

UNIVERSIDAD SIMÓN BOLÍVAR
COORDINACIÓN DE MAESTRÍA EN TECNOLOGÍA E INGENIERÍA ELECTRÓNICA,
MENCIÓN MECATRÓNICA

EC7136 - Electrónica de Potencia II

Profesor: José A. Restrepo

Estudiante: William Chacón

Carnet: 20-91334

TAREA 2:

INTRODUCCIÓN

En el presente escrito se plantea un sistema de control de 2 lazos para una máquina DC de imán permanente (Brushed DC Motor). En este caso, se consideró un modelo lineal sencillo, en el que se despreciaron los efectos de la fricción viscosas y de coulomb, así como posibles elementos térmicos. Para el mismo, como posible solución para un control de velocidad que se encargue de mantener el valor de la corriente, se decidió tomar dos lazos de control para cada una de las variables controladas.

A continuación, se hará una breve descripción del problema, así como una posible solución a partir de la implementación de controladores PI para cada lazo, con los valores de las constantes correspondientes. Adicionalmente, se procede a resolver el problema con varios métodos numéricos a fin de tener diferentes elementos para comparar los resultados obtenidos.

DESCRIPCIÓN DEL PROBLEMA

El circuito del motor DC sugerido para esta prueba se trata del modelo básico descrito por la ecuación (1). El mismo no contempla los efectos de las fuerzas no conservativas del sistema, así como tampoco fenómenos no lineales presentes en el mismo.

$$\begin{bmatrix} \dot{I} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{K_m}{L_a} \\ K_m/J_e & -\frac{B}{J_e} \end{bmatrix} \begin{bmatrix} I \\ \omega \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} & 0 \end{bmatrix} V_{in} \quad (1)$$

Con base las ecuaciones de estado mostradas anteriormente, se desarrolla un sistema de control tal que pueda alimentar de forma satisfactoria la máquina, proporcionando la salida deseada del sistema. Este se muestra en la Figura 1. Para poder asegurar un correcto control de velocidad angular para el motor, es necesario incluir un lazo interno (más rápido) que pueda garantizar el evadir picos de corriente que pudiesen ser perjudiciales para el equipo.

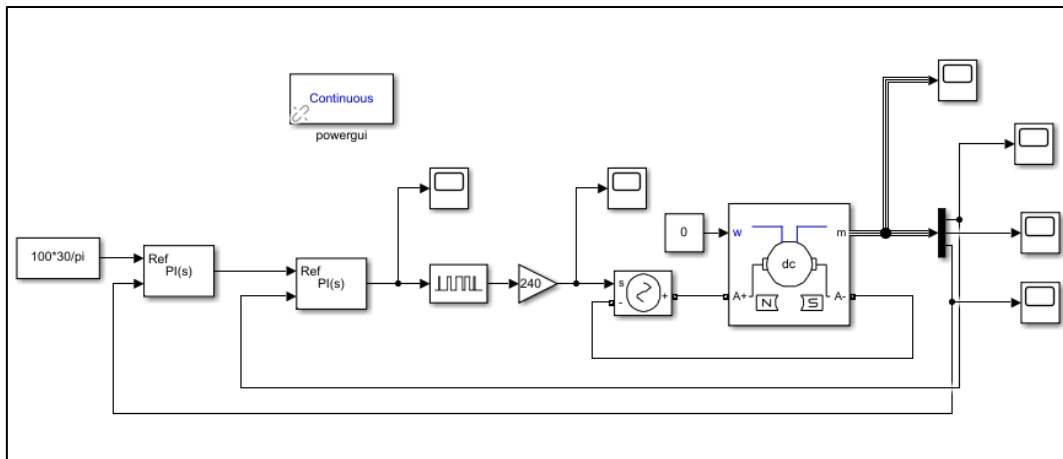


Figura 1: Diagrama del sistema de control de dos lazos para la máquina DC. El lazo externo corresponde al lazo de control de la velocidad angular y el lazo de control interno a la corriente.

Para la resolución. Se implementaron diversos algoritmos de solución de ecuaciones diferenciales, con la finalidad de tener diferentes soluciones que permita validar el comportamiento de la respuesta, así como visualizar los ligeros cambios que poseen los métodos más sencillos y menos precisos respecto a los más complejos y precisos. Estos métodos son:

- Método de Euler (EU)
- Método de Runge-Kutta de Segundo Orden (RK2)
- Método de Runge-Kutta de cuarto Orden (RK4)
- Método de Runge-Kutta de quinto Orden (RK5)
- Sarafyan's Runge-Kutta 4to Orden (SRK4)
- Sarafyan's Runge-Kutta 5to Orden (SRK5)
- Runge-Kutta–Fehlberg 4to Orden (RKF4)
- Runge-Kutta–Fehlberg 5to Orden (RKF5)
- Runge-Kutta Orden 6to Orden (RK6)
- Runge-Kutta–Fehlberg Orden 7 (RKF7)
- Huta's Improved Runge-Kutta 6to Orden (HIRK6)

CODIGO

El código presente, desarrollado en C, mantiene la estructura del código de la experiencia anterior. El archivo principal “*main.c*” contiene todos los procesos necesarios para resolver el problema.

En las declaraciones pertinentes para las variables y funciones a implementar. Dentro de la función principal, la variable “*NMtdStr*” se encarga de seleccionar el método de solución que será almacenado en el puntero “**NumMethod[]*”, de acuerdo al valor de la misma. Por los momentos, “*NMtdStr*” debe ser cambiado manualmente en el código, pero se aspira a implementar el código que permita al usuario introducir el valor del mismo directamente desde la consola, a modo de tener que realizar menos modificaciones al código previo a su ejecución. A continuación, se especifica el valor correspondiente para cada uno de los métodos ofrecidos con el respectivo nombre de cada una de estas funciones.

- 0: ODE_EU (EU)
- 1: ODE_RK2 (RK2)
- 2: ODE_RK4 (RK4)
- 3: ODE_RK5 (RK5)
- 4: ODE_SRK4 (SRK4)
- 5: ODE_SRK5 (SRK5)
- 6: ODE_RKF4 (RKF4)
- 7: ODE_RKF5 (RKF5)
- 8: ODE_RK6 (RK6)
- 9: ODE_RKF7 (RKF7)
- 10: ODE_HIRK4 (HIRK6)

Posteriormente a ello, dentro de la función principal, se salta a la función “*Solve*”, que contempla el código encargado de ejecutar el proceso de solución del problema. Para lograr la implementación del PWM, se procede a crear un bucle interno dentro del bucle principal que cumpla con las interacciones intermedias entre cada período del PWM. Para ello, se definió una frecuencia menor a la de conmutación que corresponde al paso mínimo medible entre cada PWM.

La idea de escribir el código de esta forma, es que un usuario que desee modificarlo para implementarlo solo tenga que realizar modificaciones en las primeras partes del código (y, de requerirlo, agregar métodos de solución al final), sin necesidad de tener que modificar el resto del código, cambiando nombres o valores, para que el mismo pueda ejecutarse correctamente.

El código en Matlab consta de un conjunto de un conjunto de funciones para hacer gráficas. Se recomienda correr la simulación en simulink y todos los métodos de solución del código de C para que se pueda efectuar de forma apropiada el archivo .m sin necesidad de modificarlo.

Por conveniencia, se recomienda mantener todos los códigos en la misma carpeta, pues el archivo .m fue programado para leer los valores de los archivos .dat generados por las ejecuciones del código .c directamente de la misma carpeta.

RESULTADOS

En esta experiencia, el proceso de afinación de ambos controladores fue realizado manualmente. De esta práctica, se obtuvo que los valores para el control de velocidad angular y de corriente son:

Lazo externo de control:

$$K_p = 2.33; \quad K_I = 6.17$$

Lazo interno de control:

$$K_p = 0.108; \quad K_I = 2.28$$

A continuación, se procede a presentar los resultados para cada uno de los métodos implementados en el código en C.

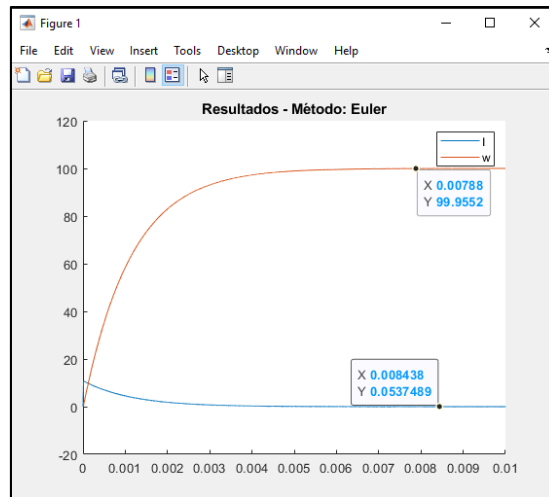


Figura 2: Resultados para Euler. Curva naranja representa la velocidad angular en rad/s y la azul la corriente en A.

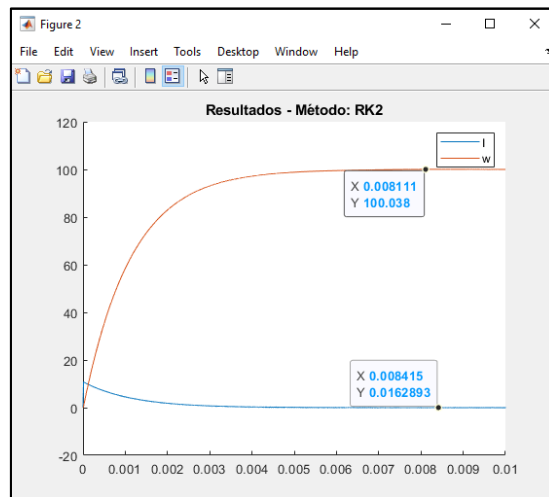


Figura 3: Resultados para RK2. Curva naranja representa la velocidad angular en rad/s y la azul la corriente en A.

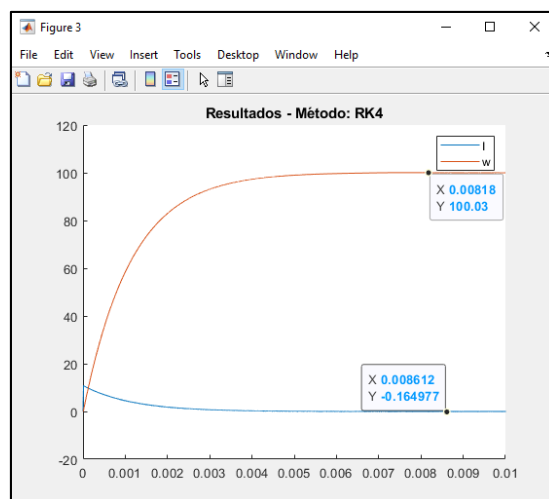


Figura 4: Resultados para RK4. Curva naranja representa la velocidad angular en rad/s y la azul la corriente en A.

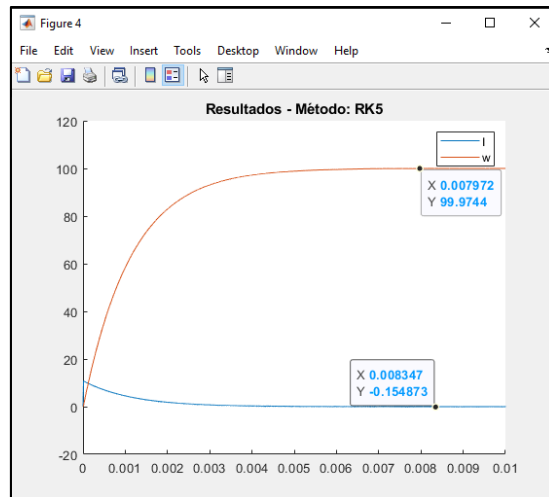


Figura 5: Resultados para RK5. Curva naranja representa la velocidad angular en rad/s y la azul la corriente en A.

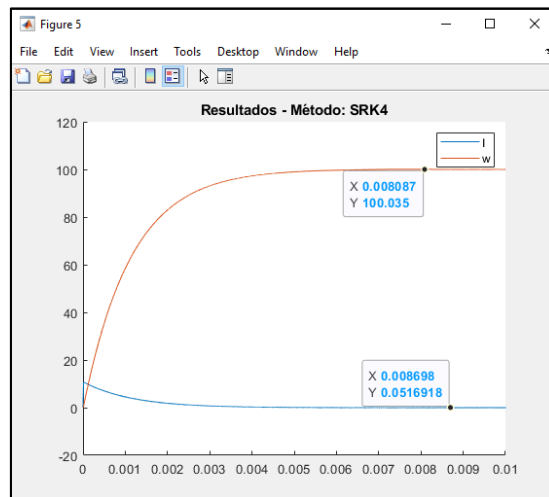


Figura 6: Resultados para SRK4. Curva naranja representa la velocidad angular en rad/s y la azul la corriente en A.

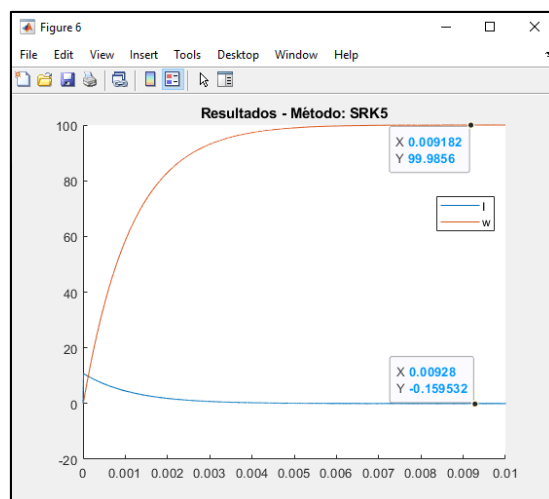


Figura 7: Resultados para SRK5. Curva naranja representa la velocidad angular en rad/s y la azul la corriente en A.

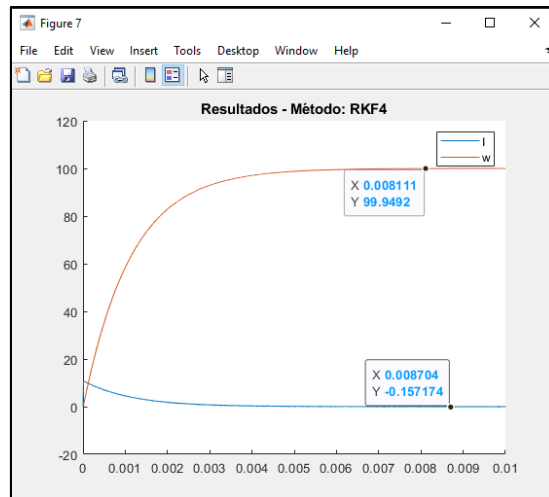


Figura 8: Resultados para RKF4. Curva naranja representa la velocidad angular en rad/s y la azul la corriente en A.

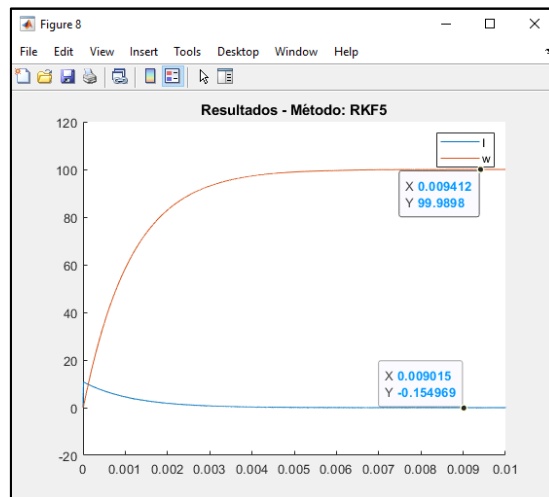


Figura 9: Resultados para RKF5. Curva naranja representa la velocidad angular en rad/s y la azul la corriente en A.

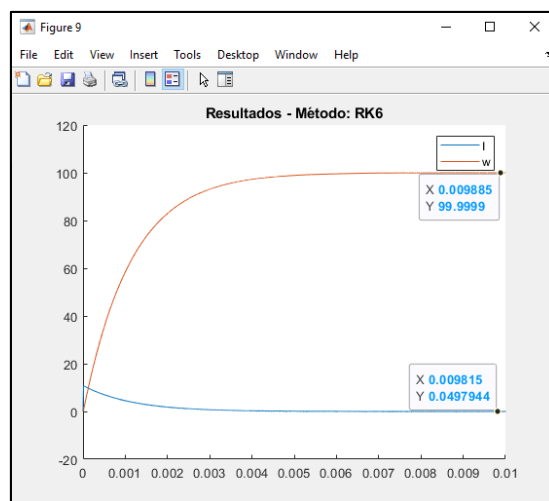


Figura 10: Resultados para RK6. Curva naranja representa la velocidad angular en rad/s y la azul la corriente en A.

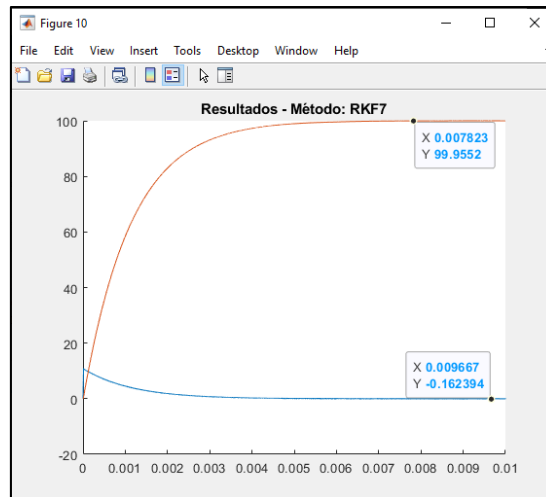


Figura 11: Resultados para RKF7. Curva naranja representa la velocidad angular en rad/s y la azul la corriente en A.

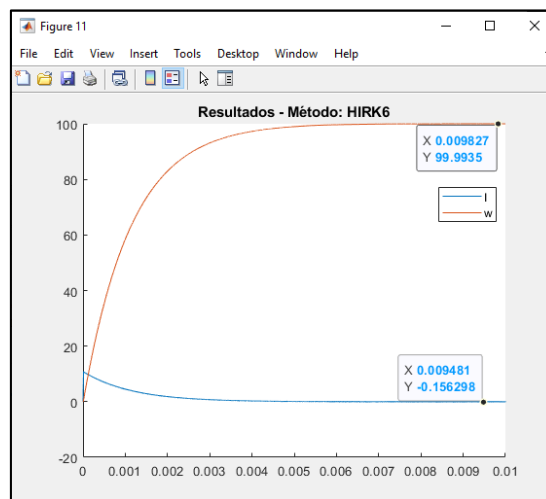


Figura 12: Resultados para HIRK6. Curva naranja representa la velocidad angular en rad/s y la azul la corriente en A.

En el caso de la simulación del sistema en Simulink/Matlab, los resultados no fueron los esperados, ya que se encuentran completamente alejados de los esperados realmente. En vista de dicha discrepancia, no se pudo realizar una comparación real entre los resultados obtenidos por los códigos en C. en las figuras 13 y 14 se pueden apreciar los resultados de la simulación, presentada en la Figura 2, de la corriente y velocidad angular respectivamente.

La enorme discrepancia de la simulación, incluso respecto al comportamiento deseado de la máquina, se considera un error en la configuración de la misma que, para la fecha de entrega de este proyecto, no se logró solventar. En vista de no tener un método que permita comparar los resultados arrojados por la simulación hecha en C, solo queda realizar un análisis cualitativo de los resultados obtenidos de la misma. De esta forma, se procede a comparar los resultados arrojados. Dicha comparación puede apreciarse en la Figura 15 y 16 para los resultados obtenidos de velocidad angular y corriente respectivamente.

En la figura 15 se puede apreciar la gran similitud en los resultados de velocidad angular de cada uno de los métodos implementados, lo que indica que la precisión es muy similar entre

ellos para esta variable. Por otra parte, para el caso de la corriente, claramente se ve una enorme discrepancia entre los resultados de cada método (ver Figura 16). En este caso, ya que no se logró configurar la simulación en Simulink/Matlab, solo se puede sugerir tomar los valores de los métodos de mayor orden de precisión como referencia más precisa.

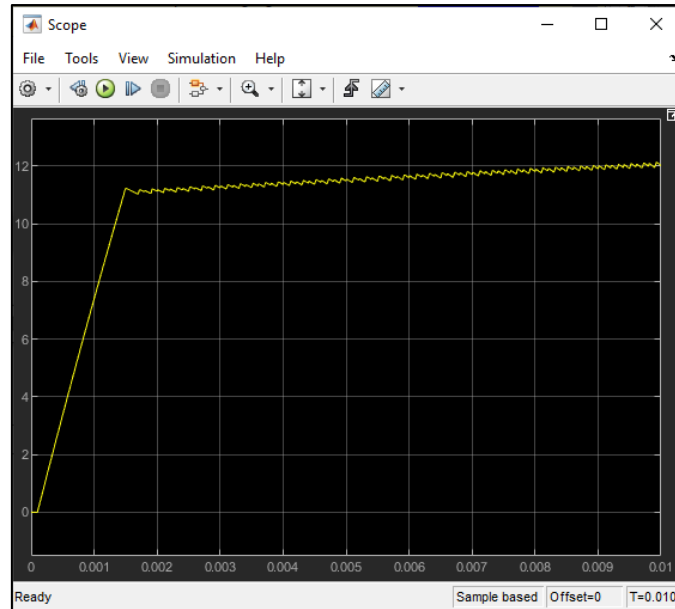


Figura 13: Corriente resultante de la simulación en Simulink/Matlab.

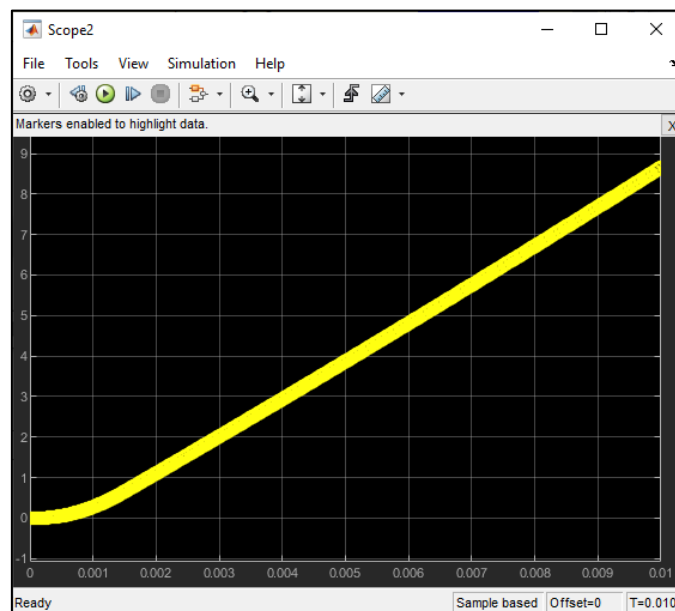


Figura 14: Velocidad angular resultante de la simulación en Simulink/Matlab.

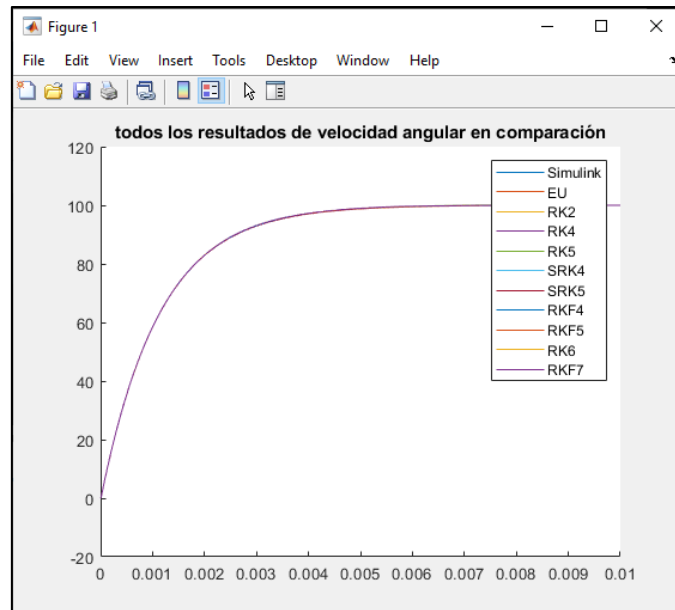


Figura 15: Velocidad angular resultante de todos los métodos programados en C.

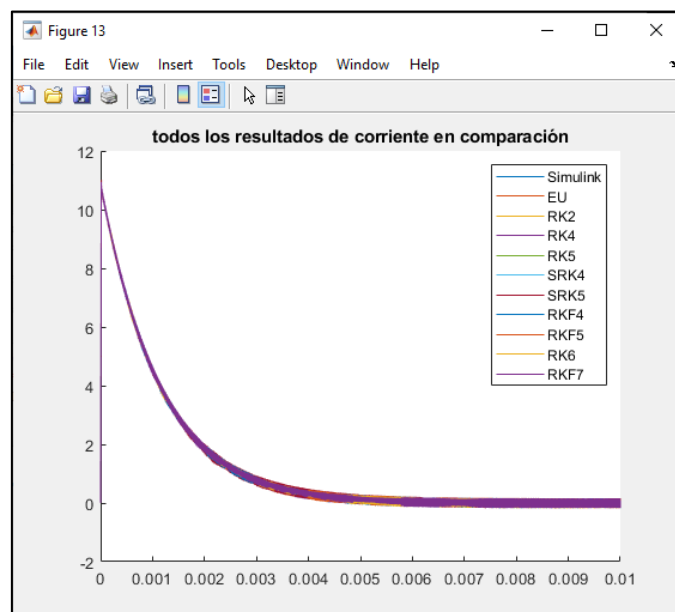


Figura 16: corriente resultante de todos los métodos programados en C.

CONCLUSIONES

De la siguiente experiencia se pudo diseñar un controlador apropiado para la velocidad angular y corriente del sistema. Los mismos permiten obtener resultados sin amortiguamiento. Así mismo, el tiempo de respuesta es considerablemente rápido y su pico de corriente considerablemente bajo (no mayor a 11A). esto refleja la efectividad del controlador.

En el presente escrito, no se pudo realizar un análisis comparativo de los errores cuadráticos medios respecto a la simulación en Simulink/ Matlab debido a no haberse podido

calibrar la misma durante las pruebas. No obstante, el comportamiento arrojado por el código en C resulta ser bastante prometedor, por lo que se debe corregir la simulación en Simulink/Matlab a modo de poder obtener resultados concretos de cada método, así como de la validez del modelo programado en C y la precisión relativa de cada uno de los métodos de solución de ecuaciones diferenciales respecto al resultado del otro software implementado.

Sin embargo, igualmente se ha de concluir con una exitosa propuesta de control, con expectativa de validez sus resultados con pruebas futuras.