

基于四元数法飞行运动方程的逆向仿真算法

薛源, 徐浩军, 胡孟权, 郭辉, 柴世杰

(空军工程大学 工程学院, 陕西 西安 710038)

摘要:逆向仿真算法对于已知飞行状态的验证具有重要的意义。为了避免奇异性,采用四元数法构建了飞机的六自由度运动方程,并进行了四元数方程的仿真分析;介绍了逆向仿真算法的思路,讨论了仿真中关键参数的确定方法;对根据三自由度方程建立的飞机规避导弹时的飞行轨迹进行逆向反推,得出了产生此轨迹需要的输入变量的变化过程,将得出的输入变量作用于基于四元数法的六自由度运动方程中,获得其逆向仿真轨迹。通过对逆向仿真轨迹和原轨迹的比较验证了原飞行轨迹的合理性。

关键词:四元数;六自由度;逆向仿真;轨迹优化

DOI:10.3969/j.issn.1009-3516.2010.03.002

中图分类号: V211 **文献标识码:** A **文章编号:** 1009-3516(2010)03-0006-05

传统的仿真方法一般是根据每个时间点的输入状态求解系统的动态方程来得出一段时间内的响应,从而仿真出系统的响应过程^[1-3]。但这种方法或多或少都存在着和真实情况的误差,而研究合理的仿真真实性验证技术一直是仿真领域的难点。逆向仿真算法是众多验证技术中的一种^[4-5],一般作为传统仿真方法的补充算法,可以计算已知输出信号的控制行为,亦可以将典型的过程信号进行逆向处理从而计算出要达到特定响应的输入量和在该输入量作用下的输出。

本文使用逆向仿真算法验证了飞行运动仿真中飞行轨迹的真实性问题,根据已知的输出信号来确定产生信号的控制状态变量。然后得出逆向的输出状态,用输出状态和原输出信号相比较,从而来验证控制信号的可行性和原输出信号的合理性。

1 基于四元数法的飞机本体运动方程建模

常规的飞机运动方程是用欧拉角来定义的^[1,6],但这种欧拉方程有明显的缺点,当俯仰角 ϑ 经过 $\pm\pi/2$ 时,滚转角速度 p 和偏航角速度 r 的微分方程将出现奇异性。为了避免这种奇异性,采用四元数法来描述飞机的运动方程^[7]。

1.1 四元数方程的建立和主要参数的确定

在机体坐标系中建立的四元数描述的飞机运动方程如下^[6-7]:

$$\begin{cases} \dot{u} = rv - qw + \frac{1}{m}(\bar{X} + F_T) + 2(q_1q_3 - q_0q_2)g \\ \dot{v} = pw - ru + \frac{1}{m}\bar{Y} + 2(q_2q_3 - q_0q_1)g \\ \dot{w} = qu - pv + \frac{1}{m}\bar{Z} + (q_0^2 - q_1^2 - q_2^2 - q_3^2)g \end{cases} \quad (1)$$

收稿日期:2009-10-29

基金项目:国家自然科学基金资助项目(70671106)

作者简介:薛源(1986-),男,山东菏泽人,博士生,主要从事飞行仿真与飞行安全研究;E-mail:szxy1986@163.com
徐浩军(1965-),男,浙江余姚人,教授,博士生导师,主要从事飞行安全及作战效能研究。

$$\begin{cases} \dot{p} = (c_1 r + c_2 p)q + c_3 \bar{L} + c_4 (\bar{N} + h_E q) \\ \dot{q} = c_5 p r - c_6 (p^2 - r^2) + c_7 (\bar{M} + F_T z_T - h_E r) \\ \dot{r} = (c_8 p - c_2 r)q + c_4 \bar{L} + c_9 (\bar{N} + h_E q) \end{cases} \quad (2)$$

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -p & -q & -r \\ p & 0 & r & -q \\ q & -r & 0 & p \\ r & q & -p & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} \dot{x}_E \\ \dot{y}_E \\ \dot{z}_E \end{bmatrix} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (4)$$

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \pm \begin{bmatrix} \cos(\varphi/2) \cos(\vartheta/2) \cos(\psi/2) + \sin(\varphi/2) \sin(\vartheta/2) \sin(\psi/2) \\ \sin(\varphi/2) \cos(\vartheta/2) \cos(\psi/2) - \cos(\varphi/2) \sin(\vartheta/2) \sin(\psi/2) \\ \cos(\varphi/2) \sin(\vartheta/2) \cos(\psi/2) + \sin(\varphi/2) \cos(\vartheta/2) \sin(\psi/2) \\ \cos(\varphi/2) \cos(\vartheta/2) \sin(\psi/2) - \sin(\varphi/2) \sin(\vartheta/2) \cos(\psi/2) \end{bmatrix} \quad (5)$$

式中 $\begin{cases} \bar{X} = \bar{q} S C_{X_T}(\alpha, \beta, p, q, r, \delta, \dots) \\ \bar{Y} = \bar{q} S C_{Y_T}(\alpha, \beta, p, q, r, \delta, \dots) \\ \bar{Z} = \bar{q} S C_{Z_T}(\alpha, \beta, p, q, r, \delta, \dots) \end{cases}, \begin{cases} \bar{L} = \bar{q} S b C_{l_T}(\alpha, \beta, p, q, r, \delta, \dots) \\ \bar{M} = \bar{q} S b C_{m_T}(\alpha, \beta, p, q, r, \delta, \dots) \\ \bar{N} = \bar{q} S b C_{n_T}(\alpha, \beta, p, q, r, \delta, \dots) \end{cases}$; F_T 为发动机在体轴系 x 轴上的推力; h_E

为发动机沿体轴系 x 方向的角动量,涉及的参数为: $c_1 = \frac{(I_y - I_z)I_z - I_{xz}^2}{I_x I_z - I_{xz}^2}, c_2 = \frac{(I_x - I_y + I_z)I_{xz}}{I_x I_z - I_{xz}^2}, c_3 = \frac{I_z}{I_x I_z - I_{xz}^2}, c_4 = \frac{I_{xz}}{I_x I_z - I_{xz}^2}, c_5 = \frac{I_z - I_x}{I_y}, c_6 = \frac{I_{xz}}{I_y}, c_7 = \frac{1}{I_y}, c_8 = \frac{I_x(I_x - I_y) + I_{xz}^2}{I_x I_z - I_{xz}^2}, c_9 = \frac{I_x}{I_x I_z - I_{xz}^2}$, 具体意义见文献[6-7]。

1.2 基于四元数的飞行运动仿真

根据构建的飞机运动方程(式1-5),建立其飞行控制系统。在SIMULINK仿真环境下建立的飞机电传操作系统的框图见图1。在 $H = 6000 \text{ m}, Ma = 0.6$ 的情况下,控制变量中平尾偏角 $\delta_e = 0.17 \text{ rad}$,副翼偏角 $\delta_a = 0.5 \text{ rad}$,仿真时间设置为 $t = (0 - 50) \text{ s}$ 。仿真得到四元数 (q_0, q_1, q_2, q_3) 的状态见图2, (α, β, V_T) 的状态见图3, (p, q, r) 的状态见图4。

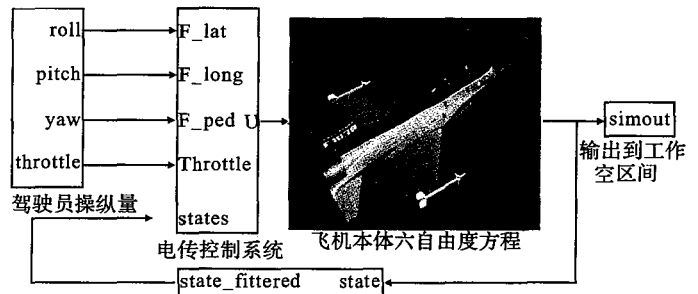


图1 飞机操作系统 SIMULINK 框图

Fig.1 The simulink model of flying control system

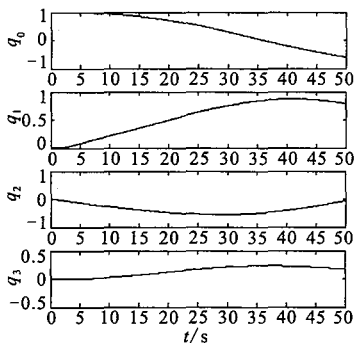


图2 (q_0, q_1, q_2, q_3) 仿真曲线
Fig.2 The simulation of (q_0, q_1, q_2, q_3)

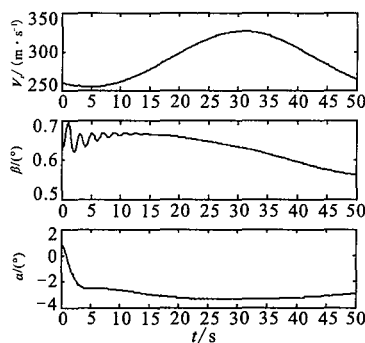


图3 (α, β, V_T) 仿真曲线
Fig.3 The simulation of (α, β, V_T)

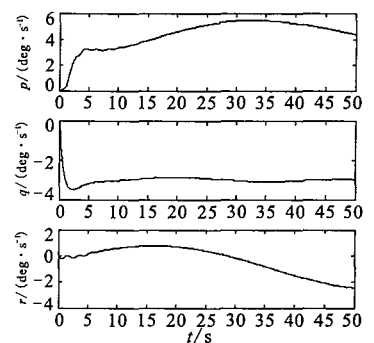


图4 (p, q, r) 仿真曲线
Fig.4 The simulation of (p, q, r)

2 逆向仿真的步骤及具体参数的设置

2.1 逆向仿真算法的具体步骤

逆向仿真算法是一种基于积分形式的算法^[8-9]:

1) 设置算法开始时间 $t=0$;

2) 设置初始的状态变量 x_0 和初始的输入变量 u_0 ;

3) 估计第 k 个时间点上的第 n 次估计变量 $\hat{x}(t_{k+1})_n$, 可以通过 $x(t_{k+1})$ 和当前的迭代估计变量 $u(t_k)_n$ 来计算:

$$\hat{x}(t_{k+1})_n = f(x(t_{k+1}), u(t_k)_n) \quad (6)$$

4) 使用 Runge - Kutta 积分方法对 $\hat{x}(t_k)_n$ 进行积分来得出下一个时间点处的状态变量 $x(t_{k+1})_n$ 和输出变量 $y(t_{k+1})_n$:

$$x(t_{k+1})_n = x(t_k)_n + \int_{t_k}^{t_{k+1}} \dot{x}(t_k) dt_n \quad (7)$$

$$y(t_{k+1})_n = g(x(t_{k+1})_n) \quad (8)$$

由于需满足的输出量 $y_{des}(t_{k+1})$ 是已知的, 误差函数可以定义为:

$$e(t_{k+1}) = y_{des}(t_{k+1}) - y(t_{k+1})_n \quad (9)$$

5) 如果 $e(t_{k+1}) \leq \varepsilon$, 则进行到下一个时刻 $k+1$, 算法流程转到步骤 3);

6) 如 $e(t_{k+1}) > \varepsilon$ 使用牛顿迭代法进行迭代计算直至误差 $e(t_{k+1})$ 满足要求。迭代公式为:

$$u(t_k)_{n+1} = u(t_k)_n - J^{-1} e(t_{k+1})_n w \quad (10)$$

式中雅可比矩阵 J 描述了输出量对控制量的变化率。逆向仿真的流程图见图 5。

2.2 雅可比矩阵 J 的确定

设定义在时间点 k 处的第 n 次迭代的雅可比矩阵 J 中的元素为 $j_{ij}(t_k)_n$, 可由输出量 $y_i(t_k)_n$ 对控制量 $u_j(t_k)_n$ 的微分方程来表示^[9]:

$$j_{ij}(t_k)_n = \frac{\partial y_i(t_{k+1})_n}{\partial u_j(t_k)_n} \quad (11)$$

为了获得数据的完整性, 采用数值方法取代解析的方法对该微分方程进行计算, 公式如下涉及的控制量有: 推力设置 δ_{PLA} , 平尾偏角 δ_e , 副翼偏角 δ_a , 方向舵偏角 δ_r , 前缘襟翼偏角 δ_{LEF} ; 输出量有: 马赫数 Ma , 真速 V_T , 过载 n_z , 滚转角 φ , 俯仰角速率 q , 滚转角速率 p 。

$$\frac{\partial y_i(t_{k+1})_n}{\partial u_j(t_k)_n} = \frac{y_i(t_{k+1}, u_j(t_k) + \delta u_j(t_k))_n - y_i(t_{k+1}, u_j(t_k) - \delta u_j(t_k))_n}{2\delta u_j(t_k)_n} \quad (12)$$

最终的雅可比矩阵公式为:

$$\begin{array}{l} \delta_{PLA} \quad \delta_e \quad \delta_a \quad \delta_r \quad \delta_{LEF} \\ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ V_T \rightarrow \\ n_z \rightarrow \\ p \rightarrow \\ q \rightarrow \\ \varphi \rightarrow \end{array} \begin{bmatrix} J_{1,1} & J_{1,2} & J_{1,3} & J_{1,4} & J_{1,5} \\ J_{2,1} & J_{2,2} & J_{2,3} & J_{2,4} & J_{2,5} \\ J_{3,1} & J_{3,2} & J_{3,3} & J_{3,4} & J_{3,5} \\ J_{4,1} & J_{4,2} & J_{4,3} & J_{4,4} & J_{4,5} \\ J_{5,1} & J_{5,2} & J_{5,3} & J_{5,4} & J_{5,5} \end{bmatrix} = J \quad (13)$$

2.3 每步的迭代计算时间选取

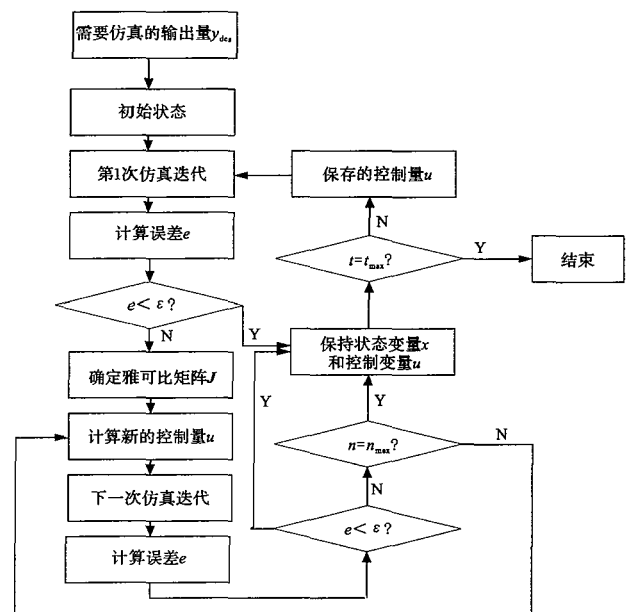


图 5 逆向仿真算法流程图

Fig. 5 The Inverse simulation flow chart

迭代计算时间 Δt 对逆向仿真算法的稳定性和精确性有重要的影响^[7],当 Δt 的值过大时,受系统的物理动态特性的影响,系统的响应会产生不稳定的振荡,此时,可以对输出状态增加高频滤波器来过滤系统的振荡现象;当 Δt 减小时可以使振荡的幅值减小,但太小的 Δt 又会增加系统的动态误差,所以选择合适的 Δt 的值是必要的。 Δt 的选取范围一般为(0.05 - 0.5)s。本文中的逆向仿真算法没有采用滤波器,并且通过尽量增大 Δt 的值来缩短计算时间。

在 Δt 内的迭代次数不能够太大,一般要在5到10次迭代中获得合适的控制变量 u ,最多不能超过50次迭代。这种限制可以使算法具有很好的鲁棒性。如果符合误差条件的控制变量 u 在规定的限制条件下没有找到。逆向仿真算法将会使用迭代过程中最合适的 u 值进行下一步的计算。

3 仿真实例分析

设置原仿真轨迹获取方案为:飞机通过使逃脱距离最大化来试图规避导弹的攻击,导弹采用单纯的追踪控制模式。基于飞机三自由度运动方程得出原仿真轨迹。

飞机以平飞状态开始逆向仿真计算,设置 $H = 6\ 000, Ma = 0.8$ 。选取 V_T, n_z, φ 作为要辨识的输出状态。如果采用滚转速率 p 作为待辨识参数,在参数 p 的逆向仿真过程中会出现振荡现象,需要设计滤波器来过滤出现的振荡环节,增加了仿真算法的复杂性。而采用滚转角 φ 作为待辨识参数,振荡现象基本上消除,得到和最优的轨迹尽可能相似的结果。

迭代计算时间 Δt 取为 0.25 s。忽略发动机的瞬时变化,逆向仿真的结果和轨迹优化算法计算出的结果对比见图6-7。inv代表逆向仿真的参数,图6中参数 V_T 已缩小150倍,使其能和过载 n_z 使用同一坐标系。

根据仿真的结果可以看出,飞机主要参数 $V_T, n_z, \varphi, \vartheta, \psi$ 辨识的精度很高。在 n_z 仿真曲线上升段会出现一些小的偏差,但在可接受的误差范围内。总的来说逆向仿真的结果和原仿真结果的拟配精度完全满足系统辨识的精度要求。

基于飞机三自由度运动方程的算法得出的原轨迹和逆向仿真的轨迹见图8。通过图8可以看出2条轨迹线有很高的相似程度,从而也说明了逆向仿真算法检验飞行轨迹控制规律的合理性与精确性。

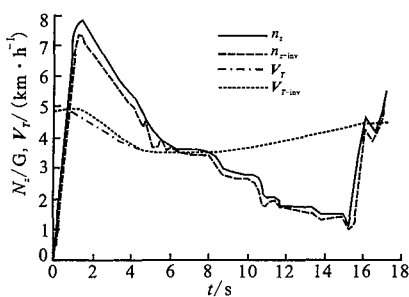


图6 法向过载及速度仿真曲线

Fig. 6 The simulation of overload and velocity

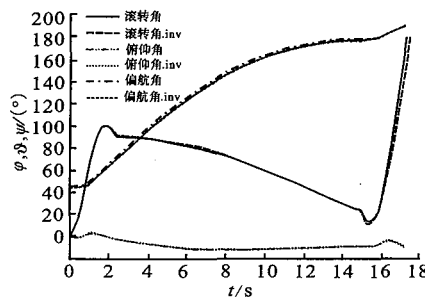


图7 欧拉角仿真曲线

Fig. 7 The simulation of Euler angle

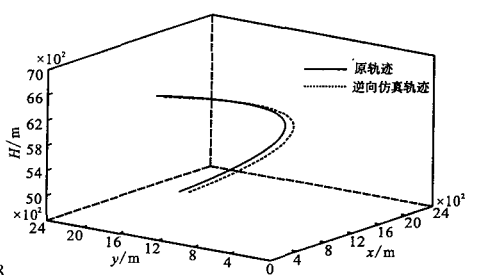


图8 原仿真轨迹及逆向仿真轨迹

Fig. 8 The trajectory of the original simulation and the inverse simulation

4 结束语

本文采用逆向仿真过程来检验仿真飞行轨迹真实性,将基于积分形式的逆向仿真算法应用到飞机的六自由度四元数方程中,并对三自由度建立的规避导弹时的飞行轨迹进行了逆向仿真验证,在对逆向仿真算法辨识出的参数和轨迹进行分析后可以看出原轨迹更符合真实状况,从而也进一步验证了该算法的合理性。

参考文献:

[1] 高金源,李陆毅,冯亚昌,等.飞机飞行品质[M].北京:国防工业出版社,2001.
GAO Jinyuan, LI Luyi, FENG Yachang, et al. Aircraft Handling Qualities[M]. Beijing: National Defense Industry Press, 2001. (in Chinese)

[2] 飞机飞行性能计算手册编写组.飞机飞行性能计算手册[M].西安:飞行力学杂志社,1987.

- Aircraft Flight Performance Reckoner Edition Board. Aircraft Flight Performance Reckoner[M]. Xi'an, Flight Dynamics Journal Press, 1987. (in Chinese)
- [3] 葛志浩,徐浩军. 先进战斗机人——机闭环系统仿真研究[J]. 弹箭与制导学报, 2004, 24(3): 389-391.
GE Zhihao, XU Haojun. MATLAB Simulation Model of Closed-loop Aircraft-pilot System[J]. Journal of Projectiles Rockets Missiles and Guidance, 2004, 24(3): 389-391. (in Chinese)
- [4] Hoffren J, Salaranta T. Maneuver Autopilot for Realistic Performance Model Simulations[J]. AIAA 2001-4376.
- [5] Gao C, Hess R A. Inverse Simulation of Large-amplitude Aircraft Maneuvers[J]. AIAA Journal of Guidance, Control and Dynamics, 1992, 16(4): 733-737.
- [6] 肖业伦. 飞行器运动方程[M]. 北京:航空工业出版社, 1987.
XIAO Yelun. Aircraft Motion Equation[M]. Beijing: Aviation Industry Press, 1987. (in Chinese)
- [7] 方振平,陈万春,张曙光. 航空飞行器飞行动力学[M]. 北京:北京航空航天大学出版社, 2005.
FANG Zhenping, CHEN Wanchun, ZHANG Shuguang. Aircraft Flight Dynamics[M]. Beijing: Beijing University of Aeronautics and Astronautics Press, 2005. (in Chinese)
- [8] Russel R S. Nonlinear F-16 Simulation Using Simulink and Matlab[R]. Minnesota: University of Minnesota, 2003.
- [9] Karelaiti J, Virtanen K, Raivio T. Near-optimal Missile Avoidance Trajectories via Receding Horizon Control [R]. AIAA 2006-6312.

(编辑:徐敏)

The Inverse Simulation Method Based on Equations of Flying Motion Using Quaternion Orientation

XUE Yuan, XU Hao-jun, HU Meng-quan, GUO Hui, CHAI Shi-jie
(Engineering Institute, Air Force Engineering University, Xi'an 710038, China)

Abstract: Inverse simulation is important in validating the known flying state. In the process of establishing equations, quaternion is used to set up six degree-of-freedom(6-DOF) flight model in order to avoid singularities, the flight simulation based on quaternion is also performed. The general method of inverse simulation is introduced; the means of identifying key parameters in the inverse simulation is discussed. The 3-DOF model trajectory acquired by avoiding a missile is inversely simulated to get the control rule for producing the initial trajectory, the inversely simulated trajectory is gained by running the 6-DOF model with the obtained control rules. The rationality of the initial optimal trajectory is verified by comparing the inversely simulated trajectory with the initial trajectory.

Key words: quaternion; 6-DOF; inverse simulation; trajectory optimization