

DOI:10.3969/j.issn.1003-5060.2024.01.009

空间运动方程快速求解器设计与实现

王晓蕾, 黄章骞, 房旭, 宋宇鲲

(合肥工业大学 微电子学院, 安徽 合肥 230601)

摘要:文章基于四阶经典龙格库塔法(classical Runge-Kutta method of order four, RK-4)和四阶 Adams 预测校正法(fourth-order Adams predictor-corrector method, Adams-4),提出一种现场可编程逻辑门阵列(field programmable gate array, FPGA)实现的数据路径可动态配置的空间运动方程快速求解器(space motion equation fast solver, SMEFS)。SMEFS 采用折叠式结构,借助高效的映射和精准的状态管理,通过资源复用和动态配置运算器内部连接关系实现数据路径的动态配置,快速求解空间运动方程,并有效节省硬件资源。采用某型运载火箭的相关数据对 SMEFS 进行大批量空间运动方程求解的性能评估,实验结果表明 SMEFS 能够快速可靠地求解发射坐标系下的五自由度空间运动方程,与软件求解的平均加速比为 12.765,求解结果最大相对误差小于 9×10^{-5} ,具备较好的加速效果和较高的计算可靠性。

关键词:四阶经典龙格库塔法(RK-4);四阶 Adams 预测校正法(Adams-4);空间运动方程;折叠技术;现场可编程逻辑门阵列(FPGA)

中图分类号:TN47

文献标志码:A

文章编号:1003-5060(2024)01-0054-08

Design and implementation of a space motion equation fast solver

WANG Xiaolei, HUANG Zhangqian, FANG Xu, SONG Yukun

(School of Microelectronics, Hefei University of Technology, Hefei 230601, China)

Abstract:Based on the classical Runge-Kutta method of order four(RK-4) and the fourth-order Adams predictor-corrector method(Adams-4), a field programmable gate array(FPGA)-implemented space motion equation fast solver(SMEFS) with dynamically configurable data paths is proposed. SMEFS adopts a foldable structure. With the help of efficient task mapping and precise state management, the dynamic configuration of data paths is realized by resource reusing and dynamic configuring connections between operators, the solutions of space motion equations can be obtained quickly by SMEFS, and hardware resources are saved effectively. The experimental results of the performance evaluation of SMEFS for solving large-scale space motion equations using the relevant data of a certain type of launch vehicle show that the solutions of five-degree-of-freedom space motion equations in the launch coordinate system can be obtained quickly and reliably by SMEFS. The average speedup compared to the software solutions is 12.765 and the maximum relative error of the solutions is less than 9×10^{-5} . Therefore, SMEFS has good acceleration effect and high calculation reliability.

Key words:classical Runge-Kutta method of order four(RK-4); fourth-order Adams predictor-corrector method(Adams-4); space motion equation; folding technology; field programmable gate array(FPGA)

空间运动方程(space motion equation, SME)作为一种高阶常微分方程,是研究运载火

箭、近地导弹、近地航天器等传统飞行器轨道力学的关键手段之一^[1];采用合适的方法计算出 SME

收稿日期:2022-05-06;修回日期:2022-06-27

基金项目:国家重点研发计划资助项目(2018YFB2202604)

作者简介:王晓蕾(1978—),女,安徽涡阳人,合肥工业大学副研究员,硕士生导师。

相对精确的数值解,在运载火箭的轨道控制中扮演着无可替代的角色。因为 SME 具有多变量、高复杂度、大迭代次数和高精度需求的特点,所以方程求解需要进行庞大的计算,影响计算速度。学者们开展了大量的研究工作以提高 SME 的求解速度,文献[2]采用牛顿迭代法减少迭代次数,实现简化弹道快速设计和发射诸元快速装订;文献[3]基于均匀实验设计、牛顿迭代法和逐步回归分析的理论提出一种发射诸元拟合方法,建立发射诸元的快速计算模型;文献[4]在研究标准弹道计算方法和分析偏差弹道快速计算方法的基础上,设计海基潜射弹道导弹诸元快速计算的总体方案,并通过仿真证明方案的正确性和可行性;文献[5]建立用 Kriging 插值实现弹上扰动引力快速计算的数学模型,结合一种飞行弹道仿真结果证明弹载计算机能够满足该方法对速度和精度的要求;文献[6]基于解析预报与数值寻优相结合的方法,利用线性回归方法建立弹道方程多项式并得到对应初值,然后采用 Levenberg-Marquardt 方法搜索诸参数的精确解,在精度要求不高的情况下可以使用该方法快速计算诸元。

上述研究从多个角度对空间运动方程及弹道方程的求解进行加速和优化,但仍然很难满足实际应用中的实时性需求。四阶经典龙格库塔法(classical Runge-Kutta method of order four, RK-4)和四阶 Adams 预测校正法(fourth-order Adams predictor-corrector method, Adams-4) 2 种直接数值求解常微分方程的方法在实际应用中得到广泛应用。借助硬件对上述方法的常微分方程初值问题求解过程进行加速,已经在众多领域得到研究和应用。文献[7]基于经典微分方程数值解法 RK-4,提出一种可重构的微分方程解算器,通过指令实现对不同微分方程运算路径的重构;文献[8]采用复用的 RK-4 求解结构,在现场可编程逻辑门阵列(field programmable gate array, FPGA)平台上完成基于五阶鼠笼式感应电机模型的风力电机动态仿真模型设计;文献[9-11]在 FPGA 上分别实现陈氏混沌系统、Pehlivan-Wei 混沌系统和 6 阶混沌系统的求解,在资源效率和吞吐率上表现良好;文献[12]完成 RK-4 求解弹道方程的 FPGA 设计与实现。

上述研究表明,采用硬件对微分方程求解进行加速具备可行性。针对传统的软件求解无法满足 SME 实时性求解的问题,选择求解速度更快的 Adams-4 方法作为 SME 的主方法,并以 RK-4

作为启动方法,采用硬件对 SME 的数值求解进行加速,在保证有效精度的情况下将运算速度提升一个数量级,对 SME 的求解具有重大工程价值和研究意义。

本文设计的空间运动方程快速求解器(space motion equation fast solver, SMEFS)通过动态配置 SMEFS 内部运算器连接关系,在同一硬件分时实现 RK-4、Adams-4 算法,有效地提高资源利用率,提升系统主频。最后完成空间运动方程快速求解器的 FPGA 设计与实现。

1 空间运动方程数值求解分析

1.1 空间运动方程介绍

根据文献[1]的结论,发射坐标系下 5 自由度空间运动方程为二阶常微分方程,表达式为:

$$m \begin{bmatrix} \frac{dv_x}{dt} \\ \frac{dv_y}{dt} \\ \frac{dv_z}{dt} \end{bmatrix} = \begin{bmatrix} \cos \phi \cos \psi & -\sin \phi & \cos \phi \sin \psi \\ \sin \phi \cos \psi & \cos \phi & \sin \phi \sin \psi \\ -\sin \psi & 0 & \cos \psi \end{bmatrix} \begin{bmatrix} P_e \\ Y_{1c} \\ Z_{1c} \end{bmatrix} + \begin{bmatrix} \cos \theta \cos \sigma & -\sin \theta & \cos \theta \sin \sigma \\ \sin \theta \cos \sigma & \cos \theta & \sin \theta \sin \sigma \\ -\sin \sigma & 0 & \cos \sigma \end{bmatrix} \begin{bmatrix} -C_x q S_M \\ C_y q S_M \alpha \\ -C_y q S_M \beta \end{bmatrix} + m \frac{g_{\omega_e}}{\omega_e} \begin{bmatrix} \omega_{ex} \\ \omega_{ey} \\ \omega_{ez} \end{bmatrix} - m \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x + R_{0x} \\ y + R_{0y} \\ z + R_{0z} \end{bmatrix} + m \frac{g_r}{r} \begin{bmatrix} x + R_{0x} \\ y + R_{0y} \\ z + R_{0z} \end{bmatrix} - m \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \quad (1)$$

其中: $dx/dt = v_x$; $dy/dt = v_y$; $dz/dt = v_z$ 。式(1)等号右侧的表达式均为中间变量,由其他复杂运算计算得到,包括三角/反三角函数运算、矩阵运算、开方运算和四则运算,计算过程涉及发射方位角、火箭静推力等,计算复杂且运算量庞大,具体表达式本文不展开描述。

1.2 高阶常微分方程数值求解方法计算原理

1.2.1 RK-4 方法

RK-4 方法作为经典的单步法,由于计算精度高、可自启动的特点,在工程上广泛应用于求解一阶或高阶常微分方程。式(1)所述二阶常微分方程通过降阶处理得到式(2)所示的抽象方程,对应的 RK-4 计算表达式为:

$$\begin{cases} y'(t) = z, \\ z'(t) = f(t, y, z), \\ y(a) = \alpha, \\ z(b) = \beta \end{cases} \quad (2)$$

$$\begin{cases} y_{n+1} = y_n + hz_n + \frac{h^2}{6}(M_1 + M_2 + M_3), \\ z_{n+1} = z_n + \frac{h}{6}(M_1 + 2M_2 + 2M_3 + M_4), \\ M_1 = f(t_n, y_n, z_n), \\ M_2 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}z_n, z_n + \frac{h}{2}M_1), \\ M_3 = f(t_n + \frac{h}{2}, y_n + \frac{h}{2}z_n + \frac{h^2}{4}M_1, z_n + \frac{h}{2}M_2), \\ M_4 = f(t_n + h, y_n + hz_n + \frac{h^2}{2}M_2, z_n + hM_3) \end{cases} \quad (3)$$

其中: h 为迭代步长; $n = 0, 1, \dots$; 初值为 $y_0 = \alpha, z_0 = \alpha'$ 。

1.2.2 Adams-4 方法

Adams-4 方法是线性多步法, 充分利用每一次的运算结果, 提升计算速度; 通过预测和校正, 减小运算结果的截断误差。式(2)方程组的 Adams-4 的计算表达式^[13]为:

$$\begin{cases} \bar{y}_{n+1} = y_n + \frac{h}{24}(55z_n - 59z_{n-1} + 37z_{n-2} - 9z_{n-3}), \\ \bar{z}_{n+1} = z_n + \frac{h}{24}[55f(t_n, y_n, z_n) - \\ 59f(t_{n-1}, y_{n-1}, z_{n-1}) + 37f(t_{n-2}, y_{n-2}, z_{n-2}) - \\ 9f(t_{n-3}, y_{n-3}, z_{n-3})], \\ y_{n+1} = y_n + \frac{h}{24}(9\bar{z}_{n+1} + 19z_n - 5z_{n-1} + z_{n-2}), \\ z_{n+1} = z_n + \frac{h}{24}[9f(t_{n+1}, \bar{y}_{n+1}, \bar{z}_{n+1}) + \\ 19f(t_n, y_n, z_n) - 5f(t_{n-1}, y_{n-1}, z_{n-1}) + \\ f(t_{n-2}, y_{n-2}, z_{n-2})] \end{cases} \quad (4)$$

2 系统设计

2.1 系统结构

SMEFS 采用数据驱动的管道式计算结构, 系统架构如图 1 所示, 系统采用的数据格式为 IEEE754 标准的双精度浮点数。

系统包括算法控制单元 (algorithmic controller unit, ACU)、数据路径动态配置 (data path dynamic configuration, DPDC) 模块、状态管理器 (state manager, SM)、空间运动方程计算单元 (space motion equation calculate unit, SMECU)、迭代管理器 (iterative manager, IM)、初始化数据存储单元 (initial data memory, IDM)、DDR SDRAM、高效复用的存储器 (efficient reuse of memory, ERM) 以及用于 PCI-E 数据传输的 AXI 数据总线。以式(4)的一次求解任务为例, SMECU 负责计算函数值 f , 而 ACU 完成变量

y, z 的线性组合计算。

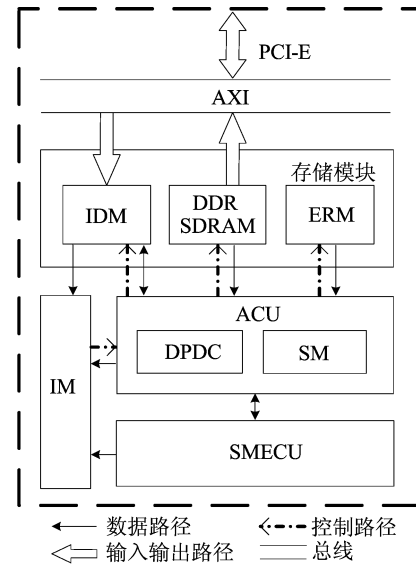


图 1 SEMFS 系统架构

SMEFS 采用折叠变换技术在同一硬件上实现 RK-4、Adams-4 方法电路的分时复用, 即通过对折叠后功能单元间的链接关系进行精准切换和动态配置, 实现数据路径的动态配置, 在有限的资源上完成 SME 数值求解任务。SMEFS 根据主机通过 PCI-E 接口下发的程序和数据独立完成求解工作并将数据回传至主机。

2.2 ACU 和 SMECU 设计方案

ACU 通过折叠对运算器资源实现复用, 根据 RK-4、Adams-4 方法的折叠集, 动态配置数据路径, 完成对应的方法运算任务, 为 SMECU 提供函数计算所需的输入。

1) RK-4 的折叠集分析与重映射。RK-4 方法的硬件实现结构可分为展开式和迭代式^[14]。迭代式结构是通过折叠技术对方法的运算任务进行重映射, 减少功能单元数目的前提下满足性能需求, 该结构为在有限的资源上完成基于该算法的复杂方程求解提供可能。实现迭代式结构到硬件的映射, 需要对 RK-4 方法完整的数据流图 (data flow graph, DFG) 进行重定时, 得到折叠因子和每个复用功能单元对应的折叠集, 并根据折叠因子和折叠集完成对功能单元链接关系进行精准切换和动态配置^[15]。

参考式(3)所述的递推关系发现, 在对 RK-4 方法的硬件结构进行折叠时, 折叠因子 N 可选择 2、4; 考虑到空间运动方程的复杂度, 为了尽可能地节约资源, 本文选择的折叠因子为 4。由于 RK-4 方法作为 Adams-4 方法的启动方法, 只完

成 3 次迭代,则 RK-4 方法的折叠结构运行的任务数为 $l=3$ 。因此,RK-4 方法每次迭代任务在折叠结构上的折叠集为 $\{Nl, Nl+1, Nl+2, Nl+3\}$ 。对 RK-4 方法的运算任务拆分和映射得到其折叠集为 $\{S_{M1}, S_{M2}, S_{M3}, S_{M4}\}$,折叠集中每个运算任务的表达式为:

$$\begin{aligned}
 S_{M1} &= \left\{ M_1, t_n + \frac{h}{2}, \right. \\
 &\quad \left. y_n + \frac{h}{2} z_n, z_n + \frac{h}{2} M_1 \right\}, \\
 S_{M2} &= \left\{ M_2, t_n + \frac{h}{2}, y_n + \right. \\
 &\quad \left. \frac{h}{2} z_n + \frac{h^2}{4} M_1, z_n + \frac{h}{2} M_2 \right\}, \\
 S_{M3} &= \left\{ M_3, t_n + h, y_n + h z_n + \right. \\
 &\quad \left. \frac{h^2}{2} M_2, z_n + h M_3 \right\}, \\
 S_{M4} &= \left\{ M_4, y_n + h z_n + \frac{h^2}{6} (M_1 + M_2 + M_3), \right. \\
 &\quad \left. z_n + \frac{h}{6} (M_1 + 2M_2 + 2M_3 + M_4) \right\} \quad (5)
 \end{aligned}$$

对折叠集进行分析发现,折叠集中 4 个运算任务的运算链长度也不同,任务 S_{M4} 的运算链较长,折叠后所需的功能单元最多;此外任务 S_{M2} 中部分运算与 S_{M1} 中的 M_1 相关。同理,任务 S_{M3} 与 S_{M2} 相关, S_{M4} 与 S_{M1}, S_{M2}, S_{M3} 相关。基于折叠集中任务依赖关系复杂、任务分配不均的问题,对折叠集中的任务进行重定时,优化任务间的依赖关系,均衡分配多个任务的复杂度,对 RK-4 方法运算任务进行重映射,得到的 DFG 如图 2 所示。

2) Adams-4 的折叠集分析与重映射。参考式(4)所示的 Adams-4 方法递推关系,对 Adams-4 方法采用折叠方法分析,其折叠因子只能为 2。针对该折叠因子和方法递推关系,得到其折叠集为 $\{S_{A1}, S_{A2}\}$,折叠集中的运算任务表达式为:

$$\begin{aligned}
 S_{A1} &= \{ \bar{y}_{n+1}, \bar{z}_{n+1}, f(t_{n+1}, \bar{y}_{n+1}, \bar{z}_{n+1}) \}, \\
 S_{A2} &= \{ y_{n+1}, z_{n+1}, f(t_{n+1}, y_{n+1}, z_{n+1}) \} \quad (6)
 \end{aligned}$$

由于 Adams-4 需要 RK-4 提供启动条件,而 RK-4 方法迭代次数 $l=3$,完成 3 次迭代后,其提供的启动条件为 $y_3, z_3, t_3, z, z_1, z_2, f(0), f(1), f(2)$,对比 Adams-4 的启动条件发现,缺少 $f(3)$ 。因此,如果采用折叠因子 $N=2$ 的折叠结构求解空间运动方程,那么其折叠集需要进行优化和重映射,将启动条件 $f(3)$ 的计算也包含在折叠集

中。通过对式(5)的递推关系进行拆分和重定,同时将 SMECU 求解 f 的运算任务在折叠集任务中的相对位置进行调整,求解得到的 DFG 如图 3 所示。

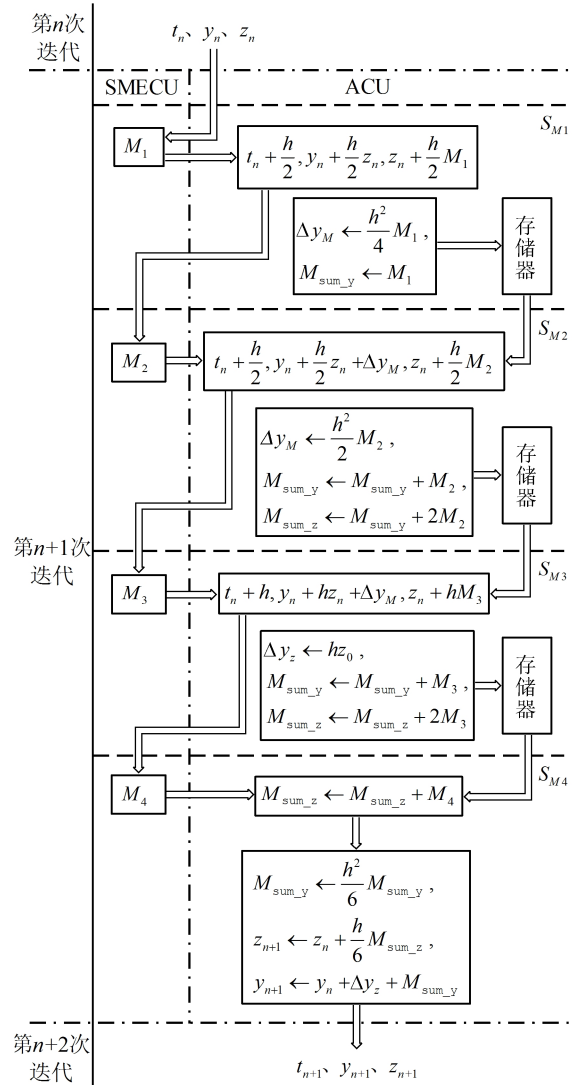


图 2 RK-4 方法折叠结构下优化的 DFG

根据 RK-4、Adam-4 的折叠集,结合 2 种数值求解方法在求解空间运动方程时的工作模式,即可得到 2 种方法在 ACU 中的总折叠集为 $\{S_M, S_A\}$,其中,折叠子集 $S_M = \{S_{M1}, S_{M2}, S_{M3}, S_{M4}\}$; $S_A = \{S_{A1}, S_{A2}\}$ 。2 个折叠子集对应重定时关系见表 1 所列。

表 1 中,折叠子集 S_M 的迭代次数为 3,即迭代 3 次后即可结束当前子集,开始运算符集 S_A ;但 2 个折叠子集对应的折叠因子 N 不同,因此在计算执行时刻时,需要添加约束条 $[4L_1+3, 2L_2]$,即时刻 $4L_1+3$ 和时刻 $2L_2$ 属于

相邻时刻,才能保证重定时后的执行时刻符合折叠子集运算规律。

由图 2、图 3 可知,在 2 种方法折叠结构所需的功能单元中, SMECU 为同一单元,而 2 个

DFG 中 ACU 功能单元所承担的运算任务复杂度不同,对 2 个 DFG 中 ACU 功能单元所需的运算器数量进行统计,得到运算器数量的统计结果,见表 2 所列。

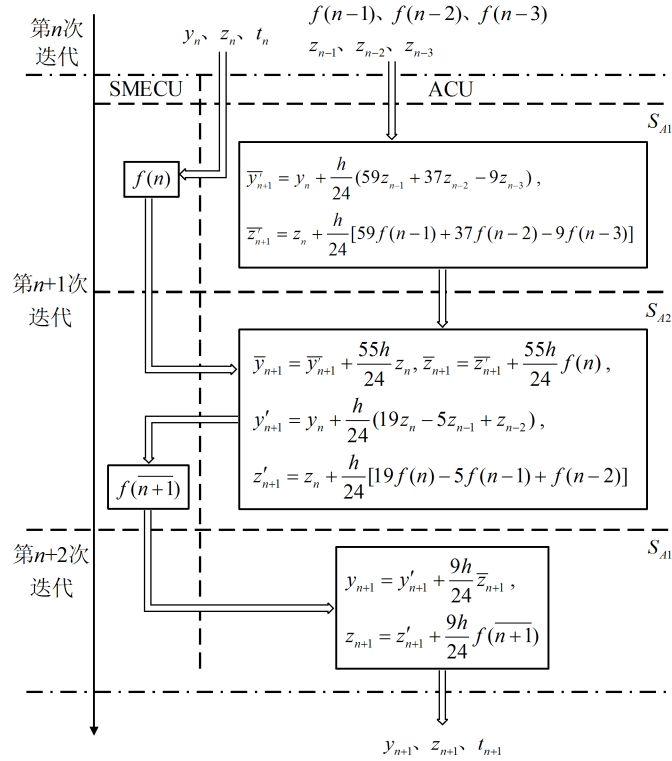


图 3 Adams-4 方法折叠结构下优化的 DFG

表 1 RK-4、Adams-4 方法折叠集相互关系

任务编号	折叠因子 N	迭代次数 l	执行时刻
S_{M1}	$N_1=4$	$1 \leq l_1 \leq 3$	$4l_1$
S_{M2}	$N_1=4$	$1 \leq l_1 \leq 3$	$4l_1+1$
S_{M3}	$N_1=4$	$1 \leq l_1 \leq 3$	$4l_1+2$
S_{M4}	$N_1=4$	$1 \leq l_1 \leq 3$	$4l_1+3$
S_{A1}	$N_2=2$	$l_2 \leq 3$	$2l_2$
S_{A2}	$N_2=2$	$l_2 \leq 3$	$2l_2+1$

表 2 RK-4、Adams-4 方法折叠结果运算器数量统计

求解方法	运算任务	所需数量/个	
		乘法器	加法器
RK-4 方法	S_{M1}	6	7
	S_{M2}	9	10
	S_{M3}	9	10
Adams-4 方法	S_{M4}	15	25
	S_{A1}	24	25
	S_{A2}	24	25
总计		24	25

因为 Adams-4 预测校正方法对 ACU 功能单

元内的运算器需求量更大,所以在考虑运算器复用后,ACU 功能单元由 24 个乘法器和 25 个加法器组成。根据上述 ACU 基本运算器组成方案,结合图 2、图 3 所示的 DFG,即可分别得到 RK-4 和 Adam-4 不同折叠任务所需的运算器链接关系。

为了求得 2 种方法对 ACU 中运算器链接关系的最小并集,将 RK-4 方法每个运算任务下的链接关系依次映射到 Adams-4 方法下的运算器链接关系中,通过对所有情况的并集进行比较,得到最小并集,并根据最小并集完成 ACU 中运算器链接关系的配置,得到数据路径动态配置方案,如图 4 所示。

参考表 1 的任务标号和执行时刻,根据迭代次数得到 ACU 中任务执行顺序,设计 SM 的控制结构,通过实时检测任务执行过程实现任务状态的精准、动态切换。

SMECU 根据 ACU 提供的函数 f 的输入,完成函数值的计算,即 SME 方程值的计算,并将结果返回 ACU。

参考式(1)及文献[1]的相关结论,选择流水线计算模式的加/减法器、乘法器、除法器、三角函数运算器等运算单元,采用管道结构对 SMECU 进行设计;分析 SME 中多个运算链之间的依赖

关系,对部分运算单元的 Latency 进行动态拉伸,避免数据缓存对存储器的访问。SMECU 所设计的管道深度为 260,具备流水工作特征能够准确批量地计算 SME 的方程值。

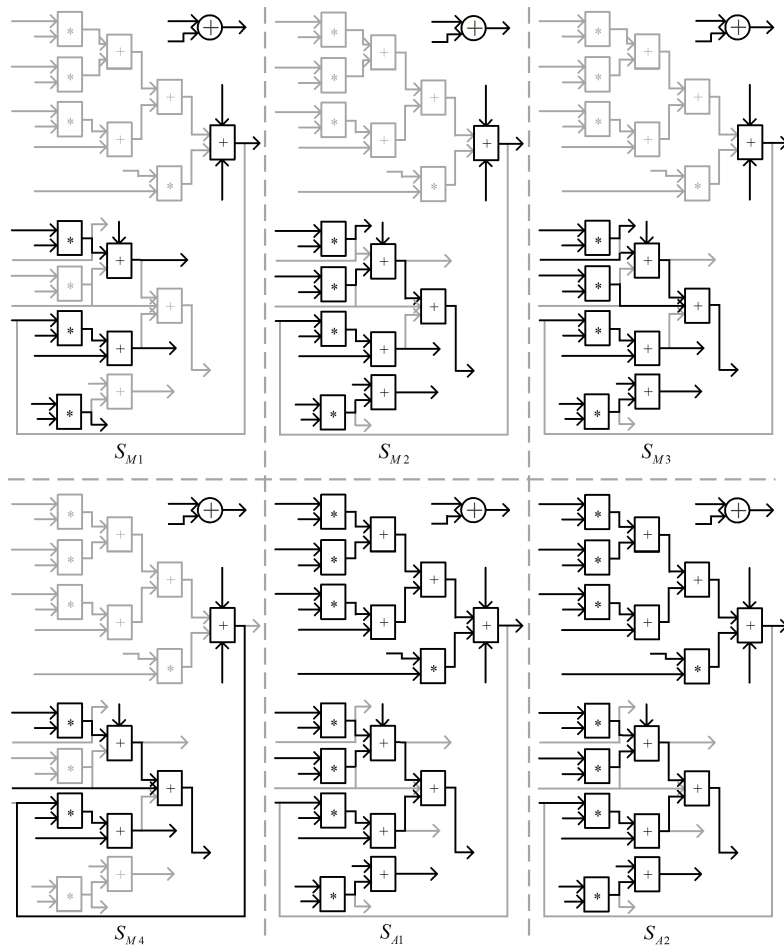


图 4 RK-4、Adams-4 方法融合工作的数据路径动态配置方案

3 实验分析与性能测试

本节针对 SMEFS 的运算模式和功能,选取多种应用场景下的 2 600 组运算任务,本文对 SMEFS 的求解速度及求解结果进行测试验证。结合批量求解任务的求解速度、求解结果相对误差分析以及部分运算任务的运算过程进行对比分析,充分证明该求解器的高效性、功能完备性及可靠性。

为更好地测试系统的峰值性能,实验选择的测试集包含 10 个运算任务,每个任务包括 260 组初值。为了尽可能贴合实际应用测试硬件的性能,所选的 10 个运算任务的迭代次数在 30 000~50 000 之间不等;每个运算任务内的 260 组初值的迭代次数极差在 1 000 以内。

3.1 实验环境

软件实现的载体是飞腾 D2000 CPU @ 2.60 GHz,32 GiB DDR4 2 800 内存,Ubuntu18.04 64 位操作系统的 PC,使用 GCC 7.3.0 编译环境和 C 语言编程工具,采用与快速求解器同样的求解方法和数据格式,对空间运动方程进行求解,硬件平台采用 Xilinx Vivado 2018.3 作为开发工具,以 Xilinx 公司 Virtex-7 XC7V690TFFG-2 型 FPGA 作为 SMEFS 实现的载体,FPGA 芯片最高资源利用率为 60.7%,系统工作主频为 150 MHz。实验使用 MATLAB 完成图像及数据精度对比等操作。

3.2 加速比测试

加速比测试实验主要测试空间运动方程快速求解器完成批量初值的空间运动方程求解任务相

对于软件的加速比。将测试集的初值分别在 PC、FPGA 上完成求解,并分别记录软硬件的计算耗时,依次统计软硬件计算完成 10 个任务的运算耗时和加速比,结果见表 3 所列。

对表 3 中 10 个任务的加速比求平均值,可以得到硬件求解相对于软件计算的加速比为 12.765。

表 3 软硬件求解五自由度 SME 计算耗时对比

实验组别	耗时/ms		加速比
	软件计算	硬件计算	
第 1 组	18.104	1.418	12.767
第 2 组	19.869	1.556	12.769
第 3 组	23.379	1.832	12.761
第 4 组	19.608	1.536	12.766
第 5 组	16.911	1.352	12.763
第 6 组	20.748	1.626	12.760
第 7 组	14.959	1.172	12.764
第 8 组	22.815	1.787	12.767
第 9 组	16.195	1.269	12.762
第 10 组	21.817	1.709	12.766

3.3 精度及误差测试

从 10 个任务中各抽取 2 组初值,绘制软硬件求解得到的弹道并进行对比,初步判断计算结果的精度。对 2 600 组初值最后一次迭代的结果进行相对误差分析,进一步判断误差范围。通过 MATLAB 对 20 组软硬件计算结果绘制图像,为直观对比图像误差,将空间直角坐标系 $\sigma-xyz$ 投影到平面坐标系 xoy 上,得到弹道对比结果,如图 5 所示。从图 5 可以看出,软硬件求解的弹道几乎完全重合。

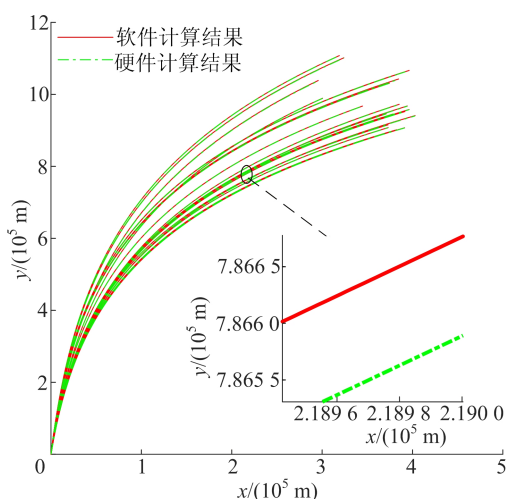
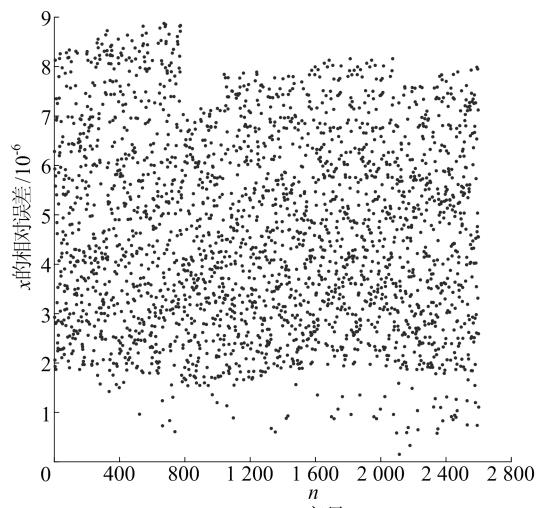


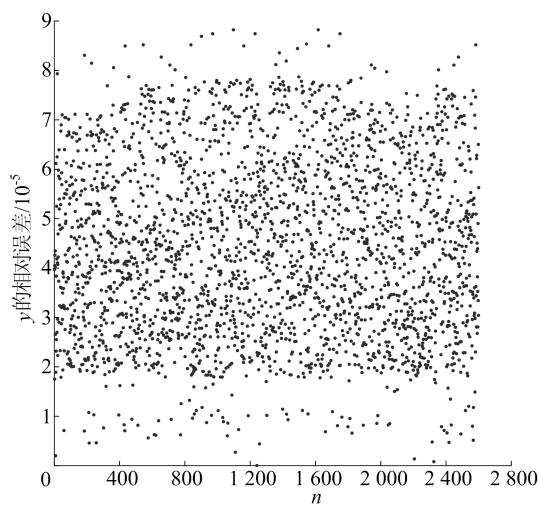
图 5 SME 软硬件求解结果对比

2 600 组初值的计算结果中,包括位移向量 (x, y, z) 和速度向量 (v_x, v_y, v_z) ,参考绘制图像的做法对位移量进行分析,仅对 x, y 进行误差分析;

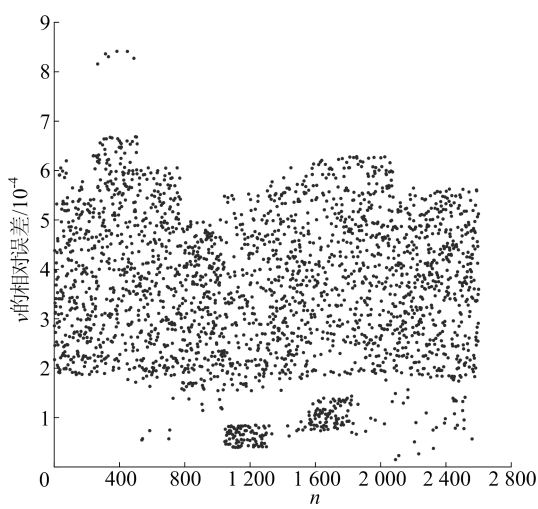
而对速度量的分析是将软硬件求解的速度矢量求模,并对速度的模值进行误差分析。借助 MATLAB 绘制 3 组变量的相对误差散点图,如图 6 所示。



(a) 变量x



(b) 变量y



(c) 变量v

图 6 变量 x, y, v 的软硬件计算结果相对误差散点图

由图 6 可知: x 的相对误差均小于 9×10^{-6} ; y 的相对误差均小于 9×10^{-5} , 误差较小; v 的相对误差均小于 9×10^{-4} , 该变量的相对误差较大, 原因是速度通过向量模值进行误差分析, 存在 3 个变量误差累积的现象, 因此误差高出一个数量级属于理论范围, 实际计算结果 (v_x, v_y, v_z) 的相对误差仍可达到 10^{-5} 数量级。

4 结 论

针对发射坐标系下五自由度空间运动方程的快速求解问题, 选择求解速度更快的 RK-4 方法和 Adams-4 方法相结合的数值求解方法, 采用折叠技术和管道式设计结构, 在给定的资源上设计一种空间运动方程快速求解器, 实现对实际应用中空间运动方程大批量求解任务的有效加速。基于 FPGA 的快速求解器验证实验表明, 在大批量求解任务中, SMEFS 相对于国产 CPU 的平均加速比为 12.765, 且硬件求解与软件求解的相对误差小于 9×10^{-5} , 具备较大的速度优势和较高的计算可靠性。

[参 考 文 献]

- [1] 陈克俊, 刘鲁华, 孟云鹤. 远程火箭飞行动力学与制导[M]. 北京: 国防工业出版社, 2014.
- [2] 王海丽, 陈磊, 胡小平. 弹道导弹基本诸元的快速装订算法研究[J]. 国防科技大学学报, 1999, 21(2): 5-8.
- [3] 多招平. 弹道式导弹射击诸元快速装订方法研究[D]. 北京: 航天科工集团第四总体设计部, 2003.
- [4] 钱山. 弹道导弹变点机动射击诸元快速计算方法研究[D]. 长沙: 国防科技大学, 2006.
- [5] 李夕杰, 舒健生, 陈摩西, 等. 基于 Kriging 插值的弹上扰动引力快速计算[J]. 四川兵工学报, 2010, 31(11): 43-45.
- [6] 韦文书, 荆武兴, 高长生. 机动发射的弹道导弹飞行诸元的快速计算[J]. 哈尔滨工业大学学报, 2012, 44(11): 7-12.
- [7] 张多利, 魏可, 胡永阳, 等. 可重构的微分方程通用解算器研究和实现[J]. 合肥工业大学学报(自然科学版), 2022, 45(3): 336-341, 355.
- [8] CHEN H, SUN S, ALIPRANTIS D C, et al. Dynamic simulation of electric machines on FPGA boards [C]//2009 IEEE International Electric Machines and Drives Conference. [S. l.]: IEEE, 2009: 1523-1528.
- [9] SADOULI S, AZZAZ M S, DJEDDOU M, et al. An FPGA real-time implementation of the Chen's chaotic system for securing chaotic communications[J]. International Journal of Nonlinear Science, 2009, 7(4): 467-474.
- [10] KOYUNCU I, OZCERIT A T, PEHLIVAN I. Implementation of FPGA-based real time novel chaotic oscillator [J]. Nonlinear Dynamics, 2014, 77(1): 49-59.
- [11] TLELO-CAUTLE E, RANGEL-MAGDALENO J J, PANO-AZUCENA A D, et al. FPGA realization of multi-scroll chaotic oscillators[J]. Communications in Nonlinear Science and Numerical Simulation, 2015, 27 (1/2/3): 66-80.
- [12] 潘艇, 杨福彪, 朱勇, 等. 基于龙格库塔的弹道微分方程解算的 FPGA 实现[J]. 计算机测量与控制, 2015, 23(12): 4217-4220.
- [13] 朱晓临. 数值分析[M]. 2 版. 合肥: 中国科学技术大学出版社, 2014: 311-319.
- [14] 魏可. 可重构的微分方程通用解算器研究和实现[D]. 合肥: 合肥工业大学, 2021.
- [15] PARHI K K. VLSI digital signal processing systems: design and implementation [M]. Hoboken: John Wiley & Sons, Inc., 2007: 148-156.
- [5] JUNG H G, KIM D S, YOON P J, et al. Parking slot markings recognition for automatic parking assist system[C]//2006 IEEE Intelligent Vehicles Symposium. [S. l.]: IEEE, 2006: 106-113.
- [6] 朱旺旺, 黄宏成, 马晋兴. 基于图像识别的泊车车位检测算法研究[J]. 汽车工程, 2019, 41(7): 744-749.
- [7] 王鹏飞. 基于全景环视系统的车位检测技术的研究[D]. 天津: 天津大学, 2018.
- [8] ZHANG L, HUANG J, LI X, et al. Vision-based parking-slot detection: a DCNN-based approach and a large-scale benchmark dataset[J]. IEEE Transactions on Image Processing, 2018, 27(11): 5350-5364.
- [9] HUANG J, ZHANG L, SHEN Y, et al. DMPS-PS: a novel approach for parking-slot detection using directional marking-point regression[C]//2019 IEEE International Conference on Multimedia and Expo (ICME). [S. l.]: IEEE, 2019: 212-217.
- [10] 李若迅. 基于环视的自动泊车系统关键技术研究[D]. 合肥: 合肥工业大学, 2021.
- [11] TEICHMANN M, WEBER M, ZOELLNER M, et al. Multinet: real-time joint semantic reasoning for autonomous driving[C]//2018 IEEE intelligent Vehicles Symposium (IV). [S. l.]: IEEE, 2018: 1013-1020.
- [12] WU D, LIAO M W, ZHANG W T, et al. Yolop: you only look once for panoptic driving perception[J]. Machine Intelligence Research, 2022, 19: 550-562.

(责任编辑 张 镗)

(责任编辑 胡亚敏)

(上接第 6 页)