & Z_2\\ X_3 & Y_3 & Z_3\end{vmatrix}$$

易知平面M法线之方向余弦为

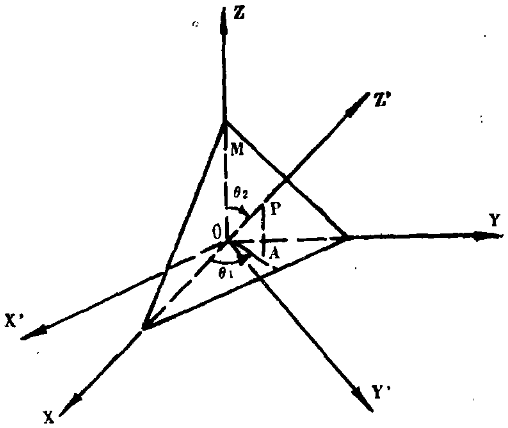

图 3

$$\cos\alpha=\frac{A}{\pm\sqrt{A^2+B^2+C^2}}$$

$$\cos\beta=\frac{B}{\pm\sqrt{A^2+B^2+C^2}}$$

$$\cos\gamma=\frac{C}{\pm\sqrt{A^2+B^2+C^2}} \tag{11}$$

其中 α,β,γ 分别为法线方向角。平面 M 到坐标原点 O 之距离 d 为

$$d=\frac{|D|}{\sqrt{A^2+B^2+C^2}} \tag{12}$$

现将坐标系 $OXYZ$ 进行欧拉变换得插值坐标系并使平面 M 平行于 $OX'Y'$ 平面。令欧拉角为 θ_1、θ_2 和 $\theta_3(=-\theta_1)$，则坐标变换方程为

$$\begin{pmatrix} X\\ Y\\ Z\\ 1\end{pmatrix}=\begin{pmatrix} c^2\theta_1c\theta_2+s^2\theta_1 & c\theta_1c\theta_2s\theta_1-s\theta_1c\theta_1 & c\theta_1s\theta_2 & 0\\ s\theta_1c\theta_2c\theta_1-c\theta_1s\theta_1 & s^2\theta_1c\theta_2+c^2\theta_1 & s\theta_1s\theta_2 & 0\\ -s\theta_2c\theta_1 & -s\theta_2s\theta_1 & c\theta_2 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}\begin{pmatrix} X'\\ Y'\\ Z'\\ 1\end{pmatrix} \tag{13}$$

其逆变换方程为

$$\begin{pmatrix} X'\\ Y'\\ Z'\\ 1\end{pmatrix}=\begin{pmatrix} c^2\theta_1c\theta_2+s^2\theta_1 & s\theta_1c\theta_2c\theta_1-c\theta_1s\theta_1 & -s\theta_2c\theta_1 & 0\\ c\theta_1c\theta_2s\theta_1-s\theta_1c\theta_1 & s^2\theta_1c\theta_2+c^2\theta_1 & -s\theta_2s\theta_1 & 0\\ c\theta_1s\theta_2 & s\theta_1s\theta_2 & c\theta_2 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}\begin{pmatrix} X\\ Y\\ Z\\ 1\end{pmatrix} \tag{14}$$

式中 $c=\cos$，$s=\sin$。

将方程（14）展开即得

$$X'=X(c^2\theta_1c\theta_2+s^2\theta_1)+Y(s\theta_1c\theta_2c\theta_1-c\theta_1s\theta_1)-Zs\theta_2c\theta_1$$

$$Y'=X(c\theta_1c\theta_2s\theta_1-s\theta_1c\theta_1)+Y(s^2\theta_1c\theta_2+c^2\theta_1)-Zs\theta_2s\theta_1$$

$$Z'=Xc\theta_1 s\theta_2+Ys\theta_1 s\theta_2+Zc\theta_2 \tag{15}$$

考虑到平面M平行于$OX'Y'$平面，OZ'为平面法线，故M上所有点之Z'坐标均为常数d，故有

$$Z'=\frac{|D|}{\sqrt{A^2+B^2+C^2}} \tag{16}$$

显然由方程(15)、(16)可得圆弧平面M上三点在$OX'Y'Z'$坐标系下的位置坐标，设为$P_1(X_1',Y_1',Z_1')$，$P_2(X_2',Y_2',Z_2')$和$P_3(X_3',Y_3',Z_3')$（见图4）。由此三点所决定的圆的方程为

$$X'^2+Y'^2-2mX'-2nY'+q=0 \tag{17}$$

$$(m^2+n^2>q)$$

将三点坐标代入得方程组

$$\begin{bmatrix}2X_1' & 2Y_1' & 1\\ 2X_2' & 2Y_2' & 1\\ 2X_3' & 2Y_3' & 1\end{bmatrix}\begin{bmatrix}-m\\ -n\\ q\end{bmatrix}=\begin{bmatrix}-(X_1'^2+Y_1'^2)\\ -(X_2'^2+Y_2'^2)\\ -(X_3'^2+Y_3'^2)\end{bmatrix} \tag{18}$$

记

$$\Delta=\begin{vmatrix}2X_1' & 2Y_1' & 1\\ 2X_2' & 2Y_2' & 1\\ 2X_3' & 2Y_3' & 1\end{vmatrix},$$

$$\Delta_m=\begin{vmatrix}-(X_1'^2+Y_1'^2) & 2Y_1' & 1\\ -(X_2'^2+Y_2'^2) & 2Y_2' & 1\\ -(X_3'^2+Y_3'^2) & 2Y_3' & 1\end{vmatrix},$$

$$\Delta_n=\begin{vmatrix}2X_1' & -(X_1'^2+Y_1'^2) & 1\\ 2X_2' & -(X_2'^2+Y_2'^2) & 1\\ 2X_3' & -(X_3'^2+Y_3'^2) & 1\end{vmatrix}$$

并解方程组（18）即得圆心坐标

$$m=\frac{-\Delta_m}{\Delta},\quad n=\frac{-\Delta_n}{\Delta} \tag{19}$$

由图4易知半径

$$R=\sqrt{(X_1'-m)^2+(Y_1'-n)^2} \tag{20}$$

以圆心G为极点建立极坐标系。令极轴GX''平行于OX'轴(见图5)，设焊枪端点当前位置所对应的极角为Q_c，定位点所对应的极角为Q_d，焊枪运动速度为V，则插点P_n所对应的极角为

$$\begin{aligned}Q_n&=\Delta Q+Q_c\\ &=\frac{V\Delta t}{2\pi R}\cdot 360+Q_c\end{aligned} \tag{21}$$

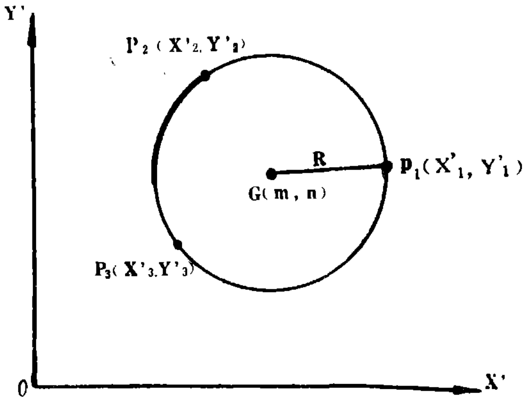

图4

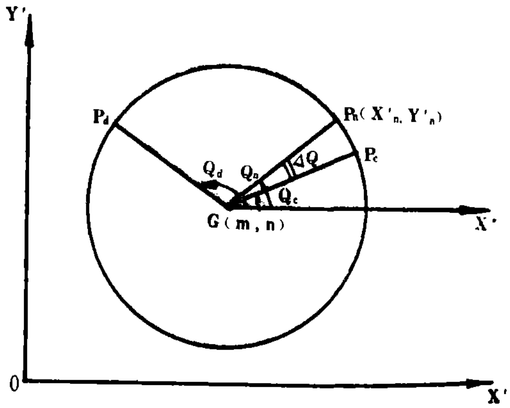

图5

此处Δt恒取100ms。易知P_n在$OX'Y'Z'$坐标系中的坐标为

$$X_n'=R\cos(Q_c+\Delta Q)+m$$

$$Y_n'=R\sin(Q_c+\Delta Q)+n$$

$$Z_n'=\frac{|D|}{\sqrt{A^2+B^2+C^2}} \tag{22}$$

由方程（13）可得$P_n(X,Y,Z)$。与此同时根据速度V还可算得焊枪姿态角的增量。最后经逆几何变换即可求得各关节角所对应的转角增量，从而驱动机器人运动到插补点P_n。

从当前点P_c到定位点P_d插补步长数为

$$N=INT\left(\frac{T}{\Delta t}\right)+1 \tag{23}$$

其中 Δt 为步长时间，T 为总运动时间

$$T=\frac{L_{dc}}{V}=\frac{2\pi R(Q_d-Q_c)}{360V} \qquad (24)$$

这里 L_{dc} 为总弧长。

实用表明采用此法进行圆弧插补轨迹精度很高，能令人满意地完成生产中的焊接作业。

四、讨 论

数值采样插补算法在一般情况下均能保证良好的轨迹精度和定位精度。精度的保证与许多因素有关，采样时间的长短便是其中之一。如果采样时间间隔较短，则意味着轨迹上插补点较多(设速度一定)，因而在位置闭环控制回路中，机器人的空间位置就能不断得到及时校正。反之则插补点减少，位置精度就会受到影响。如何确定采样时间，需要仔细的计算并经实验确定，以便使系统取得最佳控制效果而又能可靠地工作。另外运动速度的大小也是一个重要因素。一般机器人系统采样时间和插补计算步长是固定的。如果速度设定偏大，则会引起插补点减少，从而影响精度，甚至会严重偏离轨迹。故一般机器人系统均有速度限制，以保证良好的点位和轨迹控制精度。

参 考 文 献

[1] Donald H. Jones, "An Insight to Robotic Controllers and Servo System", Proceeding of the First Annual International Robot Conference, 1983.
[2] C. Bergren, Potter Instrument Co., "A Simple Algorithm for Circular Interpolation", Control Engineering, Vol. 18, No. 7, July 1971.
[3] C.Y. Ho and K.W. Copeland, "Solution of Kinematic Equations for Robot Manipulators", Computing Techniques for Robots.
[4] 姜方荣，《机器人3#板软件分析报告》，十七所资料，1986年6月。
[5] 李彦等，《工业机器人连续轨迹控制的插补算法》，机器人技术，1986年第4期。

扩大Fortran数据区的方法

航天工业部二院二部 成 康

摘要 本文介绍了在IBM-PC微机上如何扩大Fortran数据区的方法。使Fortran程序员能够处理大于64k字节的数据量。

目前，IBM-PC机在我国广泛使用。Fortran语言是该机上使用的主力语言之一。但是使用Fortran语言往往受到数据区大小的限制，即数据量不能超过64k字节。这种限制对许多Fortran程序员是不可忍受的。例如，科学计算往往涉及大的数组，矩阵的运算，数据量经常超过64k字节。下面将介绍扩大数据区的技术方法，Fortran程序员用此方法能处理64k字节以上的数据。

一、静态数据的限制

静态数据一般分配在8086的缺省数据段