

Simulación numérica de circuitos eléctricos a gran escala

Hermann Mena* & Jens Saak**

Resumen—El gran avance tecnológico de las últimas décadas ha incrementado la necesidad de simular numéricamente fenómenos físicos asociados a aplicaciones concretas en: micro-electrónica, micro-mécanica, electromagnetismo, dinámica de fluidos, etc. La mayoría de modelos matemáticos que describen estos fenómenos comparten la característica de que su dimensión es muy grande. En particular, la modelización de circuitos integrados, por ejemplo los microprocesadores, desemboca en un sistema dinámico a gran escala que no puede ser resuelto con métodos numéricos tradicionales. En su defecto, técnicas computacionales de alto desempeño son necesarias e incluso obligatorias para enfrentar este tipo de problemas.

En el presente artículo revisaremos herramientas de computación científica que permiten simular circuitos eléctricos a gran escala. Concretamente, aplicamos técnicas de reducción de modelos que explotan la estructura del problema permitiendo así reducir el costo computacional y de almacenamiento drásticamente.

Los algoritmos involucrados con los métodos propuestos han sido implementados en la *toolbox* para MATLAB, MESS (Matrix Equation Sparse Solver). Detalles de la implementación, mejoras y extensiones de estos algoritmos son discutidas brevemente.

Abstract— The ever developing technology has increased the demand for numerical simulations of the physical phenomena related to: microelectronics, micromechanics, electromagnetism, fluid dynamics, etc. Most of the mathematical models describing these phenomena have in common that their dimension is very large. Particularly, the simulation of Very Large Scale Integration VLSI circuits, like microprocessors, leads to a large scale dynamical system which can not be solved using conventional methods. In stead, high performance computing techniques have to be applied to deal with these problems.

In this paper we review modern tools from scientific computing which allow us to solve large scale problems arising in circuit simulation. Specifically, we use model reduction techniques which reduce both computational and storage costs drastically.

The underlying algorithms have been implemented in the

MATLAB toolbox MESS (Matrix Equation Sparse Solver). Details about the implementation as well as improvements and extensions are reviewed briefly.

Index Terms—Simulación de circuitos eléctricos, Reducción de modelos, Truncamiento balanceado, Ecuaciones de Lyapunov, Ecuaciones Algebraicas/Diferenciales de Riccati (EAR/EDR).

I. INTRODUCCIÓN

Debido a los avances en el proceso de producción de circuitos integrados de las últimas décadas, el tamaño de los dispositivos se ha reducido y la velocidad de los mismos se ha incrementado en una relación inversamente proporcional; así lo afirmó Gordon E. Moore, co-fundador de Intel, el 19 de abril de 1965 en su conocida *Ley de Moore*. Esta ley empírica expresa que aproximadamente cada 18 meses se duplica el número de transistores en un circuito integrado. Aunque esta ley ha perdido validez, tal como lo expresara el propio Moore: *Mi ley dejará de cumplirse dentro de 10 o 15 años desde 2007*; es importante señalar que la simulación numérica ha cumplido, y lo sigue haciendo, un rol fundamental en el diseño de estos dispositivos. Pues permite determinar con alta precisión el comportamiento y desempeño del dispositivo.

El sistema dinámico que describe el circuito es a gran escala y su resolución numérica es un reto en la comunidad científica y constituye, incluso un problema abierto si la dimensión es muy grande ($> 10^8$) y/o si el problema es denso no estructurado. Por la amplia gama de aplicaciones, sistemas con características similares aparecen en general en problemas gobernados por ecuaciones diferenciales parciales, muchas técnicas han sido propuestas para enfrentar estos problemas las cuales se enfocan en reducir el elevado costo computacional, e.g. ver [2], [3], [4], [32]. Es importante señalar que los requerimientos de memoria y tiempo de ejecución están ligados estrictamente a la dimensión del problema.

* Departamento de Matemática, Escuela Politécnica Nacional, Quito - Ecuador, hermann.mena@epn.edu.ec

**Fakultät für Mathematik, Technische Universität Chemnitz, D-09107 Chemnitz, Alemania, jens.saak@mathematik.tu-chemnitz.de

Por esta razón, el costo computacional puede crecer, incluso exponencialmente, si se requiere mayor precisión para resolver el problema numéricamente. Si el costo computacional está altamente ligado a la dimensión del sistema, parece razonable y necesario reducir la dimensión del mismo para resolverlo en tiempo real. Métodos que explotan esta idea son conocidos como reducción de modelos o reducción de orden. Estas técnicas han probado ser particularmente eficientes en la simulación de circuitos eléctricos, e.g. ver [2], [20], [27], [25], [31], [28], [29], [30], [34], [17], [37]. En general estos métodos se clasifican en:

- métodos de proyección,
- métodos *Moment Matching*,
- métodos de truncamiento balanceado.

En el presente trabajo nos enfocaremos en métodos de truncamiento balanceado y su aplicación a la simulación de circuitos a gran escala.

El presente artículo está organizado de la siguiente manera: en la Sección II describimos el modelo matemático que aparece en simulación de circuitos a gran escala, luego en la Sección III describimos la reducción de modelos y la estructura de cierto tipo de ecuaciones matriciales asociadas a esta técnica. Métodos de resolución para estas ecuaciones matriciales son presentados en la Sección IV, así como detalles de su implementación en el *toolbox* para MATLAB MESS (Matrix Equation Sparse Solver). Finalmente, ejemplos numéricos son presentados en la Sección V, para comprobar la precisión y factibilidad de las técnicas propuestas en el presente trabajo. Algunas conclusiones se presentan en la Sección VI.

II. MODELO MATEMÁTICO

Luego de linealizar un sistema no lineal o un sistema de mayor orden se obtiene un sistema lineal no dependiente del tiempo de la forma:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), & t > 0, \\ x(0) &= x_0, \\ y(t) &= Cx(t) + Du(t), & t > 0, \end{aligned} \quad (1)$$

donde $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, y $D \in \mathbb{R}^{p \times m}$ son matrices constantes. En el tipo de problemas que nos interesan la dimensión del sistema n , en general, varía entre $10^3 - 10^7$ dependiendo de la aplicación específica que se considere. Es importante notar que el número de entradas o

controles, representado por $u(t)$, y el número de salidas del sistema, representado por $y(t)$, a menudo es mucho menor que el orden del sistema, i.e. $m, p \ll n$.

Por ejemplo, si se aplica análisis nodal modificado para generar un sistema de ecuaciones para una red RLC multi puerto el sistema tiene la forma:

$$\begin{aligned} \mathcal{L}\dot{x}(t) &= -\mathcal{G}x(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \quad (2)$$

donde

$$\mathcal{L} = \begin{bmatrix} E & 0 \\ 0 & L \end{bmatrix}, \quad \mathcal{G} = \begin{bmatrix} G & M \\ -M^T & 0 \end{bmatrix},$$

$$x = \begin{bmatrix} v \\ i \end{bmatrix}.$$

Los vectores y y u las corrientes de los puertos y sus voltajes, respectivamente. Las matrices E , L , G son simétricas y denotan la capacitancia nodal, inductancia y conductividad, respectivamente. La matriz M representa la matriz de incidencia asociada con el inductor de corriente. La matriz N indica donde se encuentran las fuentes de voltaje y la matriz C especifica donde se miden los puertos de corriente. La matriz \mathcal{L} es no singular, i.e. tiene inversa, por lo cual podemos escribir el sistema (2) en la forma (1).

En la actualidad existen varias librerías y software especializado para tratar numéricamente sistemas de la forma (1). Las librerías que mejor desempeño han mostrado tienen una particularidad en común, todas hacen uso de técnicas eficientes de Algebra Lineal Numérica, como el software más relevante por el impacto que ha tenido en ciencia e ingeniería: MATLAB (Matrix Laboratory) cuyo *kernel* está basado en LAPACK (Linear Algebra Package). Usando MATLAB es posible resolver problemas densos de orden 10^3 en computadores de escritorio e incluso problemas de orden 10^4 si se utiliza la nueva versión del software que corre a 64 bits. Sin embargo, si la complejidad del sistema aumenta o el problema no es estructurado deben aplicarse modernas técnicas de computación científica.

Existen muchos métodos para resolver numéricamente sistemas a gran escala de la forma (1), ver por ejemplo [4], [8]. En la siguiente sección revisaremos brevemente los métodos de reducción de modelos via truncamiento balanceado.

III. REDUCCIÓN DE MODELOS

La idea de la reducción de modelos (RM) es la de encontrar un sistema cuyo orden sea mucho menor que el sistema original y al mismo tiempo este nuevo sistema sea una buena aproximación del sistema original. Concretamente, lo que se busca es que para la misma entrada o control, $u(t)$, la salida del sistema, $y(t)$, represente de buena manera lo que ocurre en el sistema original, i.e. se busca un sistema de la forma:

$$\begin{aligned}\dot{\tilde{x}}(t) &= \tilde{A}\tilde{x}(t) + \tilde{B}u(t), & t > 0, \\ \tilde{x}(0) &= \tilde{x}_0, \\ \tilde{y}(t) &= \tilde{C}\tilde{x}(t) + Du(t), & t > 0,\end{aligned}\quad (3)$$

cuyo orden, r , sea mucho menor que el orden del sistema original, $r \ll n$ y adicionalmente \tilde{y} sea una buena aproximación, de acuerdo a cierto criterio, de la salida del sistema original, y .

Entre los métodos de RM se destacan: Reducción/Substracción-Guyan, Truncamiento modal, Aproximaciones de Padé, Métodos de subespacio de Krylov, Truncamiento balanceado, etc. Una característica de muchos métodos es realizar proyecciones de tipo Galerkin o Petrov-Galerkin del espacio de estado en un subespacio de dimensión menor. Para una introducción a reducción de modelos ver [2], [3]. En este trabajo nos enfocaremos en métodos de truncamiento balanceado. En ingeniería uno de los criterios más utilizados para medir la aproximación de \tilde{y} a y es el denominado función de transferencia. Esto se debe a que bajo ciertas suposiciones, se puede probar que:

$$\|y - \tilde{y}\|_{L_2[0,\infty)} \leq \|G - \tilde{G}\|_\infty \|u\|_{H_2}, \quad (4)$$

donde

$$\begin{aligned}G &:= C(sI - A)^{-1}B + D, \\ \tilde{G} &:= \tilde{C}(sI - \tilde{A})^{-1}\tilde{B} + D,\end{aligned}$$

son las funciones de transferencia de los sistemas (1) y (3), respectivamente.

($\|\cdot\|_{\mathcal{H}}$ denota la norma en el espacio \mathcal{H} , la norma es una generalización del valor absoluto a espacios vectoriales y por tanto determina una forma de medir distancias, ver por ejemplo [24]).

Debido a (4) muchas técnicas de RM tratan de minimizar $\|G - \tilde{G}\|_\infty$, pues al minimizar este término implícitamente están minimizando $\|y - \tilde{y}\|_{L_2[0,\infty)}$, este es el caso de Truncamiento balanceado. Otros criterios de minimización consideran modelos de frecuencia pesados y reducción de controles, ver

[35]. El Truncamiento balanceado se basa en truncar el espacio de transformaciones de estado mediante una aplicación no singular de la forma:

$$\begin{aligned}\mathcal{T} &: (A, B, C, D) \mapsto (TAT^{-1}, TB, T^{-1}C, D) \\ &= \left(\begin{bmatrix} \mathbf{A}_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \begin{bmatrix} \mathbf{B}_1 \\ B_2 \end{bmatrix}, \begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_2 \end{bmatrix}, D \right)\end{aligned}$$

y las matrices del modelo reducido se definen como

$$(\tilde{A}, \tilde{B}, \tilde{C}, D) = (\mathbf{A}_{11}, \mathbf{B}_1, \mathbf{C}_1, D),$$

la aplicación \mathcal{T} balancea el Gramian de controlabilidad, W_c y el Gramian de observabilidad, W_o los cuales corresponden a soluciones de ecuaciones matriciales de Lyapunov de la forma:

$$\begin{aligned}AW_c + W_cA^T + BB^T &= 0, \\ AW_o + W_oA^T + CC^T &= 0.\end{aligned}\quad (5)$$

Antes de explicar algunos métodos para resolver (5), lo cual se profundizará en la Sección IV, es importante señalar que como W_c y W_o son matrices semidefinidas positivas, se pueden factorar de la forma $W_c = S^T S$, $W_o = R^T R$. Más aún si se escogen los factores $S, R \in \mathbb{R}^{n \times n}$ triangulares, estos corresponden a los factores de Cholesky de W_c y W_o , respectivamente. Esta propiedad permite definir el truncamiento balanceado usando el producto SR^T en lugar del producto de los Gramians.

Implementaciones de algoritmos de reducción de modelos y truncamiento balanceado se describen en [4], [35].

El costo computacional de la RM via truncamiento balanceado depende principalmente de la resolución de las ecuaciones de Lyapunov asociadas a los Gramians de observabilidad y controlabilidad.

En sistemas no estables la aplicación de truncamiento balanceado, conlleva la necesidad de resolver ecuaciones algebraicas de Riccati en lugar de ecuaciones de Lyapunov. Las ecuaciones de Riccati aparecen en muchas aplicaciones de ciencia e ingeniería, en especial en la teoría de control. Problemas de optimización gobernados por EDPs de tipo parabólico pueden formularse como problemas de Cauchy abstractos. Imponiendo un funcional de costo cuadrático se obtiene un problema *linear quadratic regulator* (LQR), por sus siglas en inglés, para un sistema de dimensión infinita. El control óptimo para este problema está definido vía *feedback* en términos de la solución de la EDR para operadores, ya sea esta de tipo algebraico o diferencial, dependiendo si el intervalo de integración

que se considera es infino o finito, respectivamente. La ecuación algebraica de Riccati se define como:

$$0 = \mathfrak{R}_h(X) = C^T C + A^T X + X A - X B B^T X, \quad (6)$$

y la ecuación diferencial de Riccati:

$$\dot{X} = -C^T C - A^T X - X A + X B B^T X. \quad (7)$$

La relación entre estas ecuaciones y la ecuación de Lyapunov se revisará en la Sección IV.

IV. SIMULACIÓN NUMÉRICA

Como se ha discutido en las Sección III, la aplicación de RM via truncamiento balanceado a sistemas lineales a gran escala implica el resolver ecuaciones de Lyapunov o ecuaciones de Riccati, dependiendo de si el sistema es estable o no. Estas últimas ecuaciones aparecen también en control óptimo, y más aún están escriticamente ligadas unas a otras como veremos más adelante.

Muchas librerías y software han sido propuestos en la literatura especializada. Particularmente, el software `Lyapack`, presentado en el año 2000, ha demostrado ser una valiosa herramienta para resolver eficientemente este tipo de ecuaciones en problemas de reducción de modelos y control a gran escala. En esta Sección discutimos recientes mejoras y extensiones del mencionado software, las cuales están incluidas en el software sucesor denominado MESS (Matrix Equation Sparse Solver). Los últimos resultados en la teoría de ecuaciones matriciales, sobrepasan los límites del presente artículo, y serán tratados en un trabajo futuro.

Las mejoras de MESS respecto a su antecesor son entre otras:

- menor tiempo de ejecución, principalmente debido a la implementación eficiente de los principales algoritmos ligados al software
- amplia gama de problemas que se pueden tratar, específicamente en MESS es posible resolver problemas de control donde el intervalo de integración es finito lo cual en muchas aplicaciones representa una herramienta fundamental en el modelamiento del problema
- interfaces amigables y fácil llamado de funciones, las interfaces han sufrido un cambio drástico para facilidad del usuario. Estas se basan en interfaces similares a las usadas en funciones de MATLAB.

Todos los detalles concernientes a MESS no están cubiertos en el presente trabajo, los lectores interesados pueden ver [14], nos enfocaremos en como resolver las ecuaciones matriciales descritas en la Sección III.

La estructura de la ecuación (6) es compartida por la ecuación (5) excepto por el término no lineal, i.e. $X B B^T X$. Lo mismo ocurre con el lado derecho de la ecuación (7). Y es precisamente el término no lineal lo que diferencia las ecuaciones algebraicas de Riccati de las ecuaciones de Lyapunov. La pregunta que surge es: ¿Es posible linealizar el término no lineal?. La respuesta a esta pregunta es afirmativa, y es precisamente la base de la gran mayoría de métodos para resolver ecuaciones no lineales en general: linealizar. Sin el término no lineal, se resuelven ecuaciones de Lyapunov en lugar de ecuaciones Riccati, como explicamos a continuación.

Observemos que la derivada (de Frechét, la cual es una generalización de la derivada usual a espacios de Banach, ver [24]) del operador \mathfrak{R}_h en P está dada por el operador de Lyapunov:

$$\mathfrak{R}'_h|_P : Q \mapsto (A - B R^{-1} B^T P)^T Q + Q (A - B B^T P),$$

por lo que el método de Newton aplicado a EAR puede ser escrito como

$$\begin{aligned} N_\ell &:= \left(\mathfrak{R}'_h|_{P_\ell} \right)^{-1} \mathfrak{R}_h(P_\ell), \\ X_{\ell+1} &:= X_\ell + N_\ell. \end{aligned}$$

y por tanto para una matriz inicial dada, este método puede implementarse como:

Algorithm IV.1 Método de Newton para EAR

Require: P_0 , tal que A_0 es estable

- 1: $A_\ell \leftarrow A_h - B_h B_h^T P_\ell$
 - 2: Resolver la ecuación de Lyapunov $A_\ell^T N_\ell + N_\ell A_\ell = -\mathfrak{R}_h(P_\ell)$
 - 3: $P_{\ell+1} \leftarrow P_\ell + N_\ell$
-

(En las aplicaciones consideradas aquí, es posible usar el hecho de que para intervalos pequeños, la solución aproximada de $X_k \approx X(t_k)$ en general es un valor estabilizado muy bueno para empezar la iteración, ver [9].)

Por lo tanto, toda A_ℓ es estable y la iteración P_ℓ converge a la solución, P_* , en orden cuadrático.

Es importante notar que el método de Newton para EAR puede ser reformulado como una iteración de un solo paso, si se re-escribe de tal manera que la siguiente iteración es calculada directamente de la ecuación de Lyapunov que se resuelve en el paso 2 del Algoritmo IV.1, i.e.,

$$(A - BR^{-1}B^T P_\ell)^T P_{\ell+1} + P_{\ell+1}(A - BB^T P_\ell) = -C^T \tilde{Q}C - P_\ell BR^{-1}B^T P_\ell =: -W_\ell W_\ell^T$$

De esta manera para resolver la EAR es necesario resolver una ecuación de Lyapunov en cada iteración, de la forma:

$$F^T X + X F = -W W^T \quad (8)$$

siendo F una matriz estable en cada iteración. Usualmente el método de Newton converge muy rápido; de hecho, en problemas de transferencia de calor en 2D la convergencia se alcanza en 3-5 iteraciones.

Para resolver ecuaciones de Lyapunov a gran escala usaremos el método ADI *alternating direction implicit* el mismo que se escribe como, ver [36],

$$\begin{aligned} (F^T + p_j I) Q_{(j-1)/2} &= -W W^T - Q_{j-1}(F - p_j I), \\ (F^T + \bar{p}_j I) Q_j^T &= -W W^T - Q_{(j-1)/2}(F - \bar{p}_j I), \end{aligned}$$

donde \bar{p} denota el conjugado de $p \in \mathbb{C}$. Si los parámetros p_j se escogen apropiadamente, entonces $\lim_{j \rightarrow \infty} Q_j = Q$ converge en orden super-lineal.

Sin embargo, en problemas a gran escala no es posible aplicar directamente el método ADI, incluso no es posible guardar completamente las matrices que aparecen en cada paso; pues lo anterior excede los límites de memoria fácilmente. Por esta razón, es necesario utilizar técnicas de rango bajo para su resolución, concretamente utilizaremos el método de rango bajo Newton-ADI presentado en [7], [31].

Algorithm IV.2 Método de rango bajo ADI

Require: F , W y el conjunto de parámetros ADI $\{p_1, \dots, p_k\}$

Ensure: $Z = Z_{i_{max}} \in \mathbb{C}^{n, i_{max} n_w}$ tal que $Z Z^T \approx X$.

- 1: $V_1 = \sqrt{-2\text{Re}(p_1)}(F^T + p_1 I)^{-1} W$
 - 2: $Z_1 = V_1$
 - 3: **for** $j = 2, 3, \dots$ **do**
 - 4: $V_j = \frac{\sqrt{\text{Re}(p_j)}}{\sqrt{\text{Re}(p_{j-1})}} \left(I - (p_j + \bar{p}_{j-1})(F^T + p_j I)^{-1} \right) V_{j-1}$
 - 5: $Z_j = \begin{bmatrix} Z_{j-1} & V_j \end{bmatrix}$
 - 6: **end for**
-

En el Algoritmo IV.2, Z es una matriz de rango bajo que aproxima X , i.e., $Z \in \mathbb{R}^{n \times q}$ con $q \ll n$. La aproximación de la matriz Z es muy precisa, de hecho no se requiere un costo computacional adicional para conseguir resultados similares a los que se obtienen con métodos convencionales.

Por otro lado, cabe señalar que todas las matrices V_j tienen igual número de columnas a $W \in \mathbb{R}^{n \times n_w}$, i.e., en cada iteración j , es necesario resolver w sistemas de ecuaciones lineales con los mismos coeficientes matriciales $F^T + p_j I$. En la implementación del algoritmo se explota esta propiedad. Si la convergencia del método de rango bajo ADI se alcanza con respecto a cierto criterio de parada después i_{max} pasos, entonces $Z_{i_{max}} = [V_1, \dots, V_{i_{max}}] \in \mathbb{R}^{n \times i_{max} n_w}$, con $V_j \in \mathbb{R}^{n \times n_w}$. Cuando n es grande y n_w pequeño se espera que $r_i := i_{max} n_w \ll n$. En este caso se ha calculado una aproximación de rango bajo $Z_{i_{max}}$ del factor Z de la solución X , esto es $Y = Z Z^T \approx Z_{i_{max}} Z_{i_{max}}^T$. Sin embargo, si $n_w \cdot i_{max}$ se crece demasiado el método pierde eficiencia. Para contrarrestar el crecimiento de $n_w \cdot i_{max}$, deben ser aplicadas técnicas de compresión de columnas para reducir el número de columnas en $Z_{i_{max}}$, sin aumentar un error significativo, ver [16], [18].

Si combinamos el Algoritmo IV.2 con el método de Newton, IV.1, se puede definir un método de rango bajo para resolver ecuaciones algebraicas de Riccati denominado Newton-ADI. El algoritmo que describe este método se presenta a continuación:

Algorithm IV.3 Método de rango bajo Newton-ADI

Require: $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$ y $X_0 \in \mathbb{R}^{n \times n}$

Ensure: Z_k tal que $X_k \approx Z_k Z_k^T$.

- 1: $K^0 = X_0 B$
 - 2: **for** $j = 1$ to j_{max} **do**
 - 3: Determinar parámetros (sub)óptimos ADI p_1^j, p_2^j, \dots con respecto a la matriz $F^j = A - K^j B^T$.
 - 4: $G^j = [\hat{C}_{k+1}^T K^{j-1}]$.
 - 5: Calcular Z^j usando el Algoritmo IV.2 de manera que el producto de los factores de rango bajo $Z^j Z^{jT}$ aproxime la solución de $F^{jT} X^j + X^j F^j = -G^j G^{jT}$.
 - 6: $K^j = Z^j (Z^{jT} B)$.
 - 7: **end for**
 - 8: $Z_{k+1} = Z^{j_{max}}$.
-

En teoría de control la matriz $K^{j_{max}}$ se conoce como la matriz *feedback*, puesto que el control del sistema esta definido como:

$$u(t) = -K^{j_{max}T}x(t),$$

es decir, para controlar el sistema no se necesita conocer explícitamente la matriz X , solución de la ecuación de Riccati. Lo único que se necesita es la matriz *feedback*, esta propiedad es ampliamente explotada en la implementación pues se puede re-escribir el método de rango bajo Newton-ADI de manera que la iteración se realice directamente en función de factores que aproximen la matriz *feedback*, ver [31].

Las ecuaciones diferenciales de Riccati aparecen en muchas aplicaciones de ciencia e ingeniería, en especial en teoría de control, ver por ejemplo [1], [19], [21], [33]. En reducción de modelos estas ecuaciones aparecen cuando se consideran sistemas no estables dependientes del tiempo. Por ser ecuaciones diferenciales su resolución numérica constituye un reto aún más grande que el de resolver ecuaciones de Lyapunov o ecuaciones algebraicas de Riccati. Esto se debe, principalmente, a que estas ecuaciones son diferenciales matriciales, lo cual implica que al menos se debe resolver una ecuación algebraica en cada paso de integración. En general, se espera que las EDRs sean rígidas (*stiff*), por lo que nos enfocaremos en métodos que puedan tratar el fenómeno de la rigidez eficientemente. También se quiere explotar la estructura de las matrices coeficientes, y por que no de la estructura en si de la ecuación, i.e., aplicar métodos numéricos de EDOs que preserven la estructura de la EDR.

Existen muchas interrogantes tales como: ¿se pueden aplicar los métodos de resolución de EDOs a ecuaciones matriciales?, ¿cuáles métodos preservan la estructura de la EDR?, ¿cuál es la mejor opción para resolver problemas rígidos?, ¿son éstos métodos factibles para problemas a gran escala?, etc. Muchas preguntas se han resuelto completamente; otras quedan abiertas. Concretamente, entre los métodos de un paso y paso múltiple para resolver sistemas de ecuaciones diferenciales ordinarias rígidos los métodos BDF (*backward differentiation formulae*) y los métodos de tipo Rosenbrock son usados comúnmente. Hemos propuesto versiones matriciales de estos algoritmos aplicables a EDRs a gran escala, ver [9], [11], [10], [12], [13], [14]. La idea fundamental que se ha propuesto es realizar una

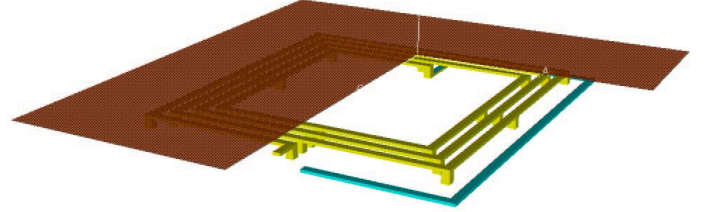


Figura 1. Inductor espiral con placa de cobre en la superficie

aproximación de rango bajo a la solución de la EDR. La descripción de los métodos para resolver EDR a gran escala así como detalles de la implementación de los mismos en MESS se pueden ver en, [14], [15].

V. EJEMPLOS NUMÉRICOS

V-0a. Modelo PEEC de un inductor espiral: Consideremos el modelo PEEC de un inductor espiral, ver [26]. En la Figura 1 se muestra un inductor espiral con una placa de cobre en la superficie. Un sistema dinámico de la forma (1) es obtenido utilizando el software *Fasthenry*, ver [22], [23]. El sistema que se obtiene es:

$$\begin{aligned}\dot{\tilde{x}}(t) &= \tilde{A}\tilde{x}(t) + \tilde{B}u(t), \\ y(t) &= \tilde{B}^T\tilde{x}(t),\end{aligned}\tag{9}$$

donde

$$\tilde{A} = -\tilde{L}^{-\frac{1}{2}}\tilde{R}\tilde{L}^{-\frac{1}{2}}, \quad \tilde{B} = B\tilde{L}^{-\frac{1}{2}},$$

la matriz de inductancia \tilde{L} es simétrica y definida positiva, $\tilde{L}^{\frac{1}{2}}$ representa la raíz cuadrada, i.e. se satisface $\tilde{L}^{\frac{1}{2}}\tilde{L}^{\frac{1}{2}} = \tilde{L}$, \tilde{R} es la matriz de resistencia la cual es cuadrada y dispersa. Referimos [26], [22], [23] para una versión detallada del modelo.

Cabe señalar que \tilde{A} no se calcula explícitamente, por su elevado costo computacional; en su lugar las operaciones para la matriz \tilde{A} se realizan implícitamente.

Hemos aplicado reducción de modelos via truncamiento balanceado, utilizando MESS, al sistema (9) cuyo orden original es $n = 1434$. El modelo reducido ha sido calculado de manera que satisfaga

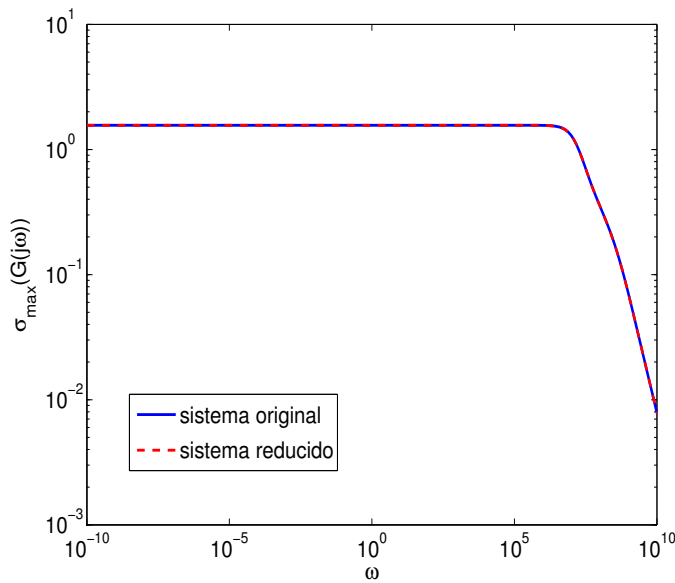


Figura 2. Funciones de transferencia del modelo original y reducido

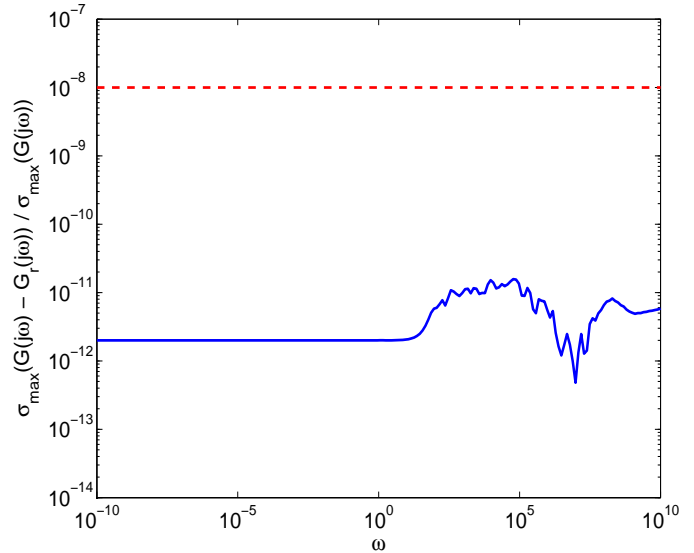


Figura 4. Error relativo del modelo reducido

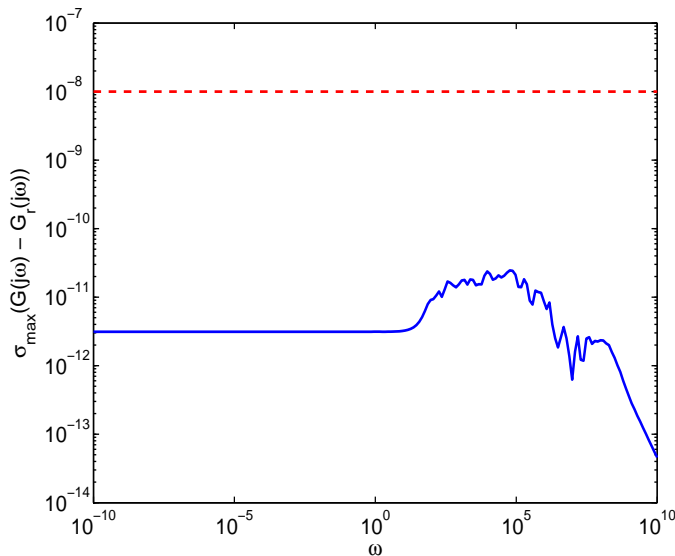


Figura 3. Error absoluto del modelo reducido

una tolerancia de 10^{-8} , se obtuvo un modelo de orden $r = 18$. En la Figura 2 se muestra el valor de las funciones de transferencia, la sobreposición de estos valores constituye un indicador cuantitativo de la eficiencia del modelo reducido para describir la dinámica del modelo original. En efecto, el modelo reducido aproxima al modelo original con 10 dígitos de precisión, ver Figuras 3 y 4 del error absoluto y relativo, respectivamente.

VI. CONCLUSIONES

Es posible simular eficientemente circuitos eléctricos a gran escala mediante la aplicación de modernas herramientas de la computación científica. La idea fundamental es explotar la estructura del problema y reducir el costo computacional. Los algoritmos presentados en este artículo han sido implementados eficientemente, por los autores, en el *toolbox* para MATLAB, MESS. Este software nos permite resolver ecuaciones matriciales a gran escala que aparecen en reducción de modelos y control óptimo como son la ecuación de Lyapunov y las ecuaciones de Riccati, tanto algebraicas como diferenciales. En la nueva versión de MESS se espera incluir nuevos métodos para resolver ecuaciones de Lyapunov como son los métodos de Krylov y también incluir ecuaciones más generales que aparecen en sistemas gobernados por ecuaciones diferenciales algebraicas así como sistemas que no se pueden escribir en forma estándar debido a la singularidad de las matrices involucradas. La versión de MESS implementada en C está en preparación.

Si los problemas son *muy grandes* o *no estructurados* las técnicas explicadas en el presente trabajo no se pueden aplicar directamente. En este caso es necesario combinar estas ideas con técnicas de matrices jerárquicas, ver [5], [6].

REFERENCIAS

- [1] H. ABOU-KANDIL, G. FREILING, V. IONESCU, AND G. JANK, *Matrix Riccati Equations in Control and Systems Theory*, Birkhäuser, Basel, Switzerland, 2003. 6
- [2] A. C. ANTOUNAS, *Lectures on the approximation of linear dynamical systems*, in Encyclopedia of Electrical and Electronics Engineering, John Wiley and Sons, 1999, pp. 403–422. 1, 2, 3
- [3] A. C. ANTOUNAS, D. C. SORESENSEN, AND S. GUGERCIN, *A survey of model reduction methods for large-scale systems*, Contemporary Mathematics, 280 (2001), pp. 193–219. 1, 3
- [4] P. BENNER, *Solving large-scale control problems*, IEEE Control Systems Magazine, 14(1) (2004), pp. 44–59. 1, 2, 3
- [5] P. BENNER AND U. BAUR, *Gramian-based model reduction for data-sparse systems*, SIAM J.Sci.Comput, 31 (2008), pp. 776–798. 7
- [6] ———, *Low rank solution of data-sparse sylvester equations*, Numerical Linear Algebra with Applications, 15 (2008), pp. 837–851. 7
- [7] P. BENNER, J.-R. LI, AND T. PENZL, *Numerical solution of large Lyapunov equations, Riccati equations, and linear-quadratic control problems*, Numer. Linear Algebra Appl., 15 (2008), pp. 755–777. 5
- [8] P. BENNER, V. MEHRMANN, AND D. SORESENSEN, eds., *Dimension Reduction of Large-Scale Systems*, vol. 45 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, Berlin/Heidelberg, Germany, 2005. 2
- [9] P. BENNER AND H. MENA, *BDF methods for large-scale differential Riccati equations*, in Proc. of Mathematical Theory of Network and Systems, MTNS 2004, B. D. Moor, B. Motmans, J. Willems, P. V. Dooren, and V. Blondel, eds., 2004. 4, 6
- [10] ———, *Numerical solution of large-scale differential riccati equations*, Chemnitz Scientific Computing Preprints, TU Chemnitz, May 2009. 6
- [11] ———, *Rosenbrock methods for solving differential riccati equations*, Chemnitz Scientific Computing Preprints, TU Chemnitz, May 2009. 6
- [12] ———, *Symplectic integrators for solving large-scale differential riccati equations*, Chemnitz Scientific Computing Preprints, May 2009. 6
- [13] P. BENNER, H. MENA, AND J. SAAK, *On the parameter selection problem in the Newton-ADI iteration for large-scale Riccati equations*, Electronic Transactions on Numerical Analysis, 29 (2008), pp. 136–149. 6
- [14] ———, *M.E.S.S. 1.0 User Manual*, Chemnitz Scientific Computing Preprints, TU Chemnitz, May 2009. 4, 6
- [15] P. BENNER AND J. SAAK, *Efficient balancing based mor for second order systems arising in control of machine tools*, in Proceedings of the MathMod 2009, I. Troch and F. Breitenecker, eds., vol. 2 of ARGESIM-Reports, 2009. 6
- [16] C. BISCHOF AND G. QUINTANA-ORTÍ, *Computing rank-revealing QR factorizations of dense matrices*, ACM Transactions on Mathematical Software, 24(2) (1998), pp. 226–253. 5
- [17] J. CONG, J. R. SHINNERL, M. XIE, T. KONG, AND X. YUAN, *Large-scale circuit placement*, ACM Trans. Des. Autom. Electron. Syst., 10 (2005), pp. 389–430. 2
- [18] S. GUGERCIN, D. SORESENSEN, AND A. ANTOUNAS, *A modified low-rank Smith method for large-scale Lyapunov equations*, Numer. Algorithms, 32(1) (2003), pp. 27–55. 5
- [19] A. ICHIKAWA AND H. KATAYAMA, *Remarks on the time-varying H_∞ Riccati equations*, Sys. Cont. Lett., 37(5) (1999), pp. 335–345. 6
- [20] M. IORDACHE AND L. DUMITRIU, *Efficient decomposition techniques for symbolic analysis of large-scale analog circuits by state variable method*, Analog Integr. Circuits Signal Process., 40 (2004), pp. 235–253. 2
- [21] O. JACOBS, *Introduction to Control Theory*, Oxford Science Publication, Oxford, 2nd ed., 1993. 6
- [22] M. KAMON, M. TSUK, AND J. WHITE, *Fasthenry: A multipole-accelerated 3-d inductance extraction program.*, IEEE Transactions on Microwave Theory and Techniques, 42 (1994), pp. 1750–1758. 6
- [23] M. KAMON, F. WANG, AND J. WHITE, *Generating nearly optimal compact models from krylov-subspace based reduced order models*, IEEE Transactions On Circuits and Systems-II: Analog and Digital Signal Processing, 47 (2000), pp. 239–248. 6
- [24] A. KOLMOGOROV AND S.V. FOMIN, *Elements of the Theory of Functions and Functional Analysis*, Dover Publications, 1999. 3, 4
- [25] J. LI AND J. WHITE, *Low rank solution of Lyapunov equations*, SIAM J. Matrix Anal. Appl., 24(1) (2002), pp. 260–280. 2
- [26] J.-R. LI AND M. KAMON, *PEEC model of a spiral inductor generated by Fasthenry*, in Dimension Reduction of Large-Scale Systems, P. Benner, V. Mehrmann, and D. Sorensen, eds., Lecture Notes in Computational Science and Engineering, Springer-Verlag, Berlin/Heidelberg, Germany, 2005. 6
- [27] J.-R. LI AND J. WHITE, *Reduction of large circuit models via low rank approximate gramians*, International Journal of Applied Mathematics and Computer Science, 11 (2001), pp. 101–121. 2
- [28] E.ÑAROSKA, U. SCHWIEGELSHOHN, R.-J. SHANG, AND F. LAI, *Hybrid parallel circuit simulation approaches*, in PACT '00: Proceedings of the 2000 International Conference on Parallel Architectures and Compilation Techniques, Washington, DC, USA, 2000, IEEE Computer Society, p. 261. 2
- [29] O.ÑASTOV, R. TELICHEVESKY, K. KUNDERT, AND J. WHITE, *Fundamentals of fast simulation algorithms for rf circuits*, Invited paper to Appear, Proceedings of the IEEE Special Issue on Tools for Mixed Signal Design, (2009). 2
- [30] H.ÑETO, L. M. SILVEIRA, J. WHITE, AND L. VIDIGAL, *On exponential fitting for circuit simulation*, IEEE Transactions on Computer-Aided Design, 11 (1992), pp. 566–574. 2
- [31] T. PENZL, *LYAPACK Users Guide*, Tech. Rep. SFB393/00-33, Sonderforschungsbereich 393 Numerische Simulation auf massiv parallelen Rechnern, TU Chemnitz, 09107 Chemnitz, Germany, 2000. Available from <http://www.tu-chemnitz.de/sfb393/sfb00pr.html>. 2, 5, 6
- [32] ———, *Algorithms for model reduction of large dynamical systems*, 415 (2006), pp. 322–343. (Reprint of Technical Report SFB393/99-40, TU Chemnitz, 1999.). 1
- [33] I. PETERSEN, V. UGRINOVSKII, AND A.V.SAVKIN, *Robust Control Design Using H^∞ Methods*, Springer-Verlag, London, UK, 2000. 6
- [34] A. RUEHLI, *Survey of analysis, simulation and modeling for large scale logic circuits*, in DAC '81: Proceedings of the 18th conference on Design automation, Piscataway, NJ, USA, 1981, IEEE Press, pp. 124–129. 2
- [35] A. VARGA, *Model reduction software in the slicot library*, in Applied and Computational Control, Signals, and Circuits, volume 629 of The Kluwer International Series in Engineering and Computer Science, Kluwer Academic Publishers, 2001, pp. 239–282. 3
- [36] E. WACHSPRESS, *Iterative solution of the Lyapunov matrix equation*, Appl. Math. Letters, 107 (1988), pp. 87–90. 5
- [37] J. WHITE AND S. LEEB, *An envelope-following approach to switching power converter simulation*, IEEE Transactions on Power Electronics, 6 (1991), pp. 303–308. 2



Hermann Mena recibió su grado de Matemático en la Escuela Politécnica Nacional (2003) y su grado de Doctor en Matemática (Ph.D.) en un programa conjunto con la Universidad de Berlín, Alemania; bajo la supervisión del Prof. Dr. Peter Benner. De enero 2007 a diciembre 2007 realizó un posdoctorado en la Universidad Técnica de Chemnitz en Alemania. Desde enero 2008 es profesor titular del Departamento

de Matemática en la Escuela Politécnica Nacional. En el semestre de primavera 2009 es profesor visitante del Massachusetts Institute of Technology (MIT) en Cambridge, Estados Unidos; invitado por el Prof. Gilbert Strang. Sus áreas de investigación son: computación científica, análisis numérico, algoritmos paralelos, optimización y su aplicación a la teoría de control. Los resultados de su investigación han sido publicados en revistas internacionales referadas y presentados a la comunidad científica en más de veinte ponencias en Europa y Estados Unidos.



Jens Saak recibió su Master en Tecnomatemáticas en la Universidad de Bremen en 2003. Desde entonces ha trabajado en proyectos de investigación acerca de problemas de control gobernados por ecuaciones diferenciales parciales y reducción de modelos de sistemas a gran escala. Actualmente cursa su último año de doctorado en la Universidad Técnica de Chemnitz, Alemania bajo la supervisión del Prof. Dr. Peter Benner.

Sus áreas de investigación son: problemas de control gobernados por ecuaciones diferenciales parciales y técnicas de truncamiento balanceado en reducción de modelos de sistemas de primer y segundo orden. Particularmente, software para ecuaciones matriciales a gran escala. Los resultados de su investigación han sido publicados en revistas internacionales referadas y presentados a la comunidad científica en más de doce ponencias en Europa.