

**Materia:**

**DISEÑO ELECTRÓNICO BASADO EN  
SISTEMAS EMBEBIDOS**

**Alumno:**

Posadas Pérez Isaac Sayeg

Paniagua Rico Juan Julian

García Azzúa Jorge Roberto

**Grado y grupo:**

8°G

**Profesor:**

Garcia Ruiz Alejandro Humberto

**Unidad 3 - Tarea 2:**

Óptimo local

# Óptimo local en sistemas embebidos

## Introducción

En el desarrollo y diseño de sistemas embebidos, el concepto de optimización es fundamental para alcanzar un balance adecuado entre rendimiento, consumo energético, tamaño del sistema y coste de producción. Cuando se abordan estos objetivos de forma aislada, centrándose en mejorar un componente o subsistema específico sin tener en cuenta las repercusiones sobre el resto del sistema, se está trabajando con una estrategia de optimización local. El óptimo local es un concepto que proviene del campo de la optimización matemática, y se refiere a una solución que es mejor que otras cercanas, pero que no necesariamente es la mejor posible dentro del conjunto completo de soluciones.

En sistemas embebidos, esto significa que se puede lograr un mejor desempeño en una parte del sistema (como reducir el consumo energético de un módulo específico) sin necesariamente mejorar el comportamiento global. Comprender el papel que juega el óptimo local en este contexto es esencial, ya que en muchas ocasiones, por limitaciones prácticas o técnicas, es preferible o incluso necesario enfocarse en soluciones locales, especialmente en etapas tempranas del diseño o en sistemas muy especializados.

## Desarrollo

Los sistemas embebidos suelen ser sistemas altamente especializados que están diseñados para realizar tareas específicas con limitaciones de recursos muy estrictas. La optimización local se convierte en una estrategia atractiva en estos contextos debido a que permite realizar ajustes finos en partes individuales del sistema sin incurrir en la complejidad computacional o de diseño que implica una optimización global.

Por ejemplo, al diseñar el firmware de un microcontrolador, se puede identificar un cuello de botella en el consumo de energía asociado al uso del periférico de

comunicaciones. Al aplicar técnicas de optimización local, como reducir la frecuencia de operación del periférico o usar modos de ahorro energético, se puede disminuir significativamente el consumo sin necesidad de alterar otras partes del sistema. Esto es útil en sistemas donde el coste de rediseñar todo el sistema es alto, o donde las modificaciones deben aplicarse de manera rápida.

Sin embargo, el enfoque de óptimo local también tiene sus riesgos. Es posible que una mejora local, como aumentar la velocidad de una operación crítica, provoque un aumento en la temperatura del sistema o en el consumo total de energía, lo cual afectaría negativamente otras partes del sistema o incluso comprometería la estabilidad general. Por esta razón, la optimización local debe aplicarse con cuidado, idealmente con una visión de conjunto que permita identificar posibles conflictos o efectos secundarios.

Desde un punto de vista algorítmico, encontrar un óptimo local suele ser más rápido y menos costoso que encontrar un óptimo global. Algoritmos como el descenso por gradiente o los métodos heurísticos suelen detenerse en mínimos o máximos locales, lo que en muchos casos es suficiente para sistemas embebidos que operan bajo configuraciones fijas o patrones de uso predecibles. Además, en contextos industriales donde el tiempo de desarrollo es limitado y los recursos son escasos, las optimizaciones locales permiten un desarrollo ágil, iterativo y centrado en problemas específicos de alto impacto.

## Ejemplos

- Control de energía en microcontroladores: En un sistema embebido simple, como un reloj inteligente, reducir la frecuencia del reloj del microcontrolador durante períodos de inactividad puede considerarse una optimización local. Esto se logra sin modificar el resto del diseño, pero contribuye a una mayor duración de la batería.
- Compresión de datos en sensores remotos: Aplicar algoritmos de compresión más eficientes en sensores individuales permite reducir el volumen de datos transmitidos, disminuyendo el uso del módulo de comunicaciones. Aunque es

una mejora localizada, puede tener un impacto significativo en el rendimiento energético.

- Optimización de funciones críticas: En algunos sistemas, se identifican funciones específicas que consumen demasiado tiempo de CPU. Reescribir solo esas funciones en lenguaje ensamblador optimizado puede mejorar el rendimiento general de forma localizada.
- Reducción de uso de memoria RAM: Reducir el tamaño de las estructuras de datos usadas por un módulo específico de la aplicación permite liberar memoria para otras tareas, sin modificar el resto del sistema.
- Ajustes de algoritmos de control: En un sistema de control embebido para drones, ajustar los parámetros del controlador PID para una mejor estabilidad puede hacerse de forma local, sin cambiar la arquitectura global del sistema.

## Conclusión

La búsqueda de óptimos locales en el diseño de sistemas embebidos es una estrategia valiosa y, en muchos casos, necesaria. Permite realizar mejoras concretas y eficientes con un costo bajo en términos de tiempo y recursos, y es especialmente útil cuando se trabaja en sistemas muy segmentados o con componentes fuertemente desacoplados.

Sin embargo, es importante reconocer que una mejora local no siempre se traduce en una mejora del sistema en su conjunto. De hecho, en algunas situaciones, puede provocar efectos negativos en otros subsistemas, por lo que se requiere una evaluación cuidadosa antes de aplicar soluciones locales.

En conclusión, el óptimo local representa una herramienta poderosa en el arsenal del diseñador de sistemas embebidos. Su correcta aplicación puede marcar la diferencia entre un sistema funcional y uno altamente eficiente, siempre que se maneje con criterio y visión de conjunto.

## Bibliografía

- ❖ S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi (1983). *Optimization by Simulated Annealing*. Science.

- ❖ Marwedel, P. (2011). *Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems*. Springer.
- ❖ Buttazzo, G. C. (2011). *Hard Real-Time Computing Systems*. Springer.
- ❖ Wolf, W. (2012). *Computers as Components: Principles of Embedded Computing System Design*. Morgan Kaufmann.
- ❖ Zitzler, E., Laumanns, M., & Thiele, L. (2001). *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*.
- ❖ Kuo, T.-W., & Li, C.-L. (2003). *Optimizing energy consumption for real-time systems with limited resources*. *Journal of Systems Architecture*.
- ❖ Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley.
- ❖ Givargis, T., & Vahid, F. (2002). *Platune: A Tuning Framework for System-on-a-Chip Platforms*. *IEEE Transactions on CAD*.