

捷联惯性导航、制导系统中方向余弦矩阵的递推算法

李连仲, 王小虎, 蔡述江

(北京控制与电子技术研究所, 北京 100038)

摘 要: 数十年来, 四元数及其解法成功地应用于捷联惯性导航和制导系统中, 成为经典的算法。它定义了从导航坐标系到飞行器体坐标系的四元数, 然后给出四元数更新方程, 再根据实时确定的四元数求出体系到导航坐标系的方向余弦矩阵, 以便将测得的体系的视速度增量转换到导航系。从制导和导航角度看, 上述方向余弦矩阵是必不可少的, 而四元数却是中间变量, 因此, 本文跨越了四元数及其算法, 根据方向余弦矩阵微分方程直接导出方向余弦矩阵的更新递推公式。数学仿真表明该算法的精度与四元数算法接近, 但它具有更容易理解、计算量小、编程简单等优点, 可以代替四元数方法。

关键词: 捷联惯性导航; 方向余弦矩阵; 四元数

中图分类号: V249.31 文献标识码: A 文章编号: 1000-1328(2006)03-0349-05

0 引言

数十年来, 四元数及其解法成功应用于捷联惯性导航。具体应用扼要概述如下, 首先定义从导航

坐标系到飞行器体坐标系的四元数为 (q_0, q_1, q_2, q_3) 。则由飞行器体坐标系到导航坐标系的方向余弦矩阵 C_b^c 可写成

$$C_b^c = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (1)$$

当飞行器体坐标系以自转角速度 ω 旋转时, 四元素 $q = (q_0, q_1, q_2, q_3)^T$ 满足微分方程

$$\frac{dq}{dt} = q \circ \omega / 2 \quad (2)$$

式中符号“ \circ ”表示四元数乘法。其递推解为:

$$\begin{bmatrix} q_0(t_{i+1}) \\ q_1(t_{i+1}) \\ q_2(t_{i+1}) \\ q_3(t_{i+1}) \end{bmatrix} = \begin{bmatrix} q_0(t_i) & -q_1(t_i) & -q_2(t_i) & -q_3(t_i) \\ q_1(t_i) & q_0(t_i) & -q_3(t_i) & q_2(t_i) \\ q_2(t_i) & q_3(t_i) & q_0(t_i) & -q_1(t_i) \\ q_3(t_i) & -q_2(t_i) & q_1(t_i) & q_0(t_i) \end{bmatrix} \begin{bmatrix} \cos \frac{\omega \Delta t}{2} \\ -\frac{x_1}{\omega} \sin \frac{\omega \Delta t}{2} \\ -\frac{y_1}{\omega} \sin \frac{\omega \Delta t}{2} \\ -\frac{z_1}{\omega} \sin \frac{\omega \Delta t}{2} \end{bmatrix} \quad (3)$$

式中 $x_1 = \int_{t_i}^{t_{i+1}} \omega_x dt$, $y_1 = \int_{t_i}^{t_{i+1}} \omega_y dt$, $z_1 = \int_{t_i}^{t_{i+1}} \omega_z dt$, 分别为 x_1 、 y_1 和 z_1 速率陀螺的输出, $\omega = \sqrt{(\omega_x^2 + \omega_y^2 + \omega_z^2)}$ 。再将 (3) 式进行归一化处理得

$$q_j(t_{i+1}) = q_j(t_i) / \sqrt{\sum_{n=0}^3 q_n^2(t_{i+1})} \dots \quad j = 0, 1, 2, 3 \quad (4)$$

若安装在体系的三个加速度计在采样区间 (t_i, t_{i+1}) 内输出的视速度增量为 $(W_{x1} \quad W_{y1} \quad W_{z1})$, 为在导航坐标系中进行导航计算, 需将体系的视速度增量转换为导航坐标系下的视速度增量, 转换公式为:

$$\begin{bmatrix} W_x \\ W_y \\ W_z \end{bmatrix} = \frac{1}{2} [C_b^c(t_{i+1}) + C_b^c(t_i)] \begin{bmatrix} W_{x1} \\ W_{y1} \\ W_{z1} \end{bmatrix} \quad (5)$$

再由 (W_x W_y W_z) 解出运动体相对导航坐标系的速度和位置,此不赘述。公式(5)表明,导航计算还要求出方向余弦矩阵 C_b^l 。另外,在进行制导计算中,需要给出运动体的指令姿态,而指令姿态很难用四元数表示,往往也是由方向余弦矩阵给出。上述分析说明,在导航、制导中方向余弦矩阵是必须用的,而四元数只是一个中间变量,不是必须的。本文试图跨越四元数直接导出方向余弦矩阵更新的递推算法。

1 方向余弦矩阵递更新推算法推导

可以导出方向余弦矩阵满足如下的微分方程

$$C_b^l(t) = C_b^l(t) [\quad \mathbf{x}] \quad (6)$$

式中 $[\quad \mathbf{x}] = \begin{bmatrix} 0 & -z_1 & y_1 \\ z_1 & 0 & -x_1 \\ -y_1 & x_1 & 0 \end{bmatrix}$, x_1 、 y_1 和 z_1 分别为飞行器旋转角速度在飞行器体坐标系 x_1 、 y_1 和 z_1 轴的分量,若在积分区间内旋转角速度的方向不变时,有

$$C_b^l(t_{i+1}) = C_b^l(t_i) e^{[\quad \mathbf{x}]} \quad (7)$$

式中

$$[\quad \mathbf{x}] = \int_{t_i}^{t_{i+1}} [\quad \mathbf{x}] dt = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix}$$

根据定义有

$$e^{[\quad \mathbf{x}]} = \sum_{n=0}^{\infty} \frac{[\quad \mathbf{x}]^n}{n!} = I + [\quad \mathbf{x}] + \frac{1}{2!} [\quad \mathbf{x}]^2 + \frac{1}{3!} [\quad \mathbf{x}]^3 + \dots$$

因

$$[\quad \mathbf{x}]^2 = \begin{bmatrix} -\frac{2}{y_1^2} - \frac{2}{z_1^2} & x_1 & y_1 & x_1 & z_1 \\ x_1 & y_1 & -\frac{2}{x_1^2} - \frac{2}{z_1^2} & y_1 & z_1 \\ x_1 & z_1 & y_1 & z_1 & -\frac{2}{x_1^2} - \frac{2}{y_1^2} \end{bmatrix}$$

$$[\quad \mathbf{x}]^3 = -\frac{2}{0} [\quad \mathbf{x}], \frac{2}{0} = \frac{2}{x_1} + \frac{2}{y_1} + \frac{2}{z_1}$$

$$[\quad \mathbf{x}]^4 = -\frac{2}{0} [\quad \mathbf{x}]^2, [\quad \mathbf{x}]^5 = -\frac{4}{0} [\quad \mathbf{x}]$$

...

所以 $e^{[\quad \mathbf{x}]}$ 可以改写为

$$e^{[\quad \mathbf{x}]} = I + \frac{[\quad \mathbf{x}]}{0} [\quad \mathbf{x}] + \frac{1}{3!} [\quad \mathbf{x}]^3 + \frac{1}{5!} [\quad \mathbf{x}]^5 - \dots +$$

$$\frac{[\quad \mathbf{x}]^2}{2} [1 - (1 - \frac{1}{2!} \frac{2}{0} + \frac{1}{4!} \frac{4}{0} - \dots)] = I + \frac{\sin_0}{0} [\quad \mathbf{x}] + \frac{1 - \cos_0}{2} [\quad \mathbf{x}]^2 \quad (8)$$

可以证明式(8)所表示矩阵为正交矩阵,为获得简化的递推公式形式,再令 $[\quad \mathbf{x}] = [\quad \mathbf{x}] / 0$, 将(8)式改写为:

$$e^{[\quad \mathbf{x}]} = I + \sin_0 [\quad \mathbf{x}] + (1 - \cos_0) [\quad \mathbf{x}]^2 = I \cos_0 + \sin_0 [\quad \mathbf{x}] + (1 - \cos_0) \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (9)$$

式中 $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = [\quad \mathbf{x}]^T / 0$ 。记

$$C_b^l(t) = [C_{ij}], i, j = 1, 2, 3 \quad (10)$$

及公式(9)、(10)代入公式(6)得:

$$C_b^l(t_{i+1}) = C_b^l(t_i) \cos_0 + \sin_0 C_b^l(t_i) [\quad \mathbf{x}] + (1 - \cos_0) \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (11)$$

式中 $b_i = \sum_{j=1}^3 C_{ij} \frac{0}{j}$ 、 $\frac{0}{1} = \frac{0}{x_1}$ 、 $\frac{0}{2} = \frac{0}{y_1}$ 和 $\frac{0}{3} = \frac{0}{z_1}$ 。记 $C_b^l(t_i + 1) = [C_{ij}], i, j = 1, 2, 3$, 则

$$C_{ij} = [C_{i,j+1} \frac{0}{j+2} - C_{i,j+2} \frac{0}{j+1}] \sin_0 + C_{ij} \cos_0 + b_i \frac{0}{j} (1 - \cos_0) \quad (12)$$

又因

$$\begin{cases} \sin_0 = 0 (1 - \frac{2}{6} + \frac{4}{120} \dots) \\ 1 - \cos_0 = \frac{2}{2!} - \frac{4}{4!} + \frac{6}{6!} \dots \end{cases} \quad \text{将(12)式改写为:}$$

$$C_{ij} = C_{ij} a_{12} + b_i \frac{0}{j} a_3 + [C_{i,j+1} \frac{0}{j+2} - C_{i,j+2} \frac{0}{j+1}] a \quad (13)$$

式中

$$\begin{cases} 1 = \frac{2}{x_1} + \frac{2}{y_1} + \frac{2}{z_1} \\ a_1 = 1 - a_3 \\ a_2 = 1 - \frac{1}{6} + \frac{2}{120} \\ a_3 = \frac{1}{2} - \frac{1}{24} + \frac{2}{720} \\ b_i = \sum_{j=1}^3 C_{ij} \frac{0}{j} \\ 1 = \frac{4}{x_1}, \frac{2}{2} = \frac{5}{y_1} \\ C_{i,4} = C_{i,1}, C_{i,5} = C_{i,2} \end{cases} \quad (14)$$

在 \sin_0, \cos_0 的级数展开形式中忽略了 $\frac{7}{0}$ 以上的

阶项。式(13)、(14)便是计算方向余弦矩阵的递推公式,(13)式与(1)~(4)式相比结构简单、有利于软件编写。

2 数学仿真计算及误差分析

2.1 导航验证

飞行距离为 Xkm 和 Ykm 的全程惯性制导飞行器为例,使用直接方向余弦矩阵递推算法和通过四元数传递求解方向余弦矩阵两种导航方法分别进行导航计算。其中,导航周期为 $t_{i+1} - t_i = 10ms$ 。

图 1~3 是飞行距离为 Xkm 飞行器姿态角与方向余弦矩阵所确定的姿态角对应量之差。其中,下标“n”为方向余弦矩阵所对应的姿态角,下标“Old”为通过四元数传递得到的方向余弦矩阵,下标“New”为直接使用方向余弦矩阵传递公式得到的方向余弦矩阵。通过图示可以得出:在 $0s \sim 150s$ 和 $1026s \sim 1126s$,使用直接求解方向余弦矩阵的方法得到的俯仰角更接近真实值,其余飞行时间段内使用四元数间接求解方向余弦矩阵得到的俯仰角比较接近真实俯仰角;偏航和滚动姿态偏差量级和变化规律基本一致。

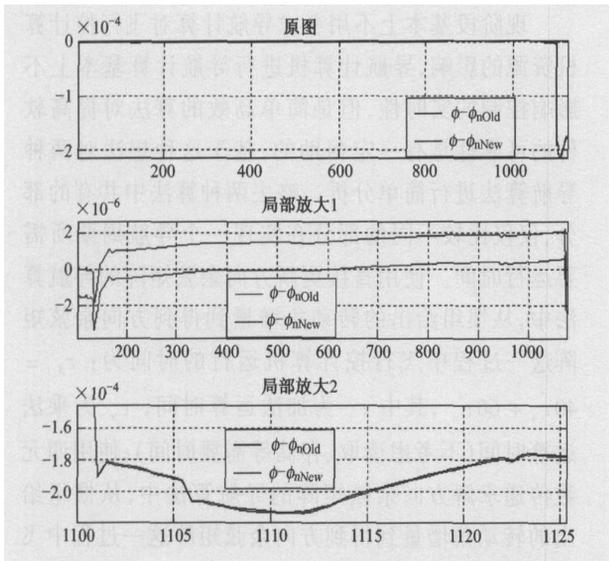


图 1 飞行距离 Xkm 两种方向余弦矩阵算法对应的俯仰姿态角偏差

Fig.1 To compare the performance of the pitch drift error of direct cosine matrix algorithm with the performance that exist in quaternion algorithm

表 1 和表 2 分别为飞行距离 Xkm 和 Ykm 的飞行器在一定时间内导航,统计飞行器在该时刻通过 Runge-Kutta45 算法得到的方向余弦矩阵 9 个量与导

航中的方向余弦对应量之差,同时给出此时不同导航方式得到最大位置和最大速度偏差。表中下标“g”表示是通过飞行器质心运动方程和绕质心转动的运动方程积分给出的;下标“n”表示是通过捷联惯性导航给出的。

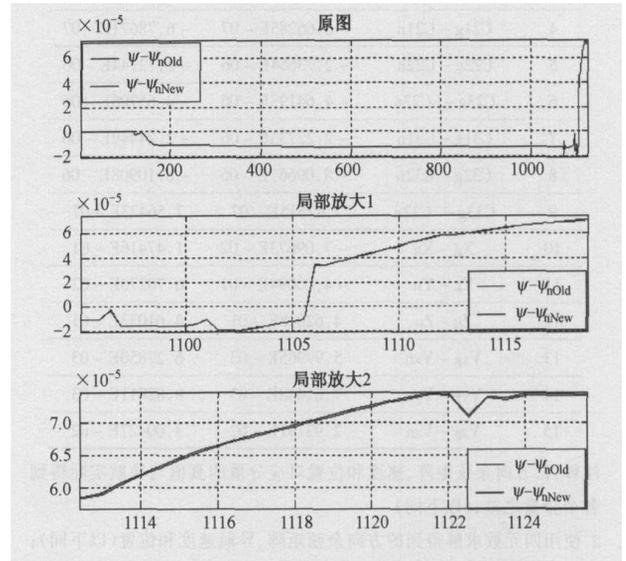


图 2 飞行距离 Xkm 两种方向余弦矩阵算法对应的偏航姿态角偏差

Fig.2 To compare the performance of the yaw drift error of direct cosine matrix algorithm with the performance that exist in quaternion algorithm

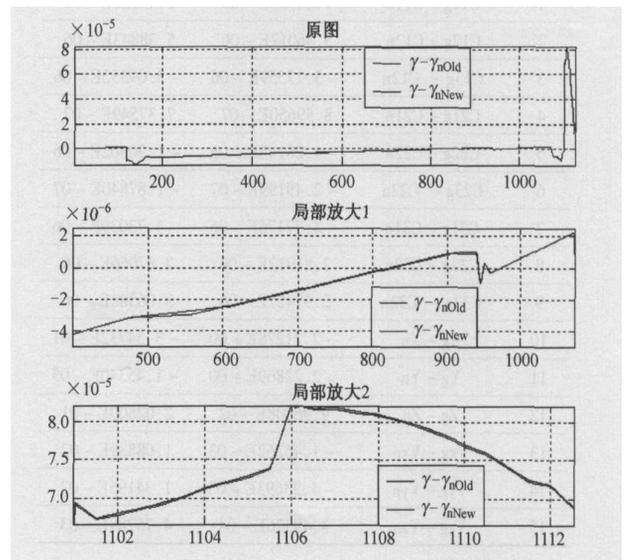


图 3 飞行距离 Xkm 两种方向余弦矩阵算法对应的滚动姿态角偏差

Fig.3 To compare the performance of the roll drift error of direct cosine matrix algorithm with the performance that exist in quaternion algorithm

表 1 仿真情况下导航精度(飞行距离 X km)

Table 1 Simulated Navigational Accuracy (Range X)

序号	比较项 ¹	方法 1 ²	方法 2 ³
1	C11g - C11n	- 3.62825E- 07	- 3.41788E- 07
2	C12g - C12n	- 8.37996E- 07	- 8.24170E- 07
3	C13g - C12n	7.55900E- 06	7.56064E- 06
4	C21g - C21n	- 6.66285E- 07	- 6.78671E- 07
5	C22g - C22n	- 2.73654E- 06	- 2.72144E- 06
6	C23g - C22n	- 4.64193E- 06	- 4.65706E- 06
7	C31g - C31n	- 2.22733E- 06	- 2.24449E- 06
8	C32g - C32n	- 8.09667E- 06	- 8.10908E- 06
9	C33g - C32n	7.42515E- 07	7.56423E- 07
10	Xg - Xn	- 7.09673E- 02	1.47416E- 01
11	Yg - Yn	- 4.00094E- 01	6.70270E- 02
12	Zg - Zn	4.62018E- 01	4.61033E- 01
13	Vxg - Vxn	5.99965E- 03	6.27850E- 03
14	Vyg - Vyn	7.61443E- 03	9.82811E- 03
15	Vzg - Vzn	2.92191E- 02	3.00427E- 02

注释:1. 方向余弦矩阵、速度和位置对应分量的真值与导航实际得到各个分量之差;(以下同)

2. 使用四元数求解得到的方向余弦矩阵、导航速度和位置(以下同);

3. 使用方向余弦矩阵直接求解得到的方向余弦矩阵、导航速度和位置(以下同)。

表 2 仿真情况下导航精度(飞行距离 Y km)

Table 2 Simulated Navigational Accuracy (Range Y)

序号	比较项 ¹	方法 1 ²	方法 2 ³
1	C11g - C11n	1.40996E- 06	1.49128E- 06
2	C12g - C12n	5.36017E- 06	5.38843E- 06
3	C13g - C12n	- 3.17359E- 06	- 3.09352E- 06
4	C21g - C21n	8.59650E- 07	7.37549E- 07
5	C22g - C22n	- 1.28143E- 06	- 1.30502E- 06
6	C23g - C22n	- 2.49199E- 07	- 1.87840E- 07
7	C31g - C31n	- 4.77176E- 06	- 4.73039E- 06
8	C32g - C32n	2.81912E- 06	2.87066E- 06
9	C33g - C32n	2.97427E- 06	3.06388E- 06
10	Xg - Xn	- 2.21278E+00	- 3.54732E- 01
11	Yg - Yn	- 2.22869E+00	- 1.45340E- 03
12	Zg - Zn	9.96808E- 02	2.02800E- 01
13	Vxg - Vxn	- 1.46262E- 03	1.08839E- 02
14	Vyg - Vyn	- 4.23693E- 03	1.34195E- 02
15	Vzg - Vzn	3.49376E- 04	4.75788E- 03

2.2 余弦矩阵求解精度和导航算法精度分析

通过数学仿真可以看出,直接使用方向余弦矩阵对姿态进行更新的方法可以满足导航算法的误差要求,特别是在长时间、高动态的情况下导航算法的误差稳定性较好,其中导航精度位置控制在 10^{-1} m 量级,速度控制在 10^{-2} m/s 量级,而且导航姿态的误

差,即方向余弦矩阵的计算误差,经过长时间的累积,其分量中的最大误差不大于 10^{-5} 。与此同时使用四元数间接求解方向余弦矩阵的方法进行导航得到的最大位置误差因飞行距离不同而略有差别,在飞行距离为 Xkm 的时候单方向最大位置量级为 10^{-1} m,在飞行距离为 Ykm 的时候单方向最大位置误差为 10^{-0} m,经过长时间的方向余弦矩阵误差积累,两种飞行距离的单方向最大值偏差量级均为 10^{-5} 。通过比较可以得出:使用四元数间接求解方向余弦矩阵得到导航误差稳定性略差,而两种方法求解捷联惯性导航中的方向余弦矩阵最大误差量级是相当的,都满足工程要求。

出现上述情况的原因大致有:工程算法中对刚体转到假设为在短时间内认为转动角速度方向不变,但是实际上,特别是飞行器在有推力且高速自旋的情况下,转动角速的为常量的假设往往不成立,这也造成一部分误差产生;同时在推导公式时对高阶小量的忽略也容易引入一部分误差;在同一种飞行环境情况下,不同算法对方向余弦矩阵分量的影响不同造成导航算法精度影响不同。

2.3 两种导航算法实现对比分析

现阶段基本上不用考虑导航计算对飞行控计算机资源的影响,导航计算机进行导航计算基本上不影响控制的实时性,但是简单高效的算法对提高软件的可靠性是有一定帮助的,基于这种想法对两种导航算法进行简单分析。略去两种算法中共有的部分,仅仅比较不同的部分在实现一个导航周期所需要运行时间。使用直接更新方向余弦矩阵的导航算法中,从惯组给出的转动角增量到得到方向余弦矩阵这一过程中飞行控计算机运行的时间为: $A = 40 + + 66 \times$, 其中 $+$ 为加法运算时间, \times 为乘法运算时间(不考虑读取、存储等需要时间);使用四元数传递求解方向余弦矩阵的导航算法中,从惯组给出的转动角增量到得到方向余弦矩阵这一过程中飞行控计算机运行的时间为: $B = 34 + + 59 \times + \sqrt{+ 4 +}$, 其中 $\sqrt{+ 4 +}$ 为开方运算时间($\sqrt{1 + (x - 1)}$ 用级数展开, x 接近 1,忽略 6 阶以上的项,可以得到 $\sqrt{+ 4 +} = 8 + + 13 \times$), \div 为除法运算时间($\div = \times$), 则有 $B = 42 + + 76 \times$ 。显见四元数传递求解方向余弦矩阵的计算机运算时间大于直接求解方向余弦矩阵的导航算法运行时间。另外,方向余弦矩阵更新的递推算算法编程容易实现、代码少的优点。

3 结论

仿真计算表明方向余弦矩阵的递推算法求解公式推导正确,使用该算法进行导航计算满足导航精度要求。它具有计算量小、编程简单的优点。该方法在捷联导航与制导应用中完全可以取代四元数方法。

参考文献:

- [1] Titterton D H and Weston J L. Strapdown Inertial Navigation Technology, Peter Peregrinus Ltd. on behalf Institution of Electrical Engineers 1988:39 - 56
- [2] 陈世年. 控制系统设计[M]. 北京:宇航出版社,1996,6:36 - 60 [CHEN Shi-nian. Design of the Control System [M]. Beijing:

Astronautic Publishing Company,1996,6:36 - 60]

- [3] Garth W. Milne. Simplifying vector rotations and coordination, AIAA 2001 - 4310 ,A01 - 37272 ,1 - 7
- [4] Purivigraipong. Spacecraft attitude and line bias estimations from phase information of global positioning system signals[J]. IEEE, 2004,4, (2729)



作者简介:李连仲(1935 -) ,男,研究员,主要研究方向为星光 - 惯性复合制导技术、闭路制导技术和能量管理技术等。
通信地址:北京控制与电子技术研究所(100038)
电话:(010)63322961

A New Recursion Formula Used In The Matrix of the Direction Cosine In The Strapdown Inertial Navigation and Guidance System

LI Lian-zhong , WANG Xiao-hu , CAI Shu-jiang
(Beijing Institute of Control And Electronic Technology , Beijing 100038 , China)

Abstract: Quaternion method , which has been successfully applied to the Strapdown Inertial Navigation and Guidance for tens year , is classical . In general , the practical using of the direction cosine matrix relating body frame to navigation frame is essential . However , the quaternion , which is only impermanent variable , can be leaved out . So the paper proposes a new recursion algorithm for the direction cosine matrix without quaternion . Mathematical simulation shows that the algorithm based on the differential equation of the matrix of the direction cosine is comprehensive , simple , and calculating time is shorter .

Key words: Strapdown inertial navigation ; The direction cosine matrix ; Quaternion

(上接第 344 页)

Direct Adaptive Trajectory Linearization Control of Aerospace Vehicle Using SHLNN

ZHU Liang , JIANG Chang-sheng , CHEN Hai-tong , FANG Wei
(College of Automation Engineering , Nanjing University of Aeronautics and Astronautics , Nanjing 210016 , China)

Abstract: This paper presents a novel nonlinear adaptive control method based on trajectory linearization control method (TLC) and neural networks. TLC method is a novel nonlinear tracking and decoupling control way. However , inherent uncertainties may render it useless. In this paper , a neural network is used to cancels uncertainties through on-line learning. The adaptive law is designed based on Lyapunov approach , so all signal boundedness of the whole system is guaranteed. Finally , the flight control system of the aerospace vehicle is designed based on the proposed method and the simulation results demonstrate the excellent performance and robustness of the controllers.

Key words: Flight control system ; Trajectory linearization control ; Neural networks ; Adaptive control