



基于忆阻器的可编程矩阵行列式求解电路设计及应用

肖平旦¹, 洪庆辉^{2*}, 杜四春², 孙辰², 黎俊², 张吉良³

1. 湖南大学物理与微电子科学学院, 长沙 410082

2. 湖南大学信息科学与工程学院, 长沙 410082

3. 湖南大学半导体学院 (集成电路学院), 长沙 410082

* 通信作者. E-mail: hongqinghui@hnu.edu.cn

收稿日期: 2022-06-10; 修回日期: 2022-08-03; 接受日期: 2022-09-02; 网络出版日期: 2023-04-25

国家自然科学基金 (批准号: 62001163, 62122023, U20A20202, 61874042)、湖南省自然科学基金 (批准号: 2020JJ4221, 2021JJ40111, 2022JJ30160)、湖南省创新型省份建设专项资金 (批准号: 2020JK4046, 2022SK2007) 和湖南省重点研发计划 (批准号: 2021RC4019) 资助项目

摘要 矩阵行列式的求解在工程应用中具有重要的意义并得到了广泛的应用, 如何高效快速地求解矩阵行列式成为物联网大数据时代的迫切需求. 为此, 本文提出了一种基于忆阻器的可编程模拟电路, 可一步完成任意阶行列式的计算. 首先, 本工作给出了二阶行列式和三阶行列式的计算电路. 在此基础上, 通过组合电路的方式得到任意阶数行列式求解电路. 同时, 该电路还可以通过调节忆阻器的电导和输入电压源来实现可编程的功能. PSPICE 的仿真结果表明, 该电路的精度在 99.4% 以上, 运行时间比 MATLAB 仿真快 3 个数量级, 即便在 10.0% 的误差干扰的情况下, 其精度仍能超过 93%. 相比于数字处理器, 该电路在面积开销与功耗上具有显著优势. 最后, 该电路可通过各个卫星的余弦参数来计算多卫星构成的空间矩阵行列式以辅助定位卫星组合的选择, 其计算结果的平均精度可达 99.7%.

关键词 忆阻器, 电路设计, 可编程, 行列式, 卫星定位

1 引言

矩阵行列式在实际的工程应用与算法设计中起到了举足轻重的作用^[1~3], 例如, 机器学习^[4]、神经网络^[5]、图像处理^[6,7]和量子计算^[8]等许多科学和工程领域上的难题, 其本质可以转换为行列式的计算问题. 在控制系统中, 判断系统是否稳定是一个非常重要的问题. 不稳定的系统在受到外界或内部的一些因素的干扰时, 该扰动会使被控制量偏离原来的平衡工作状态, 并使其随时间推移而发散, 从而无法正常工作. 因此, 使用相关的判据方法来判断系统的稳定性是非常必要的, 其中赫尔维兹 (Hurwitz) 稳定性判据是一种常用的方法. 设某系统特征方程为

引用格式: 肖平旦, 洪庆辉, 杜四春, 等. 基于忆阻器的可编程矩阵行列式求解电路设计及应用. 中国科学: 信息科学, 2023, 53: 1008–1025, doi: 10.1360/SSI-2022-0229
Xiao P D, Hong Q H, Du S H, et al. Design and application of a programmable matrix determinant-solving circuit based on memristors (in Chinese). Sci Sin Inform, 2023, 53: 1008–1025, doi: 10.1360/SSI-2022-0229

$$a_0 s^n + a_1 s^{n-1} + \cdots + a_{n-1} s^1 + a_n = 0, \quad a_0 > 0, \quad (1)$$

其各个系数可排成如下的 $n \times n$ 阶行列式:

$$\Delta_n = \begin{vmatrix} a_1 & a_0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ a_3 & a_2 & a_1 & a_0 & 0 & \cdots & 0 & 0 \\ a_5 & a_4 & a_3 & a_2 & a_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & a_{n-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & a_n \end{vmatrix}. \quad (2)$$

根据赫尔维兹稳定性判据, 该系统稳定的充分必要条件为其各个系数构成的主行列式及其顺序主子式 δ_i ($i = 1, 2, 3, \dots, n-1$) 行列式值皆大于零, 即

$$\Delta_1 = a_1 > 0, \quad \Delta_2 = \begin{vmatrix} a_1 & a_0 \\ a_3 & a_2 \end{vmatrix} > 0, \quad \Delta_3 = \begin{vmatrix} a_1 & a_0 & 0 \\ a_3 & a_2 & a_1 \\ a_5 & a_4 & a_3 \end{vmatrix}, \quad \dots, \quad \Delta_n > 0. \quad (3)$$

从这个案例中不难看出, 矩阵行列式计算在工程应用中有着举足轻重的作用. 因此, 如何精确地和高效地求解行列式是一个十分重要的课题.

到目前为止, 学界已经提出了许多公式和算法来计算行列式. 传统上, 行列式的运算是通过使用零次法或降价法解决^[9,10]的. 然而, 这些方法仅适用于一些低阶或有明显规律的行列式, 它们难以满足现在对行列式运算越来越高的要求, 同时也难以适用于越来越多样的行列式形态, 并且使用这些方法的计算行列式还需要花费大量的时间. 因此, 相关学者提出了一些新的思路, 试图解决更复杂、更高阶的特定行列式. Ishikawa 等^[11]提出了涉及广义 Vandermonde 行列式的 Cauchy 型行列式的几个恒等式, 并推广和优化了 Cauchy 行列式的运算方法. Elouafi^[12]提出了五对角对称 Toeplitz 矩阵的行列式的一个计算公式, 并且其还考虑了七对角对称矩阵, 对它们的行列式作了较为精确的猜测推理. Jiang 等^[13]在 Toeplitz-Hessenberg 矩阵行列式的基础上, 研究了两类下三角 Toeplitz-like 矩阵, 并分别给出了这些特殊矩阵的行列式和逆矩阵. 通过构造特殊的变换矩阵和利用下三角 Toeplitz-like 矩阵的行列式, 他们得到了 Gaussian Fibonacci skew-Hermitian Toeplitz 矩阵的行列式和相应的逆矩阵. 这些研究者针对一些特定的行列式给出了改进的或者是全新的计算方法, 相比于传统方法拓宽了计算思路, 简化了计算流程. 然而, 这些新的方法仅适用于特定行列式, 具有较明显的局限性.

除了计算方法的拓展与创新以外, 也有相关学者将目光转向了如何改进计算算法以提高行列式运算的速度. Li^[14]提出了一种计算多元矩阵行列式的混合方法. 该工作首先通过降维算法, 将给定的多元矩阵转换为二元矩阵. 然后, 利用并行算法来计算二元矩阵的行列式. 该算法本质上是完全确定的, 而非概率性的. 其不仅可以大大减少符号计算过程中的中间表达式膨胀, 而且可以实现更高的并行度. Takeshi^[15]提出了一种利用精确矩阵分解来计算行列式的算法, 特别是处理病态矩阵情况. 该工作使用了一种精确的逆 LU 分解算法. 即便对于范围更广的问题, 该算法仍可得到任意阶行列式的精确结果. Beliaikov 等^[16]提出了一种在计算机集群上计算行列式的并行算法. 该算法限制节点之间的数据移动, 其不仅可以计算出与特定行或列相对应的所有余子式、子矩阵以及行列式, 而且这些步骤不会

带来任何额外开销. 上述研究相比于前者对于计算方法的改进与创新, 无论是在应用范围还是在计算时间上都所有拓展与提高, 但是这仍有不足. 一方面是由于其完全依赖软件运算和算法复杂度的提高而导致其计算时间的提高有限, 另一方面是在速度提升的同时, 其行列式计算的精度有所下降.

面对上述所提到的矩阵行列式数学方法和计算方法的缺点, 而若从硬件的角度出发可以缓解甚至克服上述缺点. 然而, 当前有关利用模拟电路来求解矩阵行列式的研究较少, 主流研究是利用忆阻阵列来解决矩阵的相关问题. Sun 和 Ielmini^[17] 讨论了基于忆阻器阵列的矩阵模拟计算电路, 讨论了该电路的典型应用、挑战以及未来趋势. 基于上述讨论, 本研究的动机是提出一个基于忆阻器的可编程模拟电路作为行列式计算的实现方案, 提高行列式计算的速度, 缩短运算时间. 本文的主要贡献如下:

(1) 创新性地提出了一种以模拟电路形式计算 3 阶行列式的新结构. 该电路具有较好的泛用性, 并可成为高阶行列式的计算电路的基本模块.

(2) 在提出的 3 阶行列式的基础上, 创新性地提出了可计算任意阶数行列式的可编程模拟电路, 其在运算速度上相比于软件具有显著的优越性.

(3) 基于提出的行列式计算电路, 首次提出了利用行列式计算电路来加速卫星组合选择的方法, 这为空间导航定位中快速确定最佳卫星组合提供了新思路.

本文后续的组织结构如下. 第 2 节简要介绍行列式的定义、几何意义和计算方法. 第 3 节提出了任意阶矩阵行列式计算电路. 第 4 节给出了该电路的仿真结果, 并与软件结果进行了比较. 第 5 节对该电路进行了鲁棒性与计算速度的分析. 第 6 节介绍了该电路在空间定位领域的应用. 最后一节对全文进行了总结.

2 行列式综述

本部分主要介绍行列式的定义、几何意义以及本次工作中使用的计算方法——拉普拉斯 (Laplace) 展开法.

首先, 矩阵行列式定义为一个在 $n \times n$ 矩阵 (n 代表矩阵的阶数) 上计算而得到的标量. 行列式的形式如下^[9]:

$$\det(D) = |D| = \begin{vmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & & d_{2n} \\ \vdots & & \ddots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{nn} \end{vmatrix}. \quad (4)$$

它也可以表示为^[10]

$$\det(D) = \sum_{j_1 j_2 j_3 \cdots j_n} (-1)^{\tau(j_1 j_2 j_3 \cdots j_n)} \prod_{i=1}^n d_{i, j(i)}, \quad (5)$$

其中, $\sum_{j_1 j_2 j_3 \cdots j_n}$ 表示所有 j_n 的排列顺序求和, 然后 $\tau(j_1 j_2 j_3 \cdots j_n)$ 可表示为 $j_1 j_2 j_3 \cdots j_n$ 的逆序数. 从定义中不难发现, 正负号交替排列在相邻项之间, n 阶行列式由 $n!$ 个元素组成. 由于矩阵行列式的结果是一个数值, 而 $n \times n$ 矩阵的本质是 n 列向量的集合. 因此, 可以将行列式视为由对应矩阵中行向量或列向量所组成的平行四边形或多面体的面积或体积^[9, 10]. 例如, 2 阶矩阵行列式的几何意义是平行四边形在 xoy 平面上的面积, 如图 1(a) 所示. 类似地, 3 阶矩阵行列式表示为平行六面体的体积, 如图 1(b) 所示.

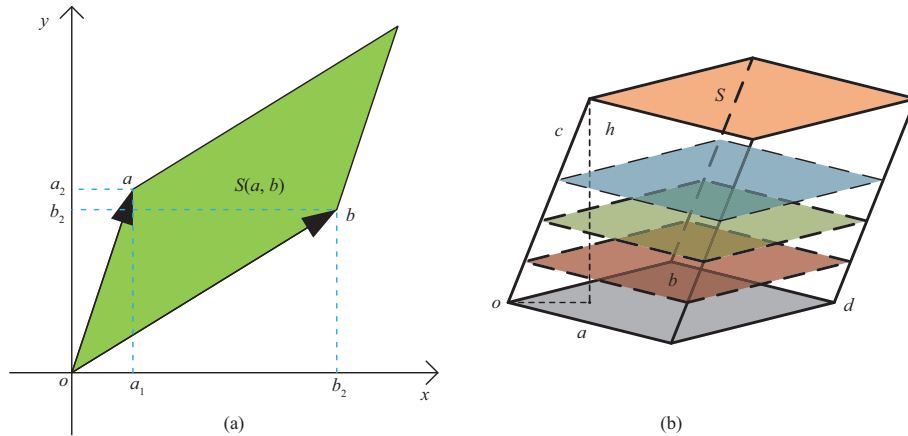


图 1 (网络版彩图) 行列式的几何意义. (a) 2 阶行列式的几何意义; (b) 3 阶行列式的几何意义

Figure 1 (Color online) Geometrical mean of determinant. The geometrical mean of (a) the 2-nd order determinant and (b) the 3-rd order determinant

行列式运算的常用方法有对角线法、等价转化法和拉普拉斯展开法. 然而对角线法仅适用于 2, 3 阶矩阵行列式, 等价转化法依赖于计算机, 只有拉普拉斯展开可适用于任意阶矩阵行列式计算, 并且从模拟电路的角度来看, 拉普拉斯展开更适合在硬件上的模块组合与级联, 因此本工作选择以拉普拉斯展开为基础设计可编程的模拟电路. 该方法需要对一个 $n \times n$ 行列式进行展开. 具体而言, n 阶矩阵的行列式 D 的值为

$$\begin{aligned}
 |D| &= d_{i1}D_{i1} + d_{i2}D_{i2} + \cdots + d_{in}D_{in} \\
 &= d_{1j}D_{1j} + d_{2j}D_{2j} + \cdots + d_{nj}D_{nj},
 \end{aligned}
 \tag{6}$$

其中, 余子式 M_{ij} 是在 n 阶行列式中把 a_{ij} 所在的第 i 行和第 j 列划去后留下的 $(n - 1)$ 阶行列式, 其乘以 $(-1)^{i+j}$ 构成行列式 D 的代数余子式 D_{ij} ($i, j \in 1, 2, 3, \dots, n$).

3 行列式计算电路

本节介绍了忆阻器的模型及仿真和行列式计算电路. 其中, 行列式计算电路包括 2 阶行列式、3 阶行列式以及 n 阶行列式的可编程模拟电路.

3.1 漂移速度自适应忆阻器模型

忆阻器在这项工作中起着重要的作用, 本文设计的电路将采用漂移速度自适应忆阻器模型 (drift speed adaptive memristors model, DSAM) 作为分析模型^[18]. 其中 $i - v$ 的关系是

$$v(t) = i(t) \cdot (R_{\text{off}} - x \cdot \Delta R),
 \tag{7}$$

其中 R_{off} 代表忆阻器中电阻值的最小值, ΔR 表示电阻值最大值和最小值的差值. x 表示范围为 $[0, 1]$ 的归一化宽带电导区域, 对应的状态变量为 $x = 0$ 和 $x = 1$. 因此, x 的导数可以表示为

$$\frac{dx}{dt} = \begin{cases} k_{\text{on}} \cdot \Delta R \cdot i(t) \cdot f(x), & v(t) > v_{\text{on}}, \\ 0, & v_{\text{off}} \leq v(t) \leq v_{\text{on}}, \\ k_{\text{off}} \cdot \Delta R \cdot i(t) \cdot f(x), & v(t) < v_{\text{off}}, \end{cases}
 \tag{8}$$

表 1 DSAM 模型的参数
Table 1 Parameters of DSAM

Parameter	a	p	k_{on}	k_{off}	v_{on} (V)	v_{off} (V)	R_{on} (k Ω)	R_{off} (k Ω)
Value	2.1	1.8	300	10	0.08	-0.08	0.1	10

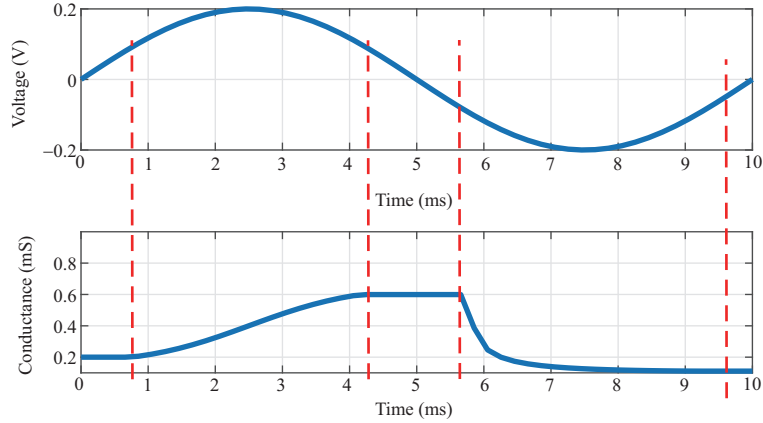


图 2 (网络版彩图) 在不同电压 (V) 下的忆阻器模型电导 (G) 变化模型
Figure 2 (Color online) Change curve of conductance (G) under the different voltages (V)

$$f(x) = \begin{cases} (a \cdot (1-x))^p, & v(t) > 0, \\ (a \cdot x)^p, & v(t) \leq 0. \end{cases} \quad (9)$$

在这些方程中, v_{off} 和 v_{on} 表示忆阻器的阈值电压, 当施加在忆阻器中的电压超过阈值电压时, 忆阻器的电导会发生变化. k_{on} 和 k_{off} 是线性的可调参数, a 和 p 是曲线拟合参数, 可以实现适应不同的 DSAM 模型需求. 表 1 展示了本工作中使用的忆阻器模型参数.

随着电压的变化, 忆阻器的电导如图 2 所示. 最开始往忆阻器上施加一个幅值为 0.2 V, 周期为 10 ms 的正弦电压. 阈值电压与电导变化的关系在图 2 中由虚线画出. 因此, 通过控制施加到忆阻器的电压的方向和持续时间可以在忆阻器中写入适当的电导.

值得注意的是, 忆阻器是一种双端器件. 因此, 将忆阻器与单刀双掷开关相连, 以方便写入电导值来构成可编程忆阻器单元 (以下称为可编程单元). 具体来说, 忆阻器和对应的开关 (PMOS 和 NMOS 构成), 如图 3(a) 所示, 控制电压 V_{ctr} 可以通过选择晶体管的通断进而控制忆阻器的工作模式. 该单元的编程模式如图 3(b) 所示, 当 V_{ctr} 高于 NMOS 晶体管的阈值电压 V_{TN} 时, NMOS 导通, PMOS 截止. 相反, 该单元的计算模式如图 3(c) 所示, 当 V_{ctr} 低于 PMOS 晶体管的阈值电压 V_{TN} 时, NMOS 关断, PMOS 导通.

3.2 二阶行列式的可编程模拟电路设计

首先使用电压信号 V_1, V_2 和电导 G_1, G_2 分别表示 2 阶行列式中的 4 个元素, 其中 V_1, V_2 表示 2 阶行列式的第一列元素, G_1, G_2 表示第二列元素. 2 阶行列式计算电路如图 4 所示, 将 V_1 与 $G_2, -V_1$ 与 G_2, V_2 与 G_1 以及 $-V_2$ 与 G_1 相连构成 4 条支路. 这里改变每个输入信号电源 $V_1, -V_1, V_2, -V_2$ 的电压值, 以及在 V_{ctr} 的控制下, 将可编程单元 G_1, G_2 调整到合适的电导值, 以实现电路的编程以

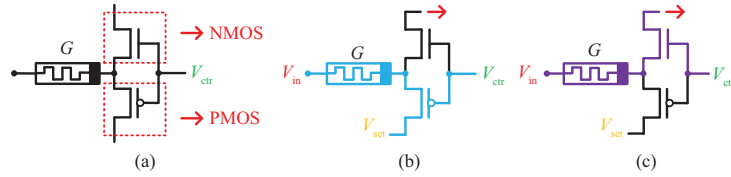


图 3 (网络版彩图) (a) 基于忆阻器的可编程单元; (b) 可编程模式; (c) 计算模式

Figure 3 (Color online) (a) Programmable memristor; (b) programming process of the unit; (c) calculation process of the unit

及对行列式系数的匹配. 2 阶行列式计算电路的原理如下:

$$\begin{cases} I_{\text{out}2} = aV_1G_2 + \bar{a}(-V_1G_2) + \bar{b}V_2G_1 + b(-V_2G_1), \\ V_- = V_+ = 0, \\ V_{\text{out}2} = V_- - I_{\text{out}2}R, \end{cases} \quad (10)$$

其中 a, b 分别代表图 4 里选控模块中输入端口 P_1, P_2 的输入电平信号 (以下简称电平信号), \bar{a}, \bar{b} 代表流经反相器后的电平信号, V_1, V_2, G_1 和 G_2 分别代表前面所提到的 2 阶行列式中被映射和转换的元素. 由于这里的计算中只需要 4 条支路中的两条支路, 因此设计了选控模块来实现支路的选取, 其中每个端口控制两条支路.

选控模块由输入端口、反相器以及电压开关组成. 图 4 中的 T_1, T_2 和 V_s 构成反相器, T_3, T_4 和 T_5, T_6 分别构成各个支路上的电压开关. 在图 4 中, 电压开关可以控制支路的通断, 当输入端口 P_1 输入低电平时, T_3 和 T_4 导通, 而流经反相器后的电平信号使 T_5 和 T_6 断开, 使第 1 条支路上的电压开关打开和第 2 支路上的电压开关关闭; 当 P_1 接入高电平时, T_3, T_4 断开, 而流经反相器后的电平信号使 T_5, T_6 导通, 使第 1 条支路上的电压开关打开和第 2 支路上的电压开关关闭. 如图 4 中给出了具体示例的分析, 假设在 2 阶行列式 $[4, -2; 3, 1]$ 中, 行列式计算电路中的电压源与忆阻器应被进行相应的调节, 其中 V_1 和 V_2 分别与行列式中的元素 4 和 3 对应. 由于 G_1, G_2 对应为忆阻器, 其只能调节为正值, 因此该忆阻器应与元素的绝对值匹配, 所以 G_1 与 G_2 应分别与元素 2 与 1 对应. 综上所述, 在该行列式的计算中,

$$A = 4 \times 1 - (-2 \times 3) = 4 \times 1 + 2 \times 3. \quad (11)$$

其中, 该计算公式由 4×1 与 2×3 组成, 不难看出此式中前项为正后项亦为正, 在这里我们设定前项为 1 后项为 1, 对应到我们设计的电路当中应该选择输入端口 P_1 的第 1 条支路和输入端口 P_2 的第 2 条支路, 输入端口 P_1, P_2 都设置为高电平. 图 4 中输入端口 P_1 的高电平信号在方程组 (7) 中转换为 $a = 1$, 输入端口 P_2 的低电平信号转换为 $b = 0$. 假设此运算放大器为理想运算放大器, 方程组 (10) 中的第 2, 3 个公式为跨阻运算放大器的工作原理, 因此根据基尔霍夫定律 (Kirchhoff's laws) 和方程组 (10), 前面提到的 4 条支路的节点中流出的电流经跨阻运算放大器后将电流信号转换为电压信号, 该电压信号为 2 阶行列式的计算结果.

3.3 三阶行列式的可编程模拟电路设计

基于拉普拉斯展开法, 2 阶行列式是计算 3 阶行列式的基础. 因为根据式 (3), 3 阶行列式的余子式为 2 阶行列式. 因此, 求解 3 阶行列式的工作需要将该行列式中第 1 列元素与对应的代数余子式相乘并相加.

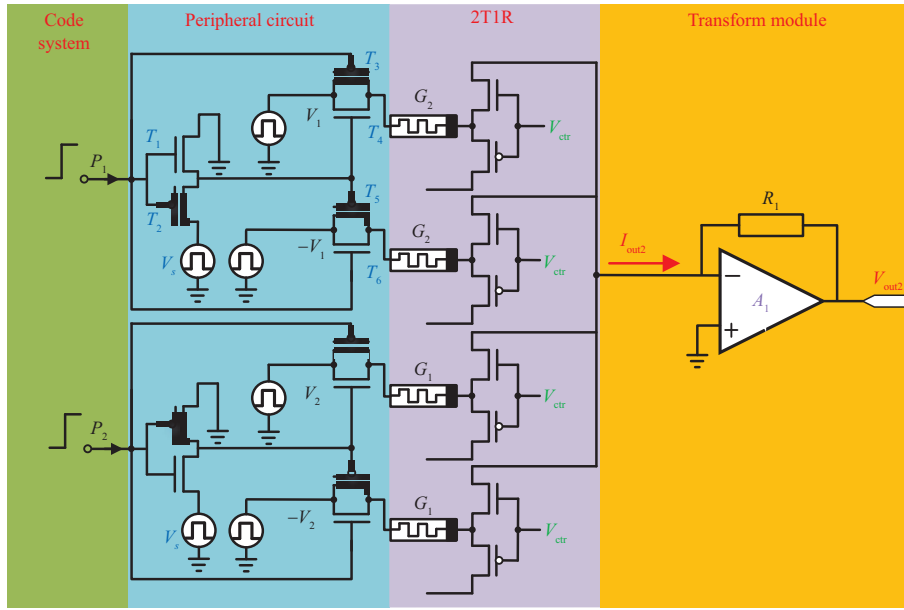


图 4 (网络版彩图) 计算 2 阶行列式的电路

Figure 4 (Color online) The circuit of the 2-nd order determinant

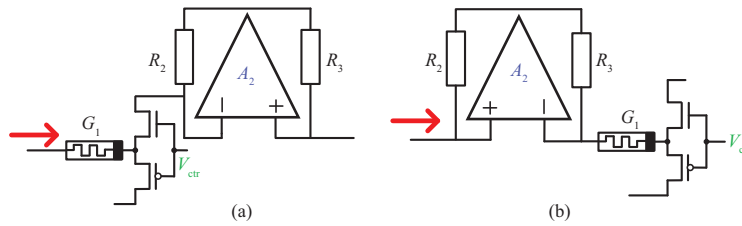


图 5 (网络版彩图) 可编程的电导模型. (a) 可调负电导模型和 (b) 相对的可调正电导模型

Figure 5 (Color online) The module of programmable conductance. (a) The negative programmable conductance and (b) the positive programmable conductance

假设 C_1, C_2, C_3 表示 3 阶行列式余子式的值, 则 c_1, c_2, c_3 表示 3 阶行列式的第 1 列元素 (以下称第 1 列元素). 首先, 调节可编程单元到合适的电导值, 以实现对该列元素的匹配. 然而行列式的元素在可编程单元中只能被编程为正电导值, 因此, 在图 5(a) 中, 本文给出由可编程单元、电阻和运算放大器组成的等效负电导电路模块: 先将第 1 列元素 (绝对值) 编程进该模块中的可编程单元, 然后将余子式 (2 阶行列式) 的电压结果从图中的箭头方向输入, 便可实现等效负电阻并得到乘积项结果. 此外, 该模块可通过改变可编程单元与运算放大器的连接方式和电压输入方向就可以改变等效电导值的正负, 因此正电导电路模块如图 5(b) 所示.

3 阶行列式的计算原理根据式 (6) 可得式 (12):

$$\begin{cases} V_1 = [a_1 V_{22} G_{33} + \bar{a}_1 (-V_{22} G_{33}) + \bar{b}_1 V_{32} G_{23} + b_1 (-V_{32} G_{23})] \cdot (-R), \\ V_2 = [a_2 V_{12} G_{33} + \bar{a}_2 (-V_{12} G_{33}) + \bar{b}_2 V_{32} G_{13} + b_2 (-V_{32} G_{13})] \cdot (-R), \\ V_3 = [a_3 V_{12} G_{23} + \bar{a}_3 (-V_{12} G_{23}) + \bar{b}_3 V_{22} G_{13} + b_3 (-V_{22} G_{13})] \cdot (-R), \\ I_{out3} = V_1 G_{11} - V_2 G_{21} + V_3 G_{31}, \end{cases} \quad (12)$$

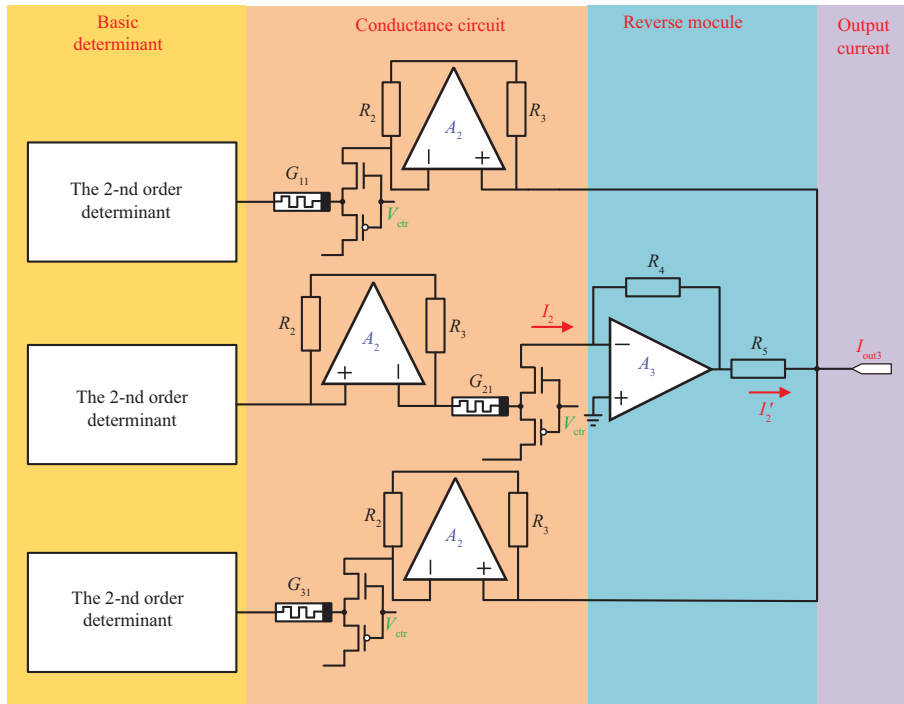


图 6 (网络版彩图) 计算 3 阶行列式的电路

Figure 6 (Color online) The circuit of the 3-rd order determinant

其中 R 表示跨阻放大器中的电阻, 可将电流信号转换为电压信号. V_1, V_2, V_3 表示 3 阶行列式的余子式结果. $a_1, a_2, a_3, b_1, b_2, b_3$ 表示选控模块的电平信号. 此外, 从方程组 (12) 中的第 4 个方程可以发现 3 阶行列式计算中每项的正负号上存在差异, 这是因为式 (6) 中每个代数余子式存在正负差异. 因此, 这就需要在每个第偶数项余子式前加上负号, 映射到电路上就需要在每个偶数编号支路加上电流反向器. 电流反向器工作原理如下:

$$\begin{cases} V_{OUT} = V_- - I_2 R_4, \\ V_- = V_+ = 0, \\ I'_2 = \frac{V_{OUT} - 0}{R_5}, \end{cases} \quad (13)$$

其中 I_2 为 3 阶行列式计算电路中第 2 个余子式的计算结果, R_4 和 R_5 为电流反向器上的电阻, V_-, V_+, V_{OUT} 分别为运算放大器的正向输入端、反向输入端以及输出端, I'_2 为 I_2 流经电流反向器后得到的反向电流.

3 阶行列式的计算电路如图 6 所示, 其原理如方程组 (12) 所示, 将 2 阶行列式的电压结果施加在由第 1 列元素映射的控制回路上以得到相应的电流, 由基尔霍夫定律可知, 从各个支路组合在一起的节点流出的电流值, 即为 3 阶行列式的计算结果.

3.4 n 阶行列式的可编程模拟电路设计 ($n \geq 4$)

如前面所述, 2 阶行列式的结果是计算 3 阶行列式的基础. 根据式 (6), n 阶行列式的计算以 $n - 1$

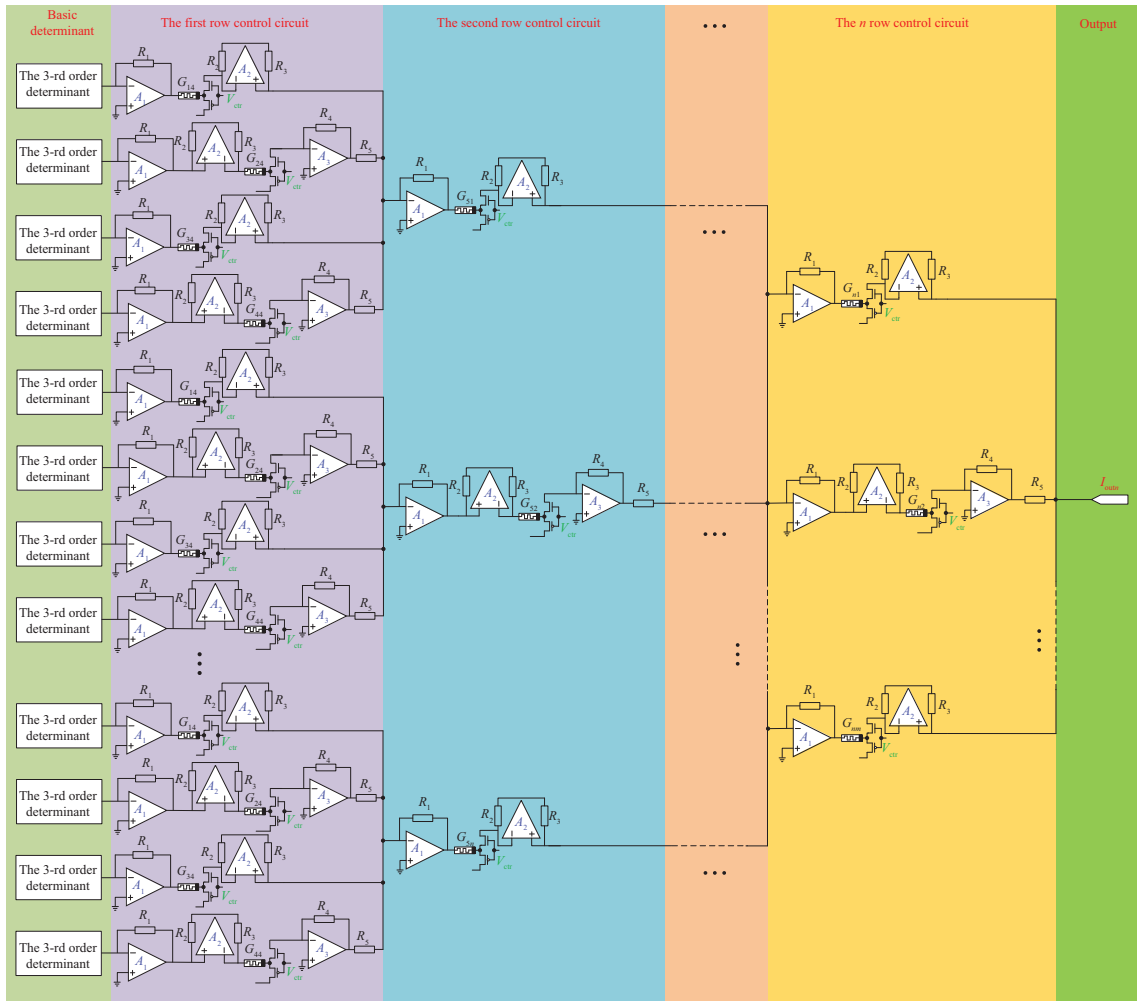


图 7 (网络版彩图) 计算 n 阶行列式的电路 ($n \geq 4$)

Figure 7 (Color online) The circuit of n -th order determinant ($n \geq 4$)

阶行列式的计算结果为基础. 因此, n 阶行列式的计算公式如下:

$$I_{outn} = V_1 G_{11} + V_2 G_{21} + V_3 G_{31} + \dots + V_{n-1} G_{(n-1)1} + V_n G_{n1}, \quad (14)$$

其中, $G_{11}, G_{21}, \dots, G_{n1}$ 为在 n 阶行列式中的第 1 列元素, 而 $V_1, V_2, V_3, \dots, V_n$ 表示 n 阶行列式除去第 1 列元素后剩下的余子式, 也就是相应的 $(n-1)$ 阶行列式值. 根据式 (6), 计算 $(n-1)$ 阶行列式需要 $(n-1)$ 个 $(n-2)$ 阶行列式的数值和 $(n-1)$ 个控制电路, 计算 $(n-2)$ 阶行列式需要 $(n-2)$ 个 $(n-3)$ 阶行列式的数值和 $(n-2)$ 个控制电路等. 因此, 可以认为 n 阶行列式的可编程模拟电路构建的基础是 3 阶行列式的可编程模拟电路, n 阶行列式的可编程模拟电路结构如图 7 所示.

在构建高阶的矩阵行列式计算电路时, 将上一阶矩阵行列式计算电路作为子模块, 再通过相应的控制回路将各个子模块连接在一起, 其节点流出的电流为电路的计算结果. 这种方式可以将低阶矩阵行列式计算电路拓展为高阶矩阵行列式计算电路, 展现了较好的可扩展性. 前文提到, 3 阶矩阵行列式计算电路为基础模块 (以下简称基础模块), 通过计算可知, n 阶矩阵行列式电路与 3 阶矩阵行列式电

路关系如下:

$$a_n = \frac{1}{6}n!, \quad n \geq 3, \quad (15)$$

其中 a_n 表示 n 阶矩阵行列式电路所需基础模块的个数, n 为矩阵行列式的阶数. 此外, 搭建矩阵行列式计算电路需要的控制回路数量与基础模块数量相同.

4 行列式计算电路的 PSPICE 仿真

本节的主要工作是验证所设计的可编程模拟电路的准确性. 这里以 MATLAB 内置函数计算的结果为基准并与 PSPICE 的仿真结果进行比较, 验证了该电路的优良性能. 以 3 阶矩阵行列式为例, 其中

$$A = \begin{bmatrix} 2 & 9 & 1 \\ 5 & 1 & 2 \\ 4 & 8 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} -2 & -5 & -3 \\ -1 & -8 & -2 \\ -4 & -2 & -3 \end{bmatrix}, \quad C = \begin{bmatrix} -2 & 1 & -1 \\ 3 & -4 & 2 \\ 1 & 2 & 1 \end{bmatrix}. \quad (16)$$

这些矩阵中的列向量 a_1 与 a_3 , b_1 与 b_3 , c_1 与 c_3 以电导的形式输入, a_2 , b_2 , c_2 以电压的形式输入. 为了简化计算, 需要将电流、电压和电导进行单位变换 $I_0 = 1 \text{ mA}$, $V_0 = 100 \text{ mV}$, $S_0 = 100 \text{ } \mu\text{S}$, 其中

$$\begin{aligned} a_1 &= \begin{bmatrix} 2 \\ 5 \\ 4 \end{bmatrix} G_0, & a_2 &= \begin{bmatrix} 9 \\ 1 \\ 8 \end{bmatrix} V_0, & a_3 &= \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix} G_0, \\ b_1 &= \begin{bmatrix} -2 \\ -1 \\ -4 \end{bmatrix} G_0, & b_2 &= \begin{bmatrix} -5 \\ -8 \\ -2 \end{bmatrix} V_0, & b_3 &= \begin{bmatrix} -3 \\ -2 \\ -3 \end{bmatrix} G_0, \\ c_1 &= \begin{bmatrix} -2 \\ 3 \\ 1 \end{bmatrix} G_0, & c_2 &= \begin{bmatrix} 1 \\ -4 \\ 2 \end{bmatrix} V_0, & c_3 &= \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix} G_0. \end{aligned} \quad (17)$$

由图 8 的结果可以发现, 该电路的输出结果与理论解 (包括 3 阶行列式值以及其每个余子式) 非常接近. 由图 8 可知, 其余子式, 也就是 2 阶矩阵行列式值的准确率均超过 99.6%, 3 阶行列式结果的准确率分别为 99.428%, 99.534%, 99.628%.

对于高阶矩阵行列式而言, 这里对 3, 4, 5, 6 行列式计算电路进行了多次测试并进行了统计. 如表 2 所示, 各个电路的平均准确率分别为 99.878%, 99.838%, 99.797%, 99.489%. 从表 2 中可以看出, 平均准确率随着阶数的提高呈下降趋势, 但总准确率均大于 99%. 因此, 可以发现该电路在准确率方面具有优异的性能.

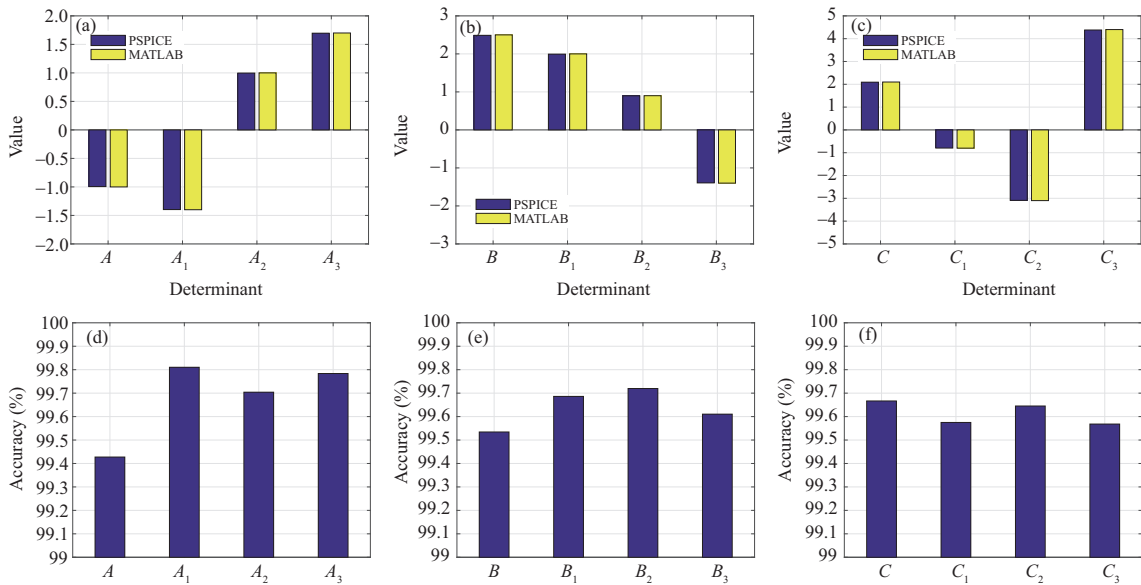


图 8 (网络版彩图) (a)~(c) 为二阶和三阶行列式结果的比较; (d)~(f) 为二阶和三阶行列式的精确度的比较
Figure 8 (Color online) (a)~(c) Solution comparison of the 2-nd and the 3-rd order determinant; (d)~(f) accuracy analysis of the 2-nd and the 3-rd order determinant

表 2 计算不同阶行列式的平均精度

Table 2 The average accuracy of calculating determinant with different orders

The order of determinant	Average accuracy (%)
3-rd	99.878
4-th	99.838
5-th	99.797
6-th	99.489

5 电路性能分析

第 4 节论证了所设计的可编程模拟电路在准确率方面的性能,但在实际应用中,电路的工作速度、鲁棒性、面积开销、功耗以及器件良率对电路计算精度的影响也是至关重要的. 本节对电路的计算速度和电路的鲁棒性进行了 MATLAB 仿真和 PSPICE 仿真分析,运行 MATLAB 和 PSPICE 的处理器规格为: RAM 大小为 15.9 GB, CPU 数量为 8 个, CPU 速度为 2.80 GHz. 此外,本节也对电路的面积开销、功耗以及器件良率对电路的计算精度影响做了相应的计算与仿真分析.

5.1 电路计算的速度分析

在大量实验的基础上,表 3 给出了 MATLAB 与该电路在 3, 4, 5, 6 阶矩阵行列式计算中的平均计算时间. 与 MATLAB 仿真相比,该电路在运行时间上具有显著的优越性,造成这种现象的主要原因是传统的冯·诺依曼 (von Neumann) 结构体系中的存储和计算单元分离,从而导致行列式计算在数据和命令的传输上花费了更多的时间. 因此,在处理大量数据时,串行计算方法所需的时间将显著增加. 然而,该电路使用由忆阻器构成的可编程单元使存储和计算单元一体化,大大减少了数据和命令传输上

表 3 MATLAB 与 PSPICE 仿真中不同阶行列式的计算时间比较

Table 3 Time comparison of different order determinants between MATLAB and PSPICE

The order of determinant	Time (PSPICE) (μs)	Time (MATLAB) (ms)
3-rd	0.19	0.136
4-th	0.23	0.208
5-th	0.30	0.314
6-th	0.36	0.472

表 4 不同阶行列式计算电路的面积开销

Table 4 Area overhead of different order determinants calculation circuits

The order of determinants	3-rd	4-th	5-th	6-th
Area overhead (mm^2)	0.00201	0.0867	0.4415	2.6587

所花费的时间. 此外, 从表 3 中可以发现, 该电路的计算时间随行列式阶数的提升而升高, 这是因为本文使用了运算放大器等电路模块来辅助运算, 而运算放大器内部还有晶体管、电容等电子元器件, 这些可能会影响电路的计算时间, 尤其表现在运算放大器的上升时间这一参数上, 其计算公式如下:

$$t_R = \frac{\ln 0.9 - \ln 0.1}{2\pi f_t}, \quad (18)$$

其中 f_t 为运算放大器的增益带宽积, 这里使用的运算放大器的增益带宽积为 10 MHz. 通过计算, 该运算放大器的上升时间大约为 35 ns. 因此, 随着行列式计算电路的阶数提升, 运算放大器的数量增多, 电路的计算时间升高, 但与计算机相比整体还是具有显著的优越性的. 此外, 该电路不仅实现了并联计算, 而且还使用模拟信号, 通过欧姆定律与基尔霍夫定律实现了一步计算. 随着处理的数据越来越多, 所设计的电路在实际应用中将展现出更优异的运算能力.

5.2 电路计算的鲁棒性分析

在实际情况下, 忆阻器很难精确调整电导值, 这可能会造成一些误差^[18], 因此, 本小节讨论了电路的鲁棒性. 首先, 在行列式计算电路的所有的可编程单元和输入信号电压源中加入 2.5%, 5%, 7.5%, 10% 的随机误差. 其次, 使用蒙特卡罗 (Monte Carlo) 方法对结果进行分析. 该电路经过多次的计算与测试, 在 4 种误差条件下的 3, 4, 5, 6 阶行列式计算电路的平均准确率如图 9 所示.

在图 9 中, 行列式电路的输出结果可以稳定在目标值附近. 从 3 阶到 6 阶矩阵行列式, 该电路输出结果的平均精度可以稳定在 93% 以上.

5.3 电路的面积分析

在电路的设计中, 面积是一个非常重要的指标, 这关系到电路性能、生产成本等问题. 因此, 本小节估计并讨论了电路的面积开销. 在 22 nm 的工艺制程下, 不同阶数的行列式计算电路面积开销如表 4 所示.

该电路以行列式的拉普拉斯展开为基础, 其阶数的扩展与各个模块之间有着明显的规律关系, 相关公式如下:

$$A_n = n \cdot A_{n-1} + \frac{(1685.5n - 413.2)}{1000000} \text{ mm}^2, \quad (19)$$

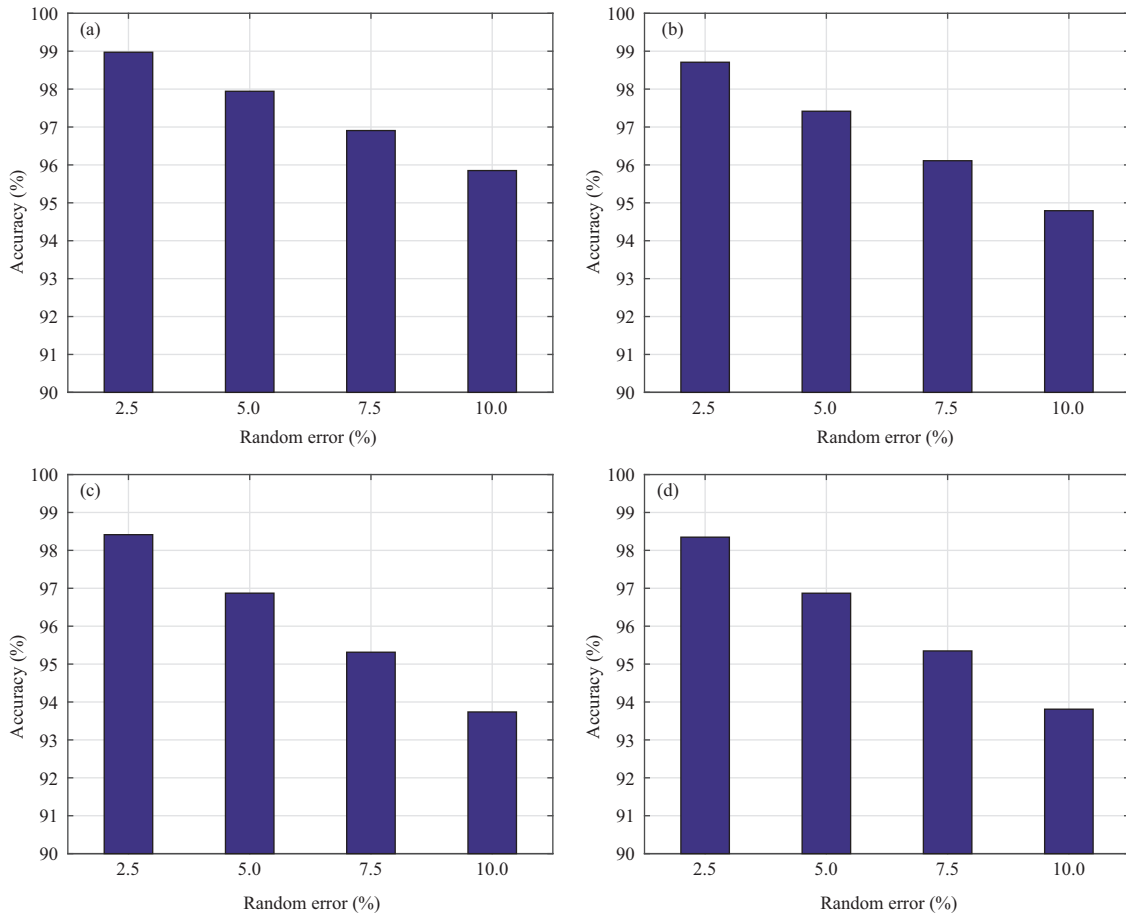


图 9 (网络版彩图) 不同随机误差条件下输出结果的平均精度. (a) 3 阶行列式; (b) 4 阶行列式; (c) 5 阶行列式; (d) 6 阶行列式

Figure 9 (Color online) Average accuracy of output results under different random error conditions. (a) 3-rd order determinant; (b) 4-th order determinant; (c) 5-th order determinant; (d) 6-th order determinant

其中 n 代表矩阵行列式的阶数, A_n, A_{n-1} 表示 n 阶, $n-1$ 阶矩阵行列式的面积. 该式表明下一阶矩阵行列式的计算电路面积与上一阶矩阵行列式的计算电路面积存在明显的递推关系.

除此之外, 通过使用数字处理器处理大规模矩阵行列式也是一种方法. 但是, 相比于本文使用模拟电路计算矩阵行列式, 数字处理器不仅需要如模数转换器等外围电路的参与来处理与转换信号, 还需要 CMOS 门电路来对数字信号进行计算与存储. 随着矩阵行列式的阶数越来越高, 计算电路的规模要求也越来越高, 因此需要更多的外围电路来处理信号以及更多的 CMOS 门电路来计算与存储数字信号, 这些因素无疑会大大增加面积开销. 而本文使用模拟电路来计算矩阵行列式, 利用模拟信号来代替数字信号以实现对信号的运算, 避免了使用大量外围电路来转换信号; 其次, 忆阻器可以代替 CMOS 单元来存储与计算行列式元素, 减少了对晶体管的使用. 综上所述, 相比于数字处理器, 本文提出的利用模拟电路来求解矩阵行列式在面积开销上具有显著优势.

5.4 电路的功耗分析

在多次的实验与测量后, 表 5 给出了不同阶数矩阵行列式计算电路的功耗.

表 5 不同阶行列式计算电路的功耗

Table 5 Power consumption of different order determinants calculation circuits

The order of determinants	3-rd	4-th	5-th	6-th
Power consumption (mW)	2.03	8.23	41.26	247.07

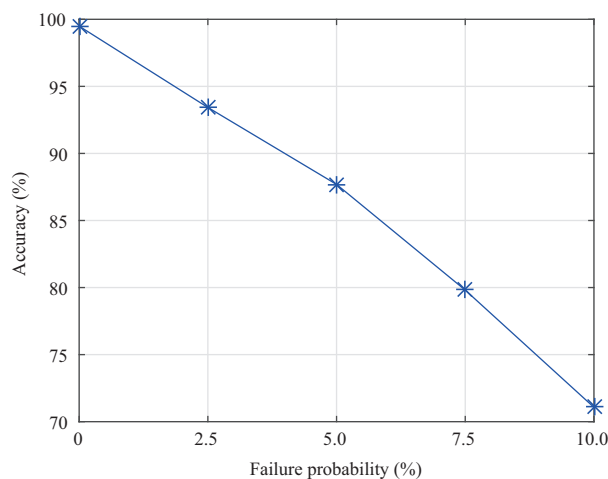


图 10 (网络版彩图) 在忆阻器编程失败的情况下电路的计算精度

Figure 10 (Color online) Accuracy of circuits in case of memristors programming failure

经过分析,发现了电路功耗与行列式阶数之间存在一定的规律关系,相关公式如下所示:

$$P_{n+1} = (n + 1) \cdot P_n - 0.11 \text{ mW}, \quad (20)$$

其中 P_{n+1} , P_n 分别代表 $n + 1$ 和 n 阶矩阵行列式电路的功耗, n 代表行列式电路的阶数. 该式表明下一阶矩阵行列式电路的功耗与上一阶矩阵行列式的电路功耗存在明显的递推关系.

此外,使用数字处理器是一种计算矩阵行列式的有效途径. 然而,数字处理器是通过使用数字信号完成计算任务的,因此需要大量的外围电路和 CMOS 门来对信号进行转换、存储与计算,这无疑增加了电路的功耗. 此外,现有的处理器大部分使用了冯·诺依曼架构,这种结构最大的缺点是存储与计算单元分离,使得数据需要通过总线从存储单元移动到计算单元来完成计算任务. 这不仅增加了处理器完成计算任务的时间,也大大增加了处理器的功耗. 而本次工作提出的计算电路使用模拟信号来实现一步求解矩阵行列式,避免了使用大量外围电路对信号的转换,减小了功耗开销. 其次,本电路使用忆阻器以实现对参数的存储与计算的一体化,避免了大量 CMOS 门电路的使用,也解决了冯·诺依曼架构由于存算分离而导致的功耗增加的问题. 相比于使用数字处理器计算矩阵行列式,该电路计算矩阵行列式的功耗大大降低.

5.5 器件良率对电路计算精度的分析

在实际使用的情况中,电路对忆阻器的编程很难实现百分之百的成功率,并且忆阻器编程的失败会对电路的计算精度带来很大的影响,因此,本小节对电路做了器件有限良率影响的分析并测量计算电路的精度. 针对电路计算高阶矩阵行列式的情况,这里以 6 阶矩阵行列式为例,分别在 2.5%, 5.0%, 7.5%, 10% 的忆阻器编程失败率的情况下,测量的电路计算精度结果如图 10 所示.

从图中可知, 随着忆阻器编程失败率的升高, 矩阵行列式计算电路的计算精度在不断下降. 当忆阻器编程失败比率达到 10% 时, 电路精度最低达到 71.07%. 经过分析后, 电路计算精度下降这么快的原因是: 高阶行列式计算电路是以上一阶行列式电路和相应的控制回路为基础而搭建的, 每一阶的行列式电路的计算误差会在各级电路中不断累加, 最终会较大影响最终的计算精度.

6 行列式计算电路在卫星组合选择中的应用

如前所述, 3 阶及以上的行列式在几何空间中可表示为对应的体积, 并且行列式计算在许多领域得到了广泛的应用. 因此, 本节主要介绍行列式在空间卫星组合选择中的应用. 通过融合多个卫星导航系统的观测数据可以实现组合定位, 然而这需要增加可见卫星的数量^[19]. 接收终端设备的用户可以在众多的卫星组合中选择最好的卫星组合来有效减少系统位置误差, 进而提高系统定位的精度和稳定性^[20]. 几何稀释精度因子 (geometric dilution of precision, GDOP) 是一个用来衡量导航精度的数值, 它表示空间卫星星座中位置误差与几何位置变化之间的关系. 因此, 它是判断所选卫星组合的关键标准^[21, 22]. “伪距”是指全球定位系统 (GPS) 观测站与卫星之间的距离, 其由 GPS 观测获得. 因为测量的距离包括时钟误差^[23] 所以叫作“伪距”. 该公式为

$$p_i = \|r - r_i\| + c\Delta t_A, \quad (21)$$

其中 $r = (x, y, z)$ 和 $r_i = (x_i, y_i, z_i)$ 表示接收器和第 i 颗卫星在三维空间中的位置, $c\Delta t_A$ 表示相对于星座的接收机时钟偏差, 单位为 m^[24]. 通常接收机的三维坐标和接收机时钟偏差构成 4 个未知参数. 因此, 接收器至少需要 4 颗卫星才能从这些参数中获得伪距离. 对式 (21) 进行泰勒 (Taylor) 展开, 可以得到

$$\Delta p_i = p_i - \tilde{p}_i = h_i \Delta r^T + c\Delta t_A, \quad (22)$$

其中 $p_i = \|r_i - \tilde{r}_i\|$, 式中的其他变量可表示为

$$\begin{cases} \Delta r = [x - \tilde{x}, y - \tilde{y}, z - \tilde{z}], \\ h = \left[\frac{\tilde{x} - x_i}{\|\tilde{r} - r_i\|}, \frac{\tilde{y} - y_i}{\|\tilde{r} - r_i\|}, \frac{\tilde{z} - z_i}{\|\tilde{r} - r_i\|} \right] = [l_i, m_i, n_i]. \end{cases} \quad (23)$$

其通常可表现为

$$P = H\Delta x, \quad (24)$$

其中 $c\Delta t_A$ (残差) 可以被认为观测值减去其中包含的有用信息后的剩余量. P 表示测量向量, 并且 Δx 表示提出的未知参数^[24]. h_i 表示卫星与接收机之间的有向余弦, H 表示方向上的余弦矩阵, 这是计算 GDOP 的关键. 因此, GDOP 的公式为^[25]

$$\text{GDOP} = \sqrt{\text{trace}[(H^T H)^{-1}]} = \frac{\sqrt{\text{trace}[(H^T H)^*]}}{|H|}. \quad (25)$$

空间定位必须要在所有的卫星组合中找到一个最好的组合, 这就需要在可选范围内选择具有最小的 GDOP 的卫星组合. 根据式 (25), 最大的行列式 H 的卫星组合意味着其对应着拥有最小 GDOP, 因此, H 的行列式值可以作为选择和判断卫星组合的标准.

表 6 所给参考文献中的卫星余弦向量参数

Table 6 The Cosine of some given satellites

The number of satellite	Cosine(x)	Cosine(y)	Cosine(z)
h_A	-0.6257	-0.2565	-0.7366
h_B	-0.6928	0.5700	0.4417
h_C	-0.3920	0.4136	-0.8218
h_D	0.8162	0.2524	-0.5197
h_E	0.1250	0.6302	0.7633

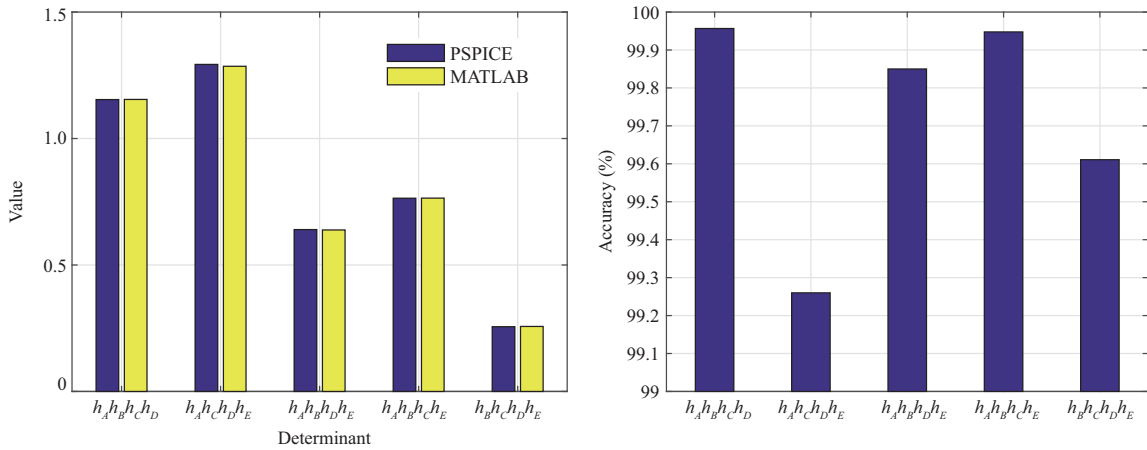


图 11 (网络版彩图) (a) 行列式值与 (b) 不同星座下电路准确率的结果比较

Figure 11 (Color online) Comparison of (a) the determinant and (b) the accuracy rate of the circuit in different constellations

表 7 MATLAB 和 PSPICE 计算不同阶行列式的平均运行时间比较

Table 7 The comparison of average running time to calculate determinant with different order between MATLAB and PSPICE

The combination of satellite	Time (PSPICE) (μ s)	Time (MATLAB) (ms)
$h_A h_B h_C h_D$	0.24	0.236
$h_A h_C h_D h_E$	0.26	0.208
$h_A h_B h_D h_E$	0.21	0.194
$h_A h_B h_C h_E$	0.27	0.271
$h_B h_C h_D h_E$	0.23	0.259

根据上述分析, 只要能够计算出所有卫星组合的行列式并进行大小比较, 就可以得到最优的卫星组合. 本文所设计的电路可以解决计算行列式 H 的问题. 为了验证该方案并证明其优越性, 表 6 给出了实例和数据. Teng 等^[24] 给出了 5 个卫星的余弦向量.

首先, 从这 5 组数据中选取任意 4 组数据进行排列组合, 以此来找出所有的卫星组合; 其次, 设计相应阶数的行列式计算电路用于计算这些组合所对应的矩阵行列式; 同时, 该电路的电学单位的转换与之前设置相同. 图 11 展示了这些行列式的计算结果, 表 7 展示了计算这些行列式在 PSPICE 和 MATLAB 中的运行时间.

由图 11 可知, 所设计的电路拥有良好的准确率, 其均大于 99%。此外, 该电路的运算时间比 MATLAB 减少了 3 个数量级。此外, 图 11 中第 2 个卫星组合在所有卫星组合中行列式值最大, 因此, 第 2 组的定位效果最佳。该方法通过所设计的电路可以求解并判断出最佳的卫星组合, 相比于软件, 该方案计算更快, 更方便。随着阶数的增加, 所设计电路的优势变得更加明显。进一步来说, 行列式计算电路几乎可以加速任何有快速计算行列式需求的算法及工程应用, 具有广阔的应用前景。

7 总结

本文首次提出了一种可以快速计算矩阵行列式的可编程模拟电路。相比于程序复杂的计算步骤, 该电路能够一步求解出任意阶数的矩阵行列式。同时, 与软件毫秒级的平均计算时间相比, 该电路能在微秒级完成矩阵行列式的计算, 使行列式的计算速度大幅提高。此外, PSPICE 的多次数据仿真结果表明, 电路的准确率均达 99.4% 以上。同时, 该电路具有较好的鲁棒性, 即便在 10.0% 的误差干扰下精度仍能保持在 93% 以上。其次, 相比于数字处理器, 该电路在面积开销与功耗上具有显著优势。此外, 本文将该电路应用于空间定位领域中的卫星组合选择计算中, 准确性平均精度可达 99.7% 以上。

参考文献

- 1 Hong Q H, Shi Z R, Sun J R, et al. Memristive self-learning logic circuit with application to encoder and decoder. *Neural Comput Applic*, 2021, 33: 4901–4913
- 2 Hashemian R. Quick access to circuit transfer functions via NAM determinant/cofactors using UaL technique. *IEEE Trans Circuits Syst II*, 2022, 69: 689–693
- 3 Meng Q Y, Zheng Y P, Jiang Z L. Determinants and inverses of weighted Loeplitz and weighted Foeplitz matrices and their applications in data encryption. *J Appl Math Comput*, 2022, 68: 3999–4015
- 4 Granzio D, Ru B X, Zohren S, et al. MEMe: an accurate maximum entropy method for efficient approximations in large-scale machine learning. *Entropy*, 2019, 21: 551
- 5 Zhu A Q, Jin P Z, Tang Y F. Approximation capabilities of measure-preserving neural networks. *Neural Networks*, 2022, 147: 72–80
- 6 Liu T, Cui H G, Xi Z M, et al. Novel estimators of equivalent number of looks in polarimetric SAR imagery based on sub-matrices. *Sci China Inf Sci*, 2016, 59: 062309
- 7 Yan R N, Hong Q H, Wang C H, et al. Multilayer memristive neural network circuit based on online learning for license plate detection. *IEEE Trans Comput-Aided Des Integr Circuits Syst*, 2021, 41: 3000–3011
- 8 Holzmann M, Moroni S. Orbital-dependent backflow wave functions for real-space quantum Monte Carlo. *Phys Rev B*, 2019, 99: 085121
- 9 Lang S. *Introduction to Linear Algebra*. New York: Springer Science & Business Media, 2012. 195–232
- 10 Defranza J, Gagliardi D. *Introduction to Linear Algebra with applications*. New York: Waveland Press, 2015. 54–65
- 11 Ishikawa M, Okada S, Tagawa H, et al. Generalizations of Cauchy's determinant and Schur's Pfaffian. *Adv Appl Math*, 2006, 36: 251–287
- 12 Elouafi M. A note for an explicit formula for the determinant of pentadiagonal and heptadiagonal symmetric Toeplitz matrices. *Appl Math Computation*, 2013, 219: 4789–4791
- 13 Jiang Z L, Sun J X. Determinant and inverse of a Gaussian Fibonacci skew-Hermitian Toeplitz matrix. *J Nonlinear Sci Appl*, 2017, 10: 3694–3707
- 14 Li Y. An effective hybrid algorithm for computing symbolic determinants. *Appl Math Computation*, 2009, 215: 2495–2501
- 15 Takeshi O. Robust computation of determinant. In: *AIP Conference Proceedings*. Rhodes: American Institute of Physics, 2012. 1504: 1119–1123
- 16 Beliakov G, Matiyasevich Y. A parallel algorithm for calculation of determinants and minors using arbitrary precision arithmetic. *Bit Numer Math*, 2016, 56: 33–50

- 17 Sun Z, Ielmini D. Invited tutorial: analog matrix computing with crosspoint resistive memory arrays. *IEEE Trans Circuits Syst II*, 2022, 69: 3024–3029
- 18 Liu Y Y, Wang C H, Sun J R, et al. One-step calculation circuit of FFT and its application. *IEEE Trans Circuits Syst I*, 2022, 69: 2781–2793
- 19 Deng C L, Tang W M, Liu J N, et al. Reliable single-epoch ambiguity resolution for short baselines using combined GPS/BeiDou system. *GPS Solut*, 2014, 18: 375–386
- 20 Angrisano A, Dardanelli G, Innac A, et al. Performance assessment of PPP surveys with open source software using the GNSS GPS-GLONASS-Galileo constellations. *Appl Sci*, 2020, 10: 5420
- 21 Wang B C, He Y, Wang G H, et al. Optimal allocation of multi-sensor passive localization. *Sci China Inf Sci*, 2010, 53: 2514–2526
- 22 Guan H L, Zhang S S, Wang W-Q. Localization deception performance of FDA signals under passive bi-satellite reconnaissance. *Sci China Inf Sci*, 2021, 64: 192305
- 23 Jing S, Zhan X Q, Lu J, et al. Characterisation of GNSS space service volume. *J Navigation*, 2015, 68: 107–125
- 24 Teng Y L, Wang J L. A closed-form formula to calculate geometric dilution of precision (GDOP) for multi-GNSS constellations. *GPS Solut*, 2016, 20: 331–339
- 25 Doong S H. A closed-form formula for GPS GDOP computation. *GPS Solut*, 2009, 13: 183–190

Design and application of a programmable matrix determinant-solving circuit based on memristors

Pingdan XIAO¹, Qinghui HONG^{2*}, Sichun DU², Chen SUN², Jun LI² & Jiliang ZHANG³

1. *School of Physics and Electronics, Hunan University, Changsha 410082, China;*

2. *College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China;*

3. *College of Semiconductors (College of Integrated Circuits), Hunan University, Changsha 410082, China*

* Corresponding author. E-mail: hongqinghui@hnu.edu.cn

Abstract The determinant of a matrix is of great significance in engineering applications and has been widely used. Obtaining the matrix determinant efficiently and quickly is a major requirement in the IoT and big data era. This paper proposes a programmable analog circuit based on a memristor, which can calculate the determinant of any order in one step. First, we present circuits for second and third order determinants. Circuits for determinants of any order are obtained by combining these circuits. The circuit can also be made programmable by adjusting the conductance of the memristor and varying the input voltage source. PSPICE simulation results show that the accuracy of the circuit is more than 99.4%, and it runs three orders of magnitude faster than the MATLAB simulation. Even when the error interference is 10.0%, the accuracy can still exceed 93%. The circuit has lower area overheads and power consumption than digital processors. Finally, the circuit can calculate the spatial matrix determinant, composed of data from multiple satellites, using the cosine parameters of each satellite to assist in the selection of positioning satellite combinations, and the average accuracy of the calculation results can reach 99.7%.

Keywords memristors, circuit design, programmable, determinant, satellite positioning