

Implementación de una derivada digital para el control de un robot seguidor de línea

Carlos Enrique Maciel García
carlos.mg@cdguzman.tecnm.mx
Instituto Tecnológico de Ciudad Guzmán, México

Rafael Pantoja González
rafael.pg@cdguzman.tecnm.mx
Instituto Tecnológico de Ciudad Guzmán, México

Luis Enrique Salvador Cano
luis.sc@cdguzman.tecnm.mx
Instituto Tecnológico de Ciudad Guzmán, México

Favio Rey Lúa Madrigal
favio.lm@cdguzman.tecnm.mx
Instituto Tecnológico de Ciudad Guzmán, México

Resumen

Este artículo presenta el desarrollo e implementación de una función derivada discreta aplicada a la señal obtenida de una cámara lineal de 128 píxeles para el control de un robot seguidor de línea. El sistema fue construido sobre una plataforma con microcontrolador Freescale, programado en el entorno CodeWarrior. El objetivo fue mejorar la capacidad del robot para identificar con precisión los bordes de la línea, permitiéndole seguir trayectorias curvas de manera eficiente. Durante la adquisición de datos, se identificó que los primeros 15 y los últimos 8 píxeles de la cámara no presentaban un comportamiento lineal, por lo que se descartaron del procesamiento. La derivada se implementó como una simple diferencia de intensidades consecutivas, asumiendo un tiempo de muestreo constante de 18 ms. Se analizaron diferentes estrategias para mejorar la detección de bordes, incluyendo la selección de pares de muestras óptimas para la derivación. Los resultados obtenidos, tanto en gráficas de comportamiento como en pruebas en pista, demostraron que el uso de la derivada mejoró notablemente la detección de cambios bruscos de intensidad, correlacionados con los bordes de la línea. Esta mejora en el control permitió al robot alcanzar el primer lugar en una competencia nacional durante dos años consecutivos. La propuesta es replicable y adaptable a contextos educativos, donde se busca integrar procesamiento de señales, microcontroladores y robótica móvil.

Palabras clave: Robot seguidor de línea, Derivada discreta, Cámara lineal, Freescale

Abstract

This article presents the development and implementation of a discrete derivative function applied to the signal obtained from a 128-pixel linear camera for the control of a line-following robot. The system was built on a platform using a Freescale microcontroller, programmed within the CodeWarrior environment. The objective was to enhance the robot's ability to accurately identify line edges, enabling it to follow curved trajectories efficiently. During data acquisition, it was determined that the first 15 and the last 8 pixels of the camera exhibited non-linear behavior and were thus excluded from processing. The derivative was implemented as a simple difference between consecutive intensity values, assuming a constant sampling time of 18 ms. Several strategies were analyzed to improve edge detection, including the selection of optimal sample pairs for differentiation. The results, both from behavioral graphs and track tests, demonstrated that the use of the derivative significantly improved the detection of abrupt intensity changes, corresponding to the edges of the line. This enhancement in control enabled the robot to achieve first place in a national competition for two consecutive years. The proposed solution is replicable and adaptable to educational contexts, where signal processing, microcontrollers, and mobile robotics are integrated into learning environments.

Keywords: Line follower robot, Discrete derivative, Linear camera, Freescale

Introducción

En el ámbito de la robótica educativa, los robots seguidores de línea se han consolidado como plataformas ideales para el desarrollo de competencias en programación, control y procesamiento de señales. Su construcción implica la integración de sensores, microcontroladores y algoritmos que permiten la interpretación del entorno y la toma de decisiones en tiempo real. Tradicionalmente, estos robots emplean sensores infrarrojos para detectar contrastes de color entre una línea negra y un fondo blanco; sin embargo, este enfoque presenta limitaciones en cuanto a resolución, velocidad de respuesta y adaptabilidad a curvas pronunciadas o condiciones de iluminación variables.

Con el propósito de superar estas limitaciones, se diseñó un robot seguidor de línea basado en una cámara lineal de 128 píxeles y una tarjeta Freescale Freedom FRDM-KL25Z, programada en el entorno CodeWarrior. La incorporación de la cámara permitió obtener una representación más detallada del entorno, posibilitando la aplicación de técnicas de procesamiento digital de señales, en particular una derivada discreta de la señal capturada, para la detección precisa de los bordes de la línea.

El desarrollo de esta solución surgió en el marco de un contexto educativo, como parte de una estrategia formativa orientada al aprendizaje activo de conceptos de control digital, matemáticas aplicadas y programación embebida. Como resultado, el robot obtuvo el primer lugar en una competencia nacional durante dos años consecutivos, validando tanto el diseño técnico como su efectividad operativa.

Este artículo presenta el proceso de diseño, implementación y validación de dicha estrategia, haciendo énfasis en la construcción de la derivada discreta y su impacto en la eficiencia del seguimiento de línea.

Referente teórico

El uso de algoritmos de control en robótica móvil permite mejorar el desempeño de los sistemas de navegación autónoma. Entre los más utilizados se encuentra el control proporcional-derivativo (PD), el cual actúa sobre el error de posición y su derivada para anticiparse a los cambios en la trayectoria. La derivada, en particular, permite detectar variaciones rápidas, siendo útil en escenarios donde el robot debe responder ante curvas cerradas o cambios bruscos en la línea guía. En aplicaciones reales, la derivada debe ser

adaptada a versiones discretas que consideren la frecuencia de muestreo del sistema (Ogata, 2010).

En lugar de sensores infrarrojos, que ofrecen una lectura puntual del entorno, el uso de cámaras lineales permite capturar información con mayor resolución espacial. Las cámaras lineales son dispositivos que registran una línea continua de píxeles, lo que permite reconstruir perfiles de intensidad en tiempo real. Esta información puede procesarse mediante técnicas de diferenciación discreta para identificar bordes, contrastes y centros de masa ópticos. En trabajos como los de Elmenreich y Klingler (2006), se ha demostrado que la implementación de sensores ópticos avanzados incrementa la precisión en robots móviles, especialmente cuando se combina con procesamiento embebido eficiente.

La derivada discreta aplicada a una señal digital, como la obtenida desde una cámara lineal, consiste en calcular la diferencia entre dos muestras adyacentes del arreglo de datos. Esto permite acentuar los puntos donde existen cambios abruptos de intensidad, es decir, donde se encuentran los bordes de la línea. Matemáticamente, si $I[n]$ representa la intensidad del píxel en la posición n , entonces su derivada discreta puede aproximarse por:

$$D[n] = I[n] - I[n-1]$$

Este método, aunque simple, ha demostrado ser eficaz para detectar con rapidez y precisión los puntos clave del trayecto. Para optimizar el resultado, se evaluaron diferentes estrategias, como ignorar zonas con comportamiento no lineal en los extremos del arreglo y ajustar dinámicamente la pareja de muestras utilizadas. Estas mejoras permitieron generar señales derivadas más limpias, con picos claramente definidos en los bordes, lo que facilitó su interpretación por parte del sistema de control.

Además, la derivada de una señal es ampliamente utilizada en el procesamiento digital de imágenes para detectar bordes, ya que estos se manifiestan como cambios bruscos en los niveles de intensidad. Como se observa en la Figura 1, una transición escalonada en una función de intensidad $f(x)$ genera un pico en la derivada $f'(x)$, que marca con precisión el punto de cambio. Esta propiedad es esencial para determinar con exactitud los límites de una línea al analizar el perfil de una imagen lineal (Gonzalez & Woods, 2008).

En este proyecto se utilizó la tarjeta Freescale Freedom FRDM-KL25Z, que integra un microcontrolador ARM Cortex-M0+, con capacidad de procesamiento de hasta 48 MHz, múltiples canales ADC y periféricos de comunicación serie. Su

programación se realizó en el entorno CodeWarrior, ampliamente utilizado en el ámbito académico por su compatibilidad con hardware Freescale y sus herramientas integradas de depuración. Estas características permitieron capturar, procesar e interpretar los datos de la cámara lineal a una frecuencia aproximada de 55 Hz, habilitando la aplicación del algoritmo derivativo propuesto.

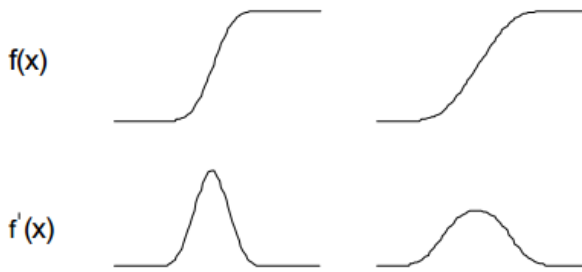


Figura 1. Usando la derivada para detección de bordes

Metodología

El proyecto se desarrolló bajo un enfoque experimental, utilizando una arquitectura embebida basada en la tarjeta Freescale Freedom FRDM-KL25Z. Esta tarjeta incorpora un microcontrolador ARM Cortex-M0+ de 32 bits, con capacidades suficientes para el procesamiento en tiempo real de señales provenientes de sensores ópticos. La programación se realizó utilizando el entorno de desarrollo CodeWarrior, el cual permitió la configuración de los periféricos necesarios para la adquisición de datos, el procesamiento digital y la generación de señales de control.

Hardware del sistema:

- Microcontrolador: Freescale FRDM-KL25Z
- Sensor principal: cámara lineal de 128 píxeles
- Actuadores: dos motores DC controlados mediante puente H
- Fuente de energía: batería recargable de polímero de litio (LiPo)
- Chasis: diseño propio, ligero, compacto y estable
- Sistema de adquisición: interfaz serial entre cámara y microcontrolador

La cámara lineal fue configurada para operar a una frecuencia de aproximadamente 55 Hz (18 ms por captura), proporcionando un arreglo de 128 valores enteros que representan niveles de intensidad de gris.

Tratamiento de la señal: Antes de aplicar la derivada, se determinó que los primeros 15 y los últimos 8 píxeles del

arreglo presentaban un comportamiento no lineal. Por lo tanto, estos fueron descartados, utilizando únicamente la zona efectiva central (píxeles 16 a 120) para el análisis.

Para observar el comportamiento real de la señal capturada por la cámara, se realizó una prueba inicial enviando los datos del arreglo de píxeles a través de una terminal serial virtual (Tera Term) a una velocidad de 115200 baudios. Estos datos fueron copiados directamente a una hoja de cálculo en Excel para su análisis visual. En la Figura 2, se puede observar la señal obtenida por la cámara, la cual presenta una meseta correspondiente al área iluminada (línea negra) sobre fondo blanco. En la Figura 3, se muestra su derivada calculada, donde los cambios abruptos de intensidad generan picos positivos y negativos bien definidos, correspondientes a los bordes izquierdo y derecho de la línea. Esta observación confirmó empíricamente el funcionamiento de la derivada como herramienta efectiva para la detección de bordes.

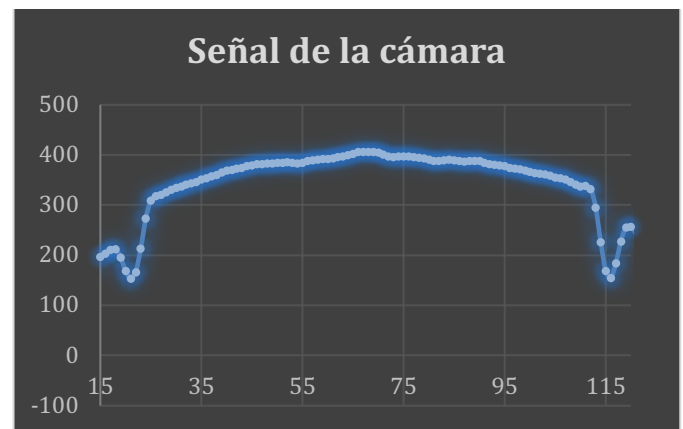


Figura 2. Representación de la señal capturada por la cámara

La derivada discreta se calculó mediante una simple resta entre pares de valores consecutivos. En su forma más básica, se usó:

$$D[n] = I[n] - I[n-1]$$

Este arreglo de diferencias resalta con claridad los bordes de la línea detectada en la imagen. Para mejorar la robustez del sistema, se evaluaron diferentes estrategias de comparación: uso de vecinos separados por dos o tres píxeles, promedio móvil para suavizar ruido, y umbrales adaptativos para filtrar picos falsos. En la Figura 3, se muestra su derivada calculada, donde los cambios abruptos de intensidad generan picos positivos y negativos bien definidos, correspondientes a los bordes izquierdo y derecho de la línea. Esta observación confirmó empíricamente el funcionamiento de la derivada como herramienta efectiva para la detección de bordes.

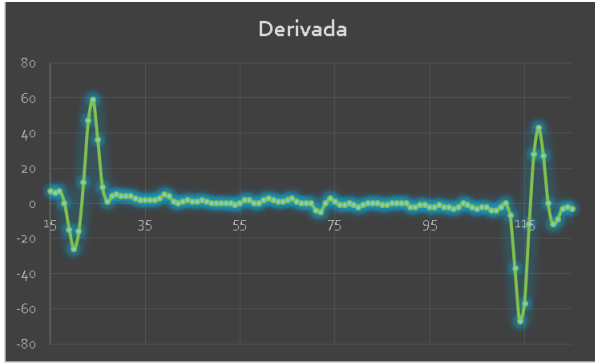


Figura 3. Representación de la señal derivada

Implementación del algoritmo:

El código se escribió en C, dentro del entorno CodeWarrior. Una vez adquirida cada imagen, el arreglo se procesaba de inmediato para obtener la derivada. Luego, se identificaban los picos positivo y negativo de mayor magnitud, correspondientes a los bordes izquierdo y derecho de la línea.

```
for(pixel_deriv = 1 ; pixel_deriv < 128 ; pixel_deriv++)
{
    pnter_buffer_strg[pixel_deriv] = (int16_t)(pnter_2_pnter_camX[pixel_deriv] - pnter_2_pnter_camX[pixel_deriv-1]);
}
```

Figura 4. Fragmento de código en lenguaje C para el cálculo de la derivada discreta

La posición central de la línea se calculó promediando ambos bordes detectados. Esta posición era usada como entrada de un sistema de control PD, que ajustaba la velocidad relativa de los motores para mantener el robot centrado sobre la trayectoria.

Validación en pista:

Se realizaron pruebas en diferentes condiciones: línea recta, curvas amplias y cerradas. Los datos se enviaban a una terminal serial y se almacenaban en hojas de cálculo para su análisis. Las gráficas obtenidas permitieron ajustar parámetros del algoritmo derivativo y del controlador para mejorar la precisión y reducir oscilaciones.

Resultados

Uno de los principales aportes de este proyecto fue la aplicación de la razón de cambio, expresada mediante la derivada discreta, como herramienta para el procesamiento de señales provenientes de la cámara lineal. En términos matemáticos, la derivada representa la velocidad con la que una magnitud varía respecto al tiempo o a una variable independiente. En el contexto del robot seguidor de línea, esta magnitud es la intensidad luminosa registrada por cada píxel de

la cámara, y la variable sobre la que cambia es la posición lineal a lo largo del arreglo de 128 píxeles.

Implementar esta razón de cambio permitió detectar bordes mediante los puntos de mayor pendiente en la curva de intensidad. Es decir, los máximos (picos positivos) y mínimos (picos negativos) del arreglo derivado indicaban con claridad dónde empezaba y terminaba la línea negra sobre fondo blanco. Esta técnica es más robusta que simplemente buscar el valor mínimo de intensidad, ya que ignora pequeñas irregularidades o ruido presentes en la señal original.

Matemáticamente, se implementó la siguiente fórmula de derivación discreta:

$$D[n] = I[n] - I[n-1]$$

Durante las pruebas, se comprobó que esta derivada generaba picos consistentes en los bordes de la línea detectada. Se desarrollaron funciones específicas en el código fuente (fnbrightest_pixel_Deriv y fnDimmest_pixel_Deriv), las cuales identificaban la posición de los picos de máxima y mínima pendiente, respectivamente. A partir de estos puntos se calculaba el centro de la línea como: $C = (p_{\text{máx}} + p_{\text{mín}}) / 2$

Esta posición central alimentaba el sistema de control PD para el servomotor, permitiendo un ajuste ágil y preciso de la dirección del robot. Adicionalmente, se implementó una rutina de detección de línea de arranque mediante el ancho de la derivada, permitiendo que el robot se detuviera automáticamente al completar el circuito dos veces.

Durante la experimentación, se observó que el uso de la última muestra disponible para el cálculo de la derivada no siempre producía resultados óptimos, particularmente en lo que respecta a la claridad y magnitud de los picos esperados. En la Figura 3, vimos una gráfica obtenida al utilizar la última muestra para la derivación, mientras que en la Figura 5 se observa el resultado al emplear la penúltima muestra. La diferencia en la nitidez de los picos es evidente, siendo más acentuados y distinguibles en la segunda opción. Este comportamiento se debe a la variabilidad en los datos de adquisición, donde el último valor puede estar afectado por ruido o inestabilidad en la conversión analógica-digital. Como parte de la solución, se implementó un ajuste en el código para permitir la comparación entre muestras anteriores (por ejemplo, $I[n]-I[n-2]$), logrando así una señal derivada más robusta para el control del robot.

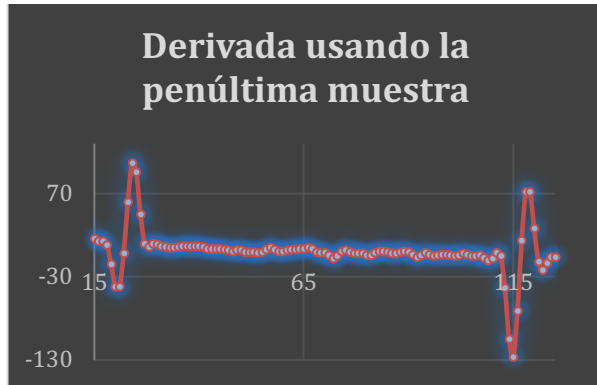


Figura 5. Representación usando la penúltima derivada

Los resultados experimentales mostraron un seguimiento de línea eficiente en condiciones de pista recta, curvas cerradas y variaciones de iluminación. La señal derivada no solo ofreció mejor detección de bordes, sino que permitió implementar decisiones basadas en umbrales dinámicos y lógica adaptativa. La integración del modelo derivativo demostró ser un caso práctico exitoso del uso de matemáticas aplicadas en un sistema embebido.

Gracias a esta implementación, el robot se posicionó como ganador del Freescale Cup en dos ediciones consecutivas, validando tanto el enfoque técnico como su efectividad en un entorno competitivo real.

Referencias

1. Elmenreich, W., y Klingler, G. (2006). *Embedded computer vision for line following robots*. In Proceedings of the 3rd IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems (pp. 643–648). <https://doi.org/10.1109/IDAACS.2005.283033>
2. Gonzalez, R. C., y Woods, R. E. (2008). *Digital Image Processing* (3rd ed.). Pearson Prentice Hall.
3. Ogata, K. (2010). *Ingeniería de control moderna* (5ª ed.). Pearson Educación.
4. Texas Instruments. (2011). *TSL1401 Linear Sensor Array datasheet*. Retrieved from <https://www.ti.com/lit/ds/symlink/tsl1401.pdf>
5. Dorf, R. C., y Bishop, R. H. (2011). *Modern Control Systems* (12th ed.). Prentice Hall.
6. BrokenChips Team. (2013). *Technical Report – Freescale Cup 2013*. Instituto Tecnológico de Ciudad Guzmán.