

On the use of artificial neural networks for the automated high-level design of Sigma-Delta Modulators

Francisco Javier Gómez Pulido

29 enero 2025

Resumen

El diseño de alto nivel de moduladores Sigma-Delta es un proceso complejo que involucra la selección de arquitectura y la optimización de múltiples parámetros de diseño para cumplir con especificaciones dadas. En este trabajo, se propone el uso de redes neuronales artificiales (ANNs) para automatizar este proceso, reduciendo significativamente el tiempo de cómputo sin comprometer la calidad del diseño.

El método se divide en dos fases: una fase de entrenamiento, en la que se generan grandes volúmenes de datos mediante simulaciones de comportamiento y se entrenan modelos de redes neuronales recurrentes (RNNs) para predecir los parámetros de diseño óptimos, y una fase de inferencia, en la que se utiliza un clasificador para seleccionar la arquitectura más adecuada y una RNN entrenada para predecir los valores de diseño. Un ajuste fino posterior mejora aún más los resultados utilizando simulaciones adicionales.

Los resultados obtenidos demuestran que la metodología basada en ANNs es capaz de reducir el tiempo de optimización en órdenes de magnitud en comparación con métodos tradicionales, al tiempo que logra diseños con una Figura de Mérito (FOM) competitiva. Además, se ha mostrado que este enfoque supera a métodos basados en búsqueda directa en tablas de datos (LUT), gracias a la capacidad de interpolación de las redes neuronales.

Índice

1	Introducción	3
2	Antecedentes y estado del arte en la síntesis basada en optimización de moduladores $\Sigma\Delta$	3
3	Metodología propuesta	4
3.1	Definición del problema	4
3.2	Solución propuesta	4
3.3	Diseño del dataset	6
3.4	Consideraciones sobre la definición del clasificador	6
3.5	Búsqueda de arquitectura de red y optimización del modelo	7
3.6	Validación cruzada y mejora de resultados	8
4	Caso de estudio	8
4.1	Entrenamiento y validación de las redes neuronales	9
5	Discusión	9

1. Introducción

Los moduladores Sigma-Delta son una de las técnicas más efectivas para implementar convertidores analógico-digitales (ADCs) en una amplia variedad de aplicaciones, que incluyen instrumentación, dispositivos biomédicos, sensores automotrices y comunicaciones. Aunque ofrecen ventajas significativas en términos de robustez y eficiencia, el diseño de $\Sigma\Delta$ Ms de alto rendimiento es complejo y requiere experiencia en un flujo de diseño jerárquico descendente/ascendente que abarca desde el sistema hasta el chip.

La principal dificultad en el diseño radica en encontrar la arquitectura adecuada para un conjunto dado de especificaciones y en mapear estas especificaciones a parámetros eléctricos a nivel de circuito. En los últimos años, se han desarrollado metodologías de diseño y herramientas de automatización (EDA) para optimizar tareas a nivel de sistema, como la selección de arquitectura, el diseño de filtros de lazo, la modelización conductual y el dimensionamiento. La aproximación más común se basa en métodos de síntesis optimizados, donde un optimizador trabaja con un simulador para encontrar el diseño óptimo.

Recientemente, los algoritmos de inteligencia artificial, como las redes neuronales artificiales, han comenzado a desempeñar un papel clave en la automatización del diseño de circuitos analógicos, permitiendo reemplazar simuladores o actuar como motores de optimización. Estas redes pueden automatizar el dimensionamiento de sistemas para especificaciones nuevas, reduciendo significativamente el tiempo y los recursos computacionales necesarios.

Este artículo presenta una metodología basada en ANNs para el diseño automatizado a nivel superior de $\Sigma\Delta$ Ms, que combina clasificación de arquitecturas y regresión para inferir variables de diseño. La metodología es válida para arquitecturas de lazo simple y en cascada, con cuantización de uno o múltiples bits, y técnicas de circuitos de tiempo continuo o discreto. Los resultados demuestran mejoras significativas en términos de tiempo de CPU y rendimiento en comparación con otros métodos de optimización.

2. Antecedentes y estado del arte en la síntesis basada en optimización de moduladores $\Sigma\Delta$

La metodología de diseño de los moduladores $\Sigma\Delta$ sigue un enfoque jerárquico de arriba hacia abajo y de abajo hacia arriba. En esta estrategia de divide y vencerás, un $\Sigma\Delta$ modulador se divide en varios niveles de abstracción (nivel de sistema, nivel de bloques de construcción, nivel de circuitos y nivel de dispositivos/físico), de manera que en cada nivel se lleva a cabo un proceso de diseño para asignar las especificaciones del sistema de manera jerárquica.

El proceso de diseño de los $\Sigma\Delta$ moduladores implica la selección de la mejor arquitectura que cumpla con las especificaciones de ENOB y ancho de banda con el menor consumo de energía posible. Para ello, se utilizan ecuaciones ideales para estimar parámetros clave como la Relación de Sobre-muestreo (OSR), el orden del filtro de lazo (L) y el número de bits del cuantificador (B). Posteriormente, se aplican modelos más precisos mediante herramientas como la $\Sigma\Delta$ Toolbox de MATLAB.

Una vez definida la arquitectura del modulador, se analizan las no idealidades del circuito para especificar los bloques de construcción principales, como amplificadores y comparadores. Para ello, se emplean simuladores precisos y eficientes como SIMSIDES, un simulador basado en MATLAB/SIMULINK que permite modelar de manera rápida y precisa los efectos de las no idealidades del circuito.

Para optimizar el diseño, se utilizan técnicas como el modelado de comportamiento y el método lifting, este último orientado a los $\Sigma\Delta$ moduladores en tiempo continuo (CT- $\Sigma\Delta$). Herramientas como Sigma-Delta Synthesis Tool automatizan el diseño, permitiendo la optimización de parámetros del filtro de lazo y la Función de Transferencia de Señal (STF).

El diseño automatizado de los $\Sigma\Delta$ moduladores se apoya en motores de optimización que exploran el espacio de diseño multidimensional. Sin embargo, este trabajo propone sustituir estos sistemas por redes neuronales artificiales (ANNs), las cuales aprenden a mapear especificaciones en parámetros de diseño a partir de conjuntos de datos extensos. Aunque este artículo no aborda el uso de ANNs para el dimensionamiento de coeficientes de filtros de lazo, esta aplicación podría explorarse en trabajos futuros.

3. Metodología propuesta

3.1. Definición del problema

El diseño de alto nivel de un modulador $\Sigma\Delta$ requiere resolver dos problemas clave:

- **Selección de arquitectura:** Se debe elegir una topología adecuada dentro de un conjunto de alternativas $\{A_j\}$, considerando especificaciones como ENOB (número efectivo de bits, resolución real de un ADC), BW (ancho de banda, rango de frecuencias de la señal de entrada que el sistema puede procesar sin degradación significativa), SNR (relación señal-ruido), SNDR (relación señal-ruido y distorsión), THD (distorsión armónica total, cantidad de energía de las armónicas no deseadas en una señal) y FOM (figura de mérito, factor que sirve para comparar diseños), además de restricciones adicionales impuestas por el sistema.
- **Estimación de variables de diseño:** Para la arquitectura seleccionada, se deben determinar las variables de diseño $\bar{\epsilon}(A_j)$ que optimicen las especificaciones dadas.

3.2. Solución propuesta

Para abordar el problema de selección de arquitectura, se utiliza un clasificador entrenado C que asigna una topología A_j en función de las métricas del sistema $\bar{0}_i$. Posteriormente, una red neuronal de tipo regresión (RNN) infiere las variables de diseño $\bar{\epsilon}_k$ correspondientes:

$$A_j = C(\bar{0}_i); \quad \bar{\epsilon}_k = RNN(\bar{0}_i, A_j) \quad (1)$$

El uso de clasificadores y redes neuronales preentrenadas permite reducir significativamente el tiempo de inferencia, logrando estimaciones en menos de 50 ms. No obstante,

la calidad de los resultados depende directamente de la precisión del clasificador y la red neuronal, así como de la calidad del conjunto de datos empleado para su entrenamiento.

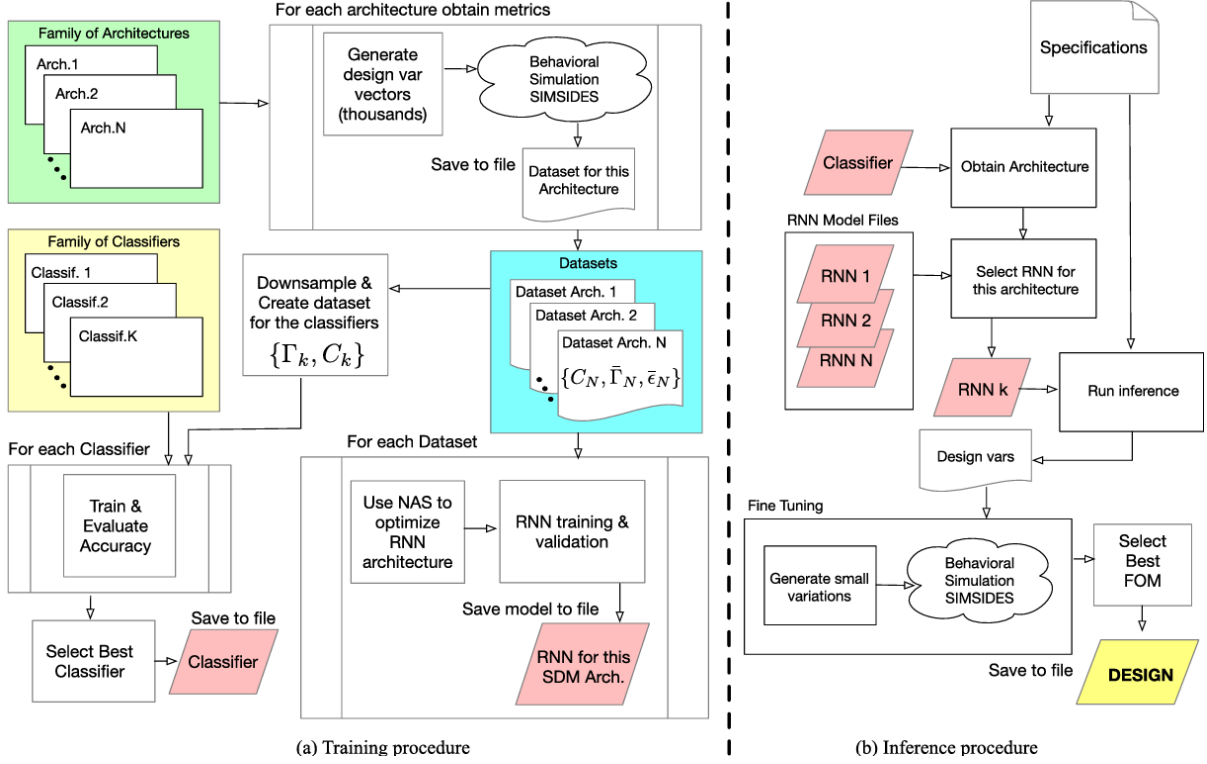


Figura 1: Flujo completo de la metodología. Se divide el proceso en dos grandes fases: entrenamiento e inferencia.

Procedimiento de entrenamiento: el objetivo es entrenar modelos capaces de predecir los parámetros de diseño de los moduladores $\Sigma\Delta$.

- **Selección de arquitecturas:** se parte de un conjunto de posibles arquitecturas, se generan miles de vectores de variables, se evalúan para obtener métricas de rendimiento y se almacenan los resultados en conjuntos de datos organizados por arquitectura.
- **Entrenamiento de clasificadores:** se construye una base de datos reducida a partir de las simulaciones y se usa para entrenar múltiples clasificadores. Se selecciona el mejor modelo, el cual se usará posteriormente en la fase de inferencia para seleccionar la arquitectura adecuada según las especificaciones de diseño.
- **Entrenamiento de redes neuronales recurrentes (RNNs):** para cada conjunto de datos, se optimiza la arquitectura de la RNN utilizando NAS (Neural Architecture Search). Se entrenan y validan las RNNs con los datos simulados y se guarda una RNN entrenada para cada arquitectura de modulador $\Sigma\Delta$.

Procedimiento de inferencia: se aplica el sistema entrenado para generar un diseño óptimo basado en especificaciones dadas.

- **Selección de arquitectura:** se reciben las especificaciones del diseño deseado y el clasificador predice la arquitectura de $\Sigma\Delta$ más adecuada.

- **Inferencia con redes neuronales:** se selecciona la RNN correspondiente a la arquitectura elegida y se ejecuta la inferencia en la RNN para obtener los valores de las variables de diseño.
- **Ajuste fino y selección del mejor diseño:** se generan pequeñas variaciones en los parámetros obtenidos y se ejecutan simulaciones para refinar los resultados. Se selecciona la mejor solución basada en FOM y el diseño final se guarda como salida del proceso.

3.3. Diseño del dataset

Nuestra propuesta es estructurar cada entrada del conjunto de datos como un triplete de la forma $\{C_i, \bar{0}_i, \bar{\epsilon}_i\}$, donde:

- C_i es una variable categórica que define la arquitectura del modulador.
- $\bar{0}_i$ es un vector de métricas de rendimiento del sistema, es decir, un punto en el espacio de especificaciones.
- $\bar{\epsilon}_i$ es un vector de variables de diseño, un punto en el espacio de variables de diseño, obtenido a partir de simulaciones de comportamiento con SIMSIDES.

El vector de especificaciones $\bar{0}_i$ debe ser común a todas las arquitecturas del conjunto de datos para que el clasificador pueda operar sobre él. Sin embargo, las variables de diseño $\bar{\epsilon}_i$ varían según la arquitectura. Para abordar esto, existen dos enfoques:

1. Un único conjunto de datos con todas las arquitecturas, utilizando un vector de diseño unificado que incluya todas las variables necesarias, marcando como N/A aquellas que no aplican a una arquitectura específica.
2. Conjuntos de datos separados para cada arquitectura, optimizando cada red neuronal de regresión de manera independiente para su arquitectura correspondiente.

El segundo enfoque es más escalable y eficiente, ya que permite entrenar redes neuronales específicas para cada arquitectura sin necesidad de reformular todo el conjunto de datos al añadir nuevas topologías.

3.4. Consideraciones sobre la definición del clasificador

La selección de una arquitectura de clasificador adecuada es crucial para el éxito del modelo. No existe una técnica única óptima, ya que la mejor opción depende de la naturaleza del conjunto de datos y las propiedades estadísticas de la información. Factores como el desequilibrio de clases, la correlación entre características, la dimensionalidad y el ruido influyen en esta decisión.

Para evitar problemas de sesgo en la clasificación, utilizamos un subconjunto equilibrado del conjunto de datos, asegurando una representación equitativa de todas las clases. Implementamos un procedimiento que explora diversas técnicas de clasificación, seleccionando la mejor opción según la matriz de confusión obtenida. Entre los clasificadores considerados se incluyen:

- Análisis discriminante lineal y cuadrático.
- Máquinas de soporte vectorial (SVM) con diferentes núcleos.
- Naive bayes gaussiano y multinomial.
- Árboles de decisión y bosques aleatorios.
- Gradient boosting classifiers.
- Redes neuronales.

Este enfoque permite seleccionar el mejor clasificador en función de los datos específicos de entrenamiento y validación, maximizando la precisión de la predicción de la arquitectura óptima.

3.5. Búsqueda de arquitectura de red y optimización del modelo

La base de nuestra solución consiste en utilizar una red neuronal de regresión multivariable optimizada para cada arquitectura de modulador $\Sigma\Delta$. Esta red mapea las métricas del sistema (especificaciones) en variables de diseño de alto nivel, desempeñando el papel del optimizador en el enfoque convencional.

Sin embargo, definir manualmente una arquitectura de red que minimice el error de regresión es un proceso tedioso y demandante en tiempo. Para superar esto, hemos integrado técnicas de Búsqueda de Arquitectura de Redes (NAS, por sus siglas en inglés) en nuestro marco de diseño. NAS permite explorar automáticamente el espacio de hiperparámetros y encontrar una arquitectura de red óptima utilizando la API Keras Tuner.

Nuestro enfoque primero define una plantilla de red parametrizada y un espacio de búsqueda para los siguientes hiperparámetros:

- Número de neuronas por capa: [32, 64] con incrementos de 4.
- Número de capas adicionales después de la concatenación: [1, 6] con incrementos de 1.
- Función de activación: ReLU o tangente hiperbólica (tanh).
- Capas de dropout: habilitadas o deshabilitadas.
- Algoritmo de optimización: Adam, SGD, RMSprop, Adadelta.

El algoritmo de búsqueda se ejecuta hasta que se agotan las combinaciones del espacio de búsqueda o se cumple una condición de parada anticipada. Posteriormente, la mejor arquitectura encontrada se ajusta con un conjunto de entrenamiento y validación en una división 80 %-20 %.

3.6. Validación cruzada y mejora de resultados

El error cuadrático medio (MSE) se usa para cuantificar la precisión de las redes neuronales. Sin embargo, debido a la naturaleza no lineal de los $\Sigma\Delta$ moduladores, este error no proporciona una visión clara sobre la fidelidad de los parámetros de diseño obtenidos respecto a las especificaciones requeridas. Para abordar este problema, integramos una fase adicional de validación cruzada y refinamiento de resultados.

Esta fase consiste en generar un pequeño número de variaciones aleatorias de la solución inicial proporcionada por la red, con ligeros ajustes en sus componentes:

$$DV_{n,a} = DV_n(1 + \alpha_{n,a}), \quad \alpha_{n,a} \sim U(-r, r) \quad (2)$$

con r representando un porcentaje de ajuste y U una distribución uniforme. Se evalúan estas variaciones en un simulador de comportamiento para seleccionar la solución con la mejor Figura de Mérito (FOM).

Este enfoque combina la velocidad de inferencia de redes neuronales con la precisión de la simulación, permitiendo encontrar configuraciones de diseño con mejor desempeño sin un costo computacional elevado.

4. Caso de estudio

Las cuatro arquitecturas de moduladores $\Sigma\Delta$ seleccionadas como casos de estudio son:

1. Modulador SC de 2º orden.
2. Modulador SC de cascada 2-1 de 3º orden.
3. Modulador SC de cascada 2-1-1 de 4º orden con cuantización de 3 bits.
4. Modulador Gm-C de 2º orden con cuantización de 3 niveles.

Para entrenar las redes neuronales se generó un conjunto de datos mediante simulaciones en SIMSIDES. Se consideraron cinco valores del factor de sobremuestreo ($OSR = 32, 64, 128, 256, 512$), y se exploraron los espacios de variables de diseño para cada arquitectura:

- En el caso de los moduladores 2º orden Gm-C y 3º orden 2-1 SC, las variables de diseño fueron generadas mediante bucles anidados (barrido sistemático).
- Para los otros dos casos (4º orden 2-1-1 SC y 2º orden SC), las variables de diseño fueron generadas aleatoriamente dentro de rangos predeterminados.

El proceso de generación de datos incluyó más de 200,000 simulaciones. Sin embargo, debido a la aleatoriedad en la exploración de variables de diseño, no todas las simulaciones generaron resultados útiles. Por lo tanto, se filtraron los diseños con una relación señal-ruido (SNR) inferior a 50 dB, reduciendo el conjunto de datos final a cerca de 120,000 puntos de datos. Cada entrada del conjunto de datos fue estructurada como un triplete:

$$\{C_i, \Theta_i, \epsilon_i\} \quad (3)$$

donde:

- C_i : arquitectura del modelador (variable categórica).
- Θ_i : valor de especificaciones (SNR, OSR, consumo de potencia).
- ϵ_i : vector de variables de diseño obtenidas en la simulación.

4.1. Entrenamiento y validación de las redes neuronales

El entrenamiento de las ANNs tuvo dos fases:

1. **Clasificación de arquitecturas:** se utilizó una red para inferir la arquitectura del modulador más adecuada a partir de un conjunto de especificaciones.
2. **Regresión para variables de diseño:** se entrenó una ANN para cada arquitectura, con el objetivo de predecir las variables de diseño correspondientes a un conjunto dado de especificaciones.

Entrenamiento del clasificador: para evitar problemas de desbalance de clases, se equilibró el conjunto de datos para que cada arquitectura tuviera la misma cantidad de entradas. Se evaluaron los distintos enfoques de clasificación mencionados anteriormente. El clasificador que obtuvo el mejor rendimiento fue el **Gradient Boosting Classifier**, con la mejor matriz de confusión.

Entrenamiento de las redes de regresión: para cada arquitectura se diseñó una red neuronal explorando diferentes hiperparámetros. Una vez determinada la mejor configuración para cada red, se llevó a cabo el entrenamiento (80 % conjunto de entrenamiento y 20 % conjunto de validación). La métrica utilizada fue el MSE.

Validación y ajuste final: se seleccionaron aleatoriamente 1000 pares de datos de validación y se calculó el error relativo en las especificaciones obtenidas frente a las deseadas. Se observó que las redes neuronales lograron una desviación mínima en la predicción de variables de diseño, asegurando que las especificaciones requeridas fueran alcanzadas con alta precisión. Para mejorar aún más los resultados, se implementó un proceso de búsqueda local alrededor de la solución obtenida, basado en la simulación de pequeñas variaciones aleatorias en las variables de diseño y la selección de la que producía el mejor Figura de Mérito de Schreier (FOM).

5. Discusión

Se comparó el rendimiento de esta metodología con tres optimizadores tradicionales: algoritmos genéticos, descenso del gradiente, positive basis Np1.

- Eficiencia en tiempo de cómputo: la solución basada en ANNs fue 60 veces más rápida.
- Calidad de las soluciones: aunque las ANNs no tienen capacidad de optimización directa, el proceso de ajuste fino con búsqueda local permitió obtener soluciones con menor consumo de potencia y mejor Figura de Mérito de Schreier (FOM) en la mayoría de los casos.