

Extracción de Modelos Comportamentales para Transistores de Alta Frecuencia

Fernando Martín-Rodríguez⁽¹⁾, María Rocío Moure⁽¹⁾, Ainhoa Morales-Fernández⁽¹⁾,
Mónica Fernández-Barciela⁽¹⁾, Paul J. Tasker⁽²⁾

fmartin@uvigo.es, monica.barciela@uvigo.es.

⁽¹⁾AtlantTIC, research center for Telecommunication Technologies, University of Vigo.
C/Maxwell SN. 36310 Vigo (Pontevedra, España).

⁽²⁾Cardiff School of Engineering, Cardiff University, Cardiff (UK).

Abstract- This communication describes the software tools developed in MATLAB environment to extract behavioral models for microwave transistors from RF calibrated experimental data (Load-Pull measurements). These tools are designed to extract both linear and nonlinear frequency domain models. Linear models are limited to a single impedance and small signal operation. In the linear case, small-signal S, Y and Z parameters are computed. Extracted nonlinear models are based on Cardiff model formulations: Cardiff conventional A-B model, Cardiff Admittance I-V model and Cardiff advanced formulations. Auxiliary apps have been designed to import calibrated measurement data and also for drawing test graphs to evaluate models extraction. Finally, another type of model has been explored: artificial neural networks based models. Hence, another MATLAB tool has been designed for training shallow neural networks (a Multi-Layer Perceptron, MLP; with one hidden layer trained for approximate fitting). A quality FoM (SNR) is computed comparing measurements with model predictions for all extracted models obtaining good results for all methods and allowing a comparison between them.

I. INTRODUCCIÓN

Esta comunicación describe el desarrollo de varias herramientas para la extracción de modelos comportamentales de transistores de microondas tipo FET a partir de medidas calibradas de load-pull obtenidas con un analizador de redes no lineal (NVNA) [1]. El entorno de desarrollo ha sido MATLAB [2] utilizando además la herramienta GUIDE (Graphical User Interface Design Environment [3]) para obtener aplicaciones gráficas de uso sencillo. La aplicación principal **TotalModel**, parte de un conjunto de medidas indexado por potencia RF de entrada y obtiene dos tipos de modelos: modelos lineales obtenidos a una impedancia de carga concreta, y valores bajos de potencia, y modelos no lineales, extrayendo los coeficientes para un intervalo de potencias de entrada.

Los modelos lineales comportamentales extraídos son: parámetros S, Y y Z [4] calculados sólo a potencias de entrada muy bajas (pequeña señal) y re-indexando los datos para hacer el cálculo para cada impedancia de carga (Z_L).

Otro tipo de modelos extraídos son los comportamentales no lineales denominados modelos de Cardiff, tanto el convencional definido en función de los fasores que representan ondas de potencia A y B, como el de admitancias, definido en función de los fasores V e I [5]. Se trata en este caso de modelos polinómicos generalizados de orden de mezcla variable. La aplicación calcula el modelo Cardiff convencional, para distintos órdenes de mezcla, el

modelo Cardiff general filtrado (que es una simplificación del anterior) y los modelos Cardiff multi-armónicos (Cardiff formulación avanzada [6]) y Cardiff smoothed que se construye a partir del anterior de acuerdo con [6].

La aplicación **TotalModel** va acompañada de otras dos aplicaciones auxiliares: **ImportApp** y **GraphEngine**. Como sus nombres indican, la primera sirve para leer los datos originales de medida load-pull y convertirlos al formato adecuado; mientras que la segunda genera los modelos extraídos obteniendo gráficas y datos de comprobación (sólo para los modelos Cardiff).

Se ha desarrollado otra aplicación independiente llamada: **NeuralModel**. En este caso se realiza una agrupación por potencias y en cada intervalo se entrena una red neuronal del tipo aproximativo (fitting neural net). Esta idea de modelos basados en IA está ganando fuerza en la comunidad del modelado actual [7]. Para este nuevo modelo, se ha elegido una red tipo (MLP: Multi Layer Perceptron) con un nivel oculto y una función de activación identidad en la salida para obtener las aproximaciones deseadas.

Para todos los modelos, se obtienen métricas de error tras el proceso de extracción.

II. DESCRIPCIÓN DE MODELOS Y MÉTODOS DE EXTRACCIÓN

En todo lo que sigue se asumirá la descripción clásica del FET como caja negra o cuadripolo (figura 1) con dos puertos: el 1 o entrada y el 2 o salida. Las variables físicas habituales son aquí, los fasores de tensión en ambos puertos (V_1 y V_2) y los de corriente entrantes (I_1 e I_2). A frecuencias altas (longitud de onda muy baja), donde no es posible medir los parámetros Z e Y, se miden las relaciones entre los fasores de las ondas incidentes en ambos puertos (A_1 y A_2) y las ondas salientes en ambos puertos (B_1 y B_2).

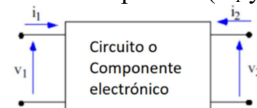


Fig. 1. Cuadripolo genérico.

En el programa dispondremos de conjuntos de medidas (fasores) consideradas como vectores de valores: (V_1, V_2, I_1, I_2) o bien (A_1, A_2, B_1, B_2). Generar un modelo consiste en hallar una relación matemática que permita calcular I_1 e I_2 a partir de V_1 y V_2 (modelo VI o de admitancias); o bien calcular B_1 y B_2 a partir de A_1 y A_2 modelo comportamental AB (o de parámetros S, en el caso de pequeña señal, o de parámetros X, en el caso de gran señal, por ejemplo).

Los valores numéricos para todos los fasores serán complejos y se obtendrán para los diferentes múltiplos enteros de la frecuencia de trabajo (armónicos). Un valor de tensión V_{11} (puerto 1, armónico fundamental) es un número complejo:

$$V_{11} = Re(V_{11}) + j Im(V_{11}) = |V_{11}|e^{j arg(V_{11})} \quad (1)$$

Que representa una señal temporal o forma de onda:

$$v_{11}(t) = Re(V_{11} e^{j 2\pi f_0 t}) = |V_{11}| \cos(2\pi f_0 t + arg(V_{11})) \quad (2)$$

Nótese que el conjunto de fasores para una variable dada puede verse como los coeficientes de Fourier de la forma de onda total.

A. Obtención de medidas

Las medidas, realizadas para una única frecuencia fundamental f_0 y sus armónicos, se obtienen utilizando un sistema de medida de gran señal calibrado, basado en un PNA-X, un tuner multi-armónico y hardware externo (acopladores, atenuadores...). Este tipo de sistemas obtienen directamente las ondas A y B (cuyas unidades son \sqrt{Watt}) en el dominio del tiempo o de la frecuencia. Los fasores V e I se pueden calcular a partir de las ondas medidas A y B [6].

B. Modelos Lineales

Los modelos de caja negra comportamentales más básicos que podemos utilizar son los obtenidos a partir de los parámetros S medidos a f_0 . Así, los más conocidos, son los correspondientes a los parámetros S, Y y Z, definidos en función de los fasores A, B o V, I de la siguiente forma:

$$\begin{aligned} B_1 &= S_{11} A_1 + S_{12} A_2 & I_1 &= Y_{11} V_1 + Y_{12} V_2 \\ B_2 &= S_{21} A_1 + S_{22} A_2 & I_2 &= Y_{21} V_1 + Y_{22} V_2 \\ V_1 &= Z_{11} I_1 + Z_{12} I_2 & & \\ V_2 &= Z_{21} I_1 + Z_{22} I_2 & & \end{aligned} \quad (3)$$

Extraer el modelo, implica obtener los cuatro parámetros de cada uno de estos sistemas (o una matriz compleja 2x2). La información de continua no está contenida en el modelo, si bien está implícita, pues las medidas se realizan para un punto de polarización concreto del transistor. Centrándonos en el caso S, cada medida es un vector complejo de cuatro componentes: (A_1, A_2, B_1, B_2) . Considerando los parámetros Y/S/Z como incógnitas vamos a tener dos ecuaciones por medida. Con dos medidas, ya se podría extraer un modelo.

En la práctica tendremos más de dos medidas (las que se han realizado para una impedancia Z_L , habitualmente 50 Ohm, y potencias menores al límite marcado por el usuario). En esta situación se plantea un sistema de ecuaciones super-determinado (con más ecuaciones que incógnitas):

$$Mx = n \quad (4)$$

M es una matriz compleja que en este caso tendrá 4 columnas (número de incógnitas) y un número de filas igual al doble del número de medidas. Este sistema no tiene solución exacta pero sí que se puede resolver en términos de mínimo error cuadrático, pasando al sistema de 4 ecuaciones con 4 incógnitas [8]:

$$([\text{Conj}(M)]^T M) x = ([\text{Conj}(M)]^T n) \quad (5)$$

Donde se aplica el conjugado complejo $\text{Conj}(\cdot)$, nótese que todas las matrices, datos y soluciones son complejas. La matriz $M^* = ([\text{Conj}(M)]^T M)$ es, por construcción, una matriz cuadrada simétrica y definida positiva.

En algunos casos, M^* , puede resultar mal condicionada para resolver el sistema. Esto es: el sistema está muy cerca de no admitir solución. Aunque pueda calcularse una solución, es probable que contenga errores importantes. Por eso se recurre al siguiente procedimiento de cálculo:

- Calcular el parámetro "Rcond" para la matriz del sistema. Rcond: "reciprocal condition of matrix in 1-norm". El resultado siempre está en el intervalo [0,1]. Un valor cercano a 0 indica mal condicionamiento.
- Por debajo de cierto umbral (10^{-3}), el cálculo se hace factorizando la matriz por el método de Cholesky.
- La factorización de Cholesky siempre debería ser posible para una matriz definida positiva, pero a veces, el mal condicionamiento hace que el cálculo fracase. En este caso se aplica la factorización LU.

C. Modelos no lineales de Cardiff

Los modelos de Cardiff, resultantes de la aplicación de la teoría de mezcla en un sistema no lineal, no resultan de ningún tipo de linealización (a diferencia de los modelos de parámetros X) y se construyen como modelos polinómicos generalizados. Se pueden definir en el dominio AB, o en el dominio VI, o modelo de admitancias. En cada tipo, el modelo obtiene las variables de salida (B's o I's) a partir de las de entrada (A's o V's) con ecuaciones del tipo:

$$B_{im} = \sum_{k=1}^K L_{ik}(m) |A_{21}|^{p_k} [e^{j ar(A_{21})}]^{q_k} \dots |A_{qn}|^{r_k} [e^{j arg(A_{qn})}]^{s_k} \quad (7)$$

Hablamos de polinomio generalizado porque módulo y fase tienen exponentes diferentes. Nótese que no se utiliza explícitamente el valor en el puerto de entrada (A_{11}/V_{11}), ya que esta será la variable de indexación. Así, al indexar por potencia de entrada se calculará un solo modelo para cada A_{11}/V_{11} , con lo que la influencia de este valor está implícita. Recordemos que el comportamiento del transistor a altas potencias es muy no lineal con la variación de la variable de indexación, y también con la inyección de señales a f_0 y nf_0 .

Los exponentes se suponen conocidos y proceden de un estudio previo basado en la teoría de mezcladores. Pueden ser diferentes para cada dispositivo, para cada punto de polarización y/o para cada rango de potencias. En nuestro caso se cargan desde un fichero. El orden del modelo es el máximo exponente considerado para modelar el transistor como mezclador.

Conocidos los exponentes, calcular el modelo es despejar los coeficientes $L_{ik}(m)$. Aunque el modelo sea no lineal, conocidas las medidas (A's/B's o V's/I's) la ecuación (7) se convierte, para cada valor de indexación, en una ecuación lineal donde los $L_{ik}(m)$'s son las incógnitas. Entonces, este modelo se calcula igual que los anteriores con un sistema lineal super-determinado.

El modelo general filtrado, realmente, es el mismo que el general, sólo que reteniendo los coeficientes relevantes: asumiendo valor cero para los que menos influencia tienen según un cálculo previo.

El modelo avanzado utiliza una ecuación muy similar, pero añadiendo un factor: $|A_{11}|^{t_k} [e^{j arg(A_{11})}]^{u_k}$ que introduce de manera más explícita la influencia de la potencia de entrada. Este modelo se suele calcular de manera separada para cada

armónico, algo que puede hacerse también en los anteriores. El método de cálculo es el mismo de los modelos anteriores.

A partir del modelo avanzado se crea el modelo smoothed re-introduciendo la variable A_{11}/V_{11} . Esto es: multiplicando los coeficientes por $|\bar{A}_{11}|^{t_k} [e^{j \arg(\bar{A}_{11})}]^{u_k}$, siendo $\bar{A}_{11}/\bar{V}_{11}$ la media de A_{11}/V_{11} en el conjunto de datos. Este modelo es el más adecuado para simulaciones indexando por la variable V_{11} .

D. Modelo neuronal

En el modelo neuronal se trata de entrenar un tipo de red capaz de aproximar funciones para actuar como modelo. Los datos de entrada se agrupan por potencia igual que en los modelos de Cardiff. Una red neuronal ordinaria no admite entradas ni salidas complejas por lo que el vector de entrada duplica su tamaño descomponiendo cada fasor como dos variables reales: parte real e imaginaria.

Para un conjunto de medidas con N armónicos (contando DC y fundamental) tendremos $4N$ variables de entrada y otras tantas de salida (dos puertos y dos componentes real e imaginaria para cada valor A o V). El número de variables de salida es igual al de entrada.

La estructura utilizada es un MLP (Multi Layer Perceptron) con un nivel oculto, las funciones de activación son: tangente hiperbólica en el nivel oculto y función identidad en el de salida.

El número de neuronas en el nivel oculto es igual al número de entradas. Se comprobó que se obtenían menores errores con un número mayor pero se descartó para evitar el sobre-entrenamiento (overfitting).

Cada subconjunto de medidas se divide en una parte para entrenamiento (70% de los valores), otra para validación (15%), la validación es el control del progreso del algoritmo durante el proceso de entrenamiento. El 15% restante se deja aparte hasta el final para realizar la prueba de la red. El método de entrenamiento en este caso es la retro-propagación (backpropagation) del tipo Levenberg-Marquardt. Se trata de un método adecuado para una red neuronal aproximativa (fitting network). El nivel de salida se construye de forma progresiva, comenzando con “cero” parámetros y aumentando su número a medida que se entrena. De hecho, en una red final de ejemplo (figura 2), muestra sólo 10 neuronas en el nivel de salida, debido a que dos variables de salida: las partes imaginarias de los valores DC; son siempre nulas.

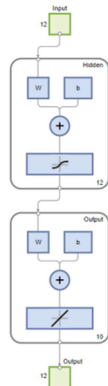


Fig. 2. Estructura de red para un caso concreto.

En la figura 2 vemos la estructura para $N=3$ (DC, fundamental y armónico $2f_0$). El número de entradas y de salidas es $4N=12$ y también el tamaño del nivel oculto. En

estas condiciones el número de pesos de la red es 226. Se ha entrenado con los datos de tres potencias consecutivas, en este caso $3 \times 1026=3078$. El conjunto de entrenamiento se reduce al 70%, esto es: 2155. La relación entre este número de muestras y el de pesos es: $2155/226=9.53$. De este número depende la probabilidad de overfitting, siendo baja en este caso.

III. RESULTADOS Y DISCUSIÓN

A. Las aplicaciones

Como se ha comentado anteriormente, las aplicaciones desarrolladas son app's gráficas en entorno MATLAB (utilizando la herramienta GUIDE para la creación de interfaces gráficas). La aplicación principal (**TotalModel**) tiene este aspecto:

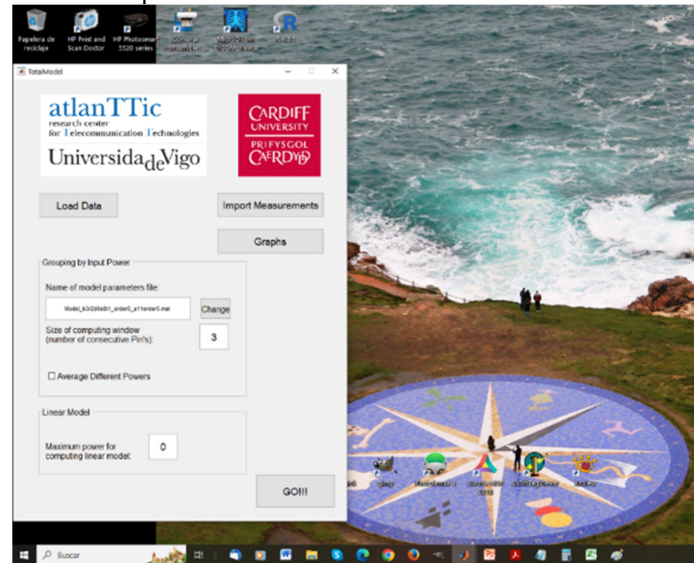


Fig. 3. Ventana principal de TotalModel.

Como se puede deducir de la imagen, la aplicación permite cargar los datos de medidas (desde un fichero **.mat**) y también decidir ciertos parámetros del cálculo: número de potencias que se van a agrupar para el cálculo de modelos no lineales, realización o no de promediado de los datos de las diferentes potencias del intervalo y potencia máxima (dBm) para el cálculo de los modelos lineales básicos.

Al presionar el botón de ejecución (**GO!!!**) se nos pedirá el nombre para el fichero de salida, en dicho fichero se almacenarán en forma de matrices todos los modelos calculados (figura 4).

```

Command Window
>> load Test.mat
>> whos
      Name              Size              Bytes  Class
CARDIFFMODELS          1x9                106448  struct
GENDATA                 1x1                 1498   struct
LINMODELS               1x1026             1420496  struct
PHDMODELS               1x9                 34800   struct

fx >>
    
```

Fig. 4. Resultados de TotalModel.

Como podemos ver en la figura 3, podemos lanzar procesos de importación de medidas y de creación de gráficos. Estos procesos son realmente aplicaciones gráficas independientes que pueden ejecutarse por separado.

La aplicación **ImportApp** está pensada para leer las medidas de los ficheros originales creados por el sistema de medida convirtiéndolos a matrices MATLAB que son grabadas en un fichero de salida que, posteriormente será leído por **TotalModel**.

El botón “**Graphs**” lanza la aplicación **GraphEngine** que permite crear gráficas ejecutando los modelos obtenidos (sólo modelos de Cardiff). Además de los gráficos, la aplicación calcula datos numéricos que son mostrados al usuario y grabados en un fichero de texto.

La aplicación **NeuralModel** es independiente y con una interfaz muy similar a **TotalModel**. Los modelos se calculan para los mismos intervalos de potencias utilizados en los modelos de Cardiff.

De nuevo, los modelos son almacenados en variables que se graban en ficheros. Para cada intervalo de potencias se entrenan dos redes: NETS predice valores de ondas B a partir de las ondas A y NETY predice valores de I’s a partir de V’s.

B. Resultados numéricos, validación de modelos

El método que se juzgó más adecuado para calcular un factor de mérito (FoM: Figure of Merit) para cada modelo es la relación señal a ruido en dB. Para calcularlo se consideran los valores medidos de las variables de salida (I’s o B’s) como señales “deseadas” y los valores de esas mismas variables producidas por un modelo como señales “recibidas”. Además se separa el cálculo por armónicos, para evaluar cada modelo en cada uno de ellos.

TABLA I
VALORES DE SNR MEDIA

Parámetros S			Parámetros Y			Parámetros Z		
f_0			f_0			f_0		
69.11			68.85			68.45		
CARDIFF AB			CARDIFF Y					
DC	f_0	$2f_0$	DC	f_0	$2f_0$			
22.17	25.95	3.72	20.94	20.97	4.15			
CARDIFF FLT AB			CARDIFF FLT Y					
DC	f_0	$2f_0$	DC	f_0	$2f_0$			
22.10	25.88	3.70	20.87	20.89	4.10			
CARDIFF ADV AB			CARDIFF ADV Y					
DC	f_0	$2f_0$	DC	f_0	$2f_0$			
28.82	30.37	3.71	26.91	25.02	4.14			
CARDIFF SMTH AB			CARDIFF SMTH Y					
DC	f_0	$2f_0$	DC	f_0	$2f_0$			
21.64	25.48	3.66	18.57	19.00	3.89			
NEURONAL AB			NEURONAL Y					
DC	f_0	$2f_0$	DC	f_0	$2f_0$			
33.96	44.26	33.61	34.53	36.06	29.01			

En la tabla I, se presentan los resultados obtenidos a partir de las medidas de un transistor real de la foundry Win (<https://winfoundry.com>) de geometría $4 \times 125 \mu\text{m}$, medido a una frecuencia fundamental de 2.4 GHz y con polarización: $V_{ds}=28 \text{ V}$, $I_d = 9 \text{ mA}$ (cerca de clase B).

Los modelos lineales simples son sólo válidos a bajas potencias y para f_0 . Sin embargo, en su intervalo de aplicación los resultados son buenos, prediciendo resultados en toda la carta de Smith. Los modelos Cardiff y neuronal son los más versátiles. Ambos se indexan por potencia de entrada. Los resultados corresponden a modelos Cardiff de orden 5, en la figura 5 se presenta una gráfica obtenida por **GraphEngine** para un modelo Cardiff general (transistor WIN).

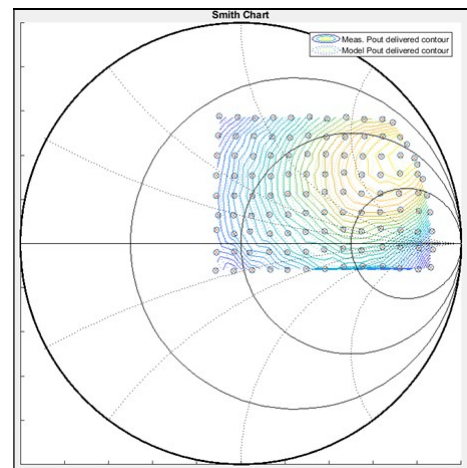


Fig. 5. Líneas continuas → medidas, líneas discontinuas → modelo.

Ante los valores de la tabla I, el modelo neuronal parece muy prometedor. Sin embargo, hay que apuntar que por proceder de Machine Learning, podría producir soluciones físicamente imposibles por lo que, en su implementación, deben incorporarse condiciones de contorno adecuadas. Nótese también como todos los modelos funcionan mejor para el armónico fundamental y peor para el segundo. Para mejorar resultados en $2f_0$ en los modelos Cardiff, el orden debe aumentarse a 7 u 11.

IV. CONCLUSIONES

Se han desarrollado herramientas MATLAB de fácil utilización que permiten el cálculo de varios tipos de modelos comportamentales FET’s de microondas. Los valores obtenidos pueden utilizarse para generación de gráficas pero también para ser incorporados a simuladores compatibles donde se usarán en diseño de circuitos.

AGRADECIMIENTOS

Este trabajo ha contado con la colaboración del Ministerio de Ciencia e Innovación (proyecto PID2020-116569RB-C33/AEI/10.13039/501100011033) y de la Universidade de Vigo (ayudas para la movilidad de investigadores). Los autores también agradecen a “WIN Semiconductors” por las muestras facilitadas.

REFERENCIAS

- [1] Steve Dudkiewicz et al, “Comparing Nonlinear Vector Network Analyzer Methodologies”, Microwave Journal, 2015.
- [2] <https://es.mathworks.com/products/matlab.html>.
- [3] <https://es.mathworks.com/discovery/matlab-gui.html>.
- [4] D.M. Pozar, Microwave engineering, Addison-Wesley, 1993.
- [5] A. Al-Rawachy et al, "Cardiff Behavioural Model Analysis using a Two-Tone Stimulus," 2019 IEEE PAWR, Orlando, FL, USA, 2019, pp. 1-4, doi: 10.1109/PAWR.2019.8708726.
- [6] M. R. Moure et al, "Direct extraction of an admittance domain behavioral model from large-signal load-pull measurements", 2017 IEEE MTT-S IMS, Honolulu, HI, USA, 2017, pp. 1057-1060, doi: 10.1109/MWSYM.2017.8058775.
- [7] M. Tian et al, "An Extraction Method Based on Artificial Neural Network Techniques for Novel Cardiff Model With Reasonable Extrapolation Behavior," in IEEE Microwave and Wireless Technology Letters, vol. 34, no. 1, pp. 5-8, Jan. 2024, doi: 10.1109/LMWT.2023.3329979.
- [8] J. Burgos, Curso de Álgebra y Geometría, Alhambra, 1994.