



Facultad de Ingeniería

ISSN: 0121-1129

revista.ingenieria@uptc.edu.co

Universidad Pedagógica y Tecnológica
de Colombia
Colombia

Páez-Logreira, Heyder David; Zamora-Musa, Ronald; Bohórquez-Pérez, José
Programación de Controladores Lógicos (PLC) mediante Ladder y Lenguaje de Control
Estructurado (SCL) en MATLAB
Facultad de Ingeniería, vol. 24, núm. 39, julio-diciembre, 2015, pp. 109-119
Universidad Pedagógica y Tecnológica de Colombia
Tunja, Colombia

Disponible en: <http://www.redalyc.org/articulo.oa?id=413940776010>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica
Red de Revistas Científicas de América Latina, el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

Programación de Controladores Lógicos (PLC) mediante Ladder y Lenguaje de Control Estructurado (SCL) en MATLAB

Programming logic controllers (PLC) using ladder and structured control language (SCL) in MATLAB

Programação de Controladores Lógicos (PLC) mediante Ladder e Linguagem de Controle Estruturado (SCL) em MATLAB

Fecha de Recepción: 03 de Diciembre de 2014
Fecha de Aceptación: 04 de Abril de 2015

Heyder David Páez-Logreira*
Ronald Zamora-Musa**
José Bohórquez-Pérez***

Resumen

Existen diferentes lenguajes de programación de Controladores Lógicos Programables (PLC), el más común de ellos es el lenguaje de programación gráfico para autómatas programables llamado LADDER. Sin embargo, el lenguaje LADDER no es amigable en programas que requieren de cálculos de ecuaciones matemáticas, por ejemplo, el cálculo del volumen de un tanque en un control de nivel. Lenguajes alternativos como el Lenguaje de Control Estructurado (SCL) son menos estudiados dada su complejidad y poca popularidad en la pequeña y mediana industria. En este artículo se compara los resultados del lenguaje de programación LADDER y el lenguaje SCL generado mediante la herramienta Simulink® PLC Coder de MATLAB® tomando como caso de estudio el cálculo de volumen de un tanque. Finalmente, este trabajo demuestra que la herramienta Simulink® PLC Coder presenta una alternativa de desarrollo más eficiente.

Palabras Claves: MATLAB, Simulink PLC Coder, LADDER, Bloque de función FB, PLC.

* Universidad de la Costa (Barranquilla – Atlántico, Colombia). hpaez@cuc.edu.co.

** Universidad de la Costa (Barranquilla – Atlántico, Colombia). rzamora2@cuc.edu.co.

*** Universidad de la Costa (Barranquilla – Atlántico, Colombia). jbohorper@gmail.com.

Abstract

There are different programming languages for Programmable Logic Controllers (PLC), the most common of these is the graphical programming language called LADDER. However, LADDER is not friendly in programs requiring calculations of mathematical equations, for example, to calculate the volume of a tank on level control situations. Alternative languages such as Structured Control Language (SCL) are less studied because of their complexity and unpopular in small and medium industry. In this article, the results LADDER programming language and language are compared with SCL generated by Simulink PLC Coder MATLAB tool taking as a case study the calculating the volume of a tank. Finally, this work shows that the Simulink PLC Coder tool presents an alternative to more efficient development.

Keywords: MATLAB, Simulink PLC Coder, LADDER, Function Block FB, PLC.

Resumo

Existem diferentes linguagens de programação de Controladores Lógicos Programáveis (PLC), o mais comum deles é a linguagem de programação gráfica para autômatos programáveis chamada LADDER. Porém, a linguagem LADDER não é amigável em programas que requerem de cálculos de equações matemáticas, por exemplo, o cálculo do volume de um tanque em um controle de nível. Linguagens alternativas como a Linguagem de Controle Estruturado (SCL) são menos estudadas devido a sua complexidade e pouca popularidade na pequena e média indústria. Neste artigo se compara os resultados da linguagem de programação LADDER e a linguagem SCL gerado mediante a ferramenta Simulink® PLC Coder de MATLAB®, tomando como caso de estudo o cálculo de volume de um tanque. Finalmente, este trabalho demonstra que a ferramenta Simulink® PLC Coder apresenta uma alternativa de desenvolvimento mais eficiente.

Palavras Chave: MATLAB, Simulink PLC Coder, LADDER, Bloco de função FB, PLC.

Cómo citar este artículo:

[1] H Páez-Logreira, R. Zamora-Musa & J. Bohórquez-Pérez, “Programación de Controladores Lógicos (PLC) mediante Ladder y Lenguaje de Control Estructurado (SCL) en MATLAB”, Fac. Ing., vol. 24 (39), pp. 109–119, Mayo-Ago. 2015.

I. INTRODUCCIÓN

En la actualidad existen diferentes lenguajes de programación que permiten a los programadores de PLC (*Programmable Logic Controller*) [2] expresar y ejecutar la lógica que resuelve una problemática o desarrolle un cálculo matemático correspondiente a una planta o proceso industrial. Lenguajes como LADDER (Lenguaje de contactos o en escalera), AWL (del alemán *Anweisungsliste* que significa lista de instrucciones), CFC (de las siglas en inglés *Continuous Function Chart*) y SCL (de las siglas en inglés *Structured Control Language*) son imprescindibles a la hora de realizar operaciones matemáticas que se requieran en un proceso.

El cálculo matemático en un proceso industrial permite evaluar y medir variables indirectas del proceso. En los diferentes lenguajes de programación de PLC, es posible insertar un cálculo matemático. Sin embargo, esta tarea puede resultar compleja según el número de variables de la ecuación y las capacidades del lenguaje.

Los bloques de función, o FB por las siglas en inglés para *Function Blocks*, juegan un papel importante en el control y programación de estos procesos industriales [1], por ejemplo, en el cálculo del volumen de líquido en un tanque.

El lenguaje empleado puede variar la dificultad de inserción e interpretación de ecuaciones matemáticas en los Bloques del programa. Insertar una ecuación matemática en un proceso empleando un software de cálculo matemático, por ejemplo MATLAB®, facilita la programación de un bloque de función FB para el cálculo del volumen y nivel de un tanque.

La herramienta *Simulink® PLC Coder*, incluida en el paquete de instalación del software de cálculo matemático MATLAB®, permite exportar un bloque de función matemático hecho en SIMULINK® y programado en MATLAB® a un lenguaje de programación con el estándar internacional IEC 61131 entendible por los PLC.

En este artículo se propone utilizar la herramienta *Simulink® PLC Coder*, para diseñar un bloque de función FB que calcule el volumen y nivel de un

tanque. Se realiza la comparación con un bloque de función FB diseñado en Lenguaje LADDER para el mismo escenario. La programación y ejecución se realiza en el software SIMATIC Manager, sobre un PLC S7-400 (CPU 416-2 DP), de la marca SIEMENS.

El artículo está organizado de la siguiente manera: en la siguiente sección se desarrolla una revisión del estado de la literatura incluyendo trabajos relacionados, a continuación en la sección de métodos y materiales se especifica los pasos del desarrollo e implementación de los programas que permiten comparar el lenguaje Ladder y SCL. Posteriormente se presentan los resultados obtenidos y se realiza el análisis de los mismos, y finalmente en la última sección se presentan las conclusiones obtenidas de la investigación planteada.

II. DISEÑO DE BLOQUES DE FUNCIÓN FB PARA PROGRAMACIÓN DE PLC

Los bloques de función FB son esenciales y son usados en los PLC de la marca SIEMENS. Estos tienen la ventaja de ser invocados desde el programa principal OB. Los datos de las variables de los FB se van guardando en posiciones de memoria llamadas DB, con el fin de que estos datos puedan ser usados posteriormente en el programa principal. Los FB tienen un parámetro llamado STATIC; cuando se llama a un FB, se genera una instancia DB que acompaña a esta llamada del FB. Dentro del DB están todos los parámetros estáticos del FB, disponibles en cualquier momento para otro bloque en el programa, es decir se guardan todos los valores de las variables contenidas en STATIC automáticamente en el FB.

El uso de bloques de función FB resulta conveniente en el control y programación de un proceso industrial. Los Bloques de Función FB permiten ser invocados simultáneamente en un mismo programa con una DB distinta para cada llamado. Garantizando que cada FB tenga su propia instancia de DB y obtener datos independientes en cada bloque insertado en el programa principal OB.

Los bloques de función son definidos por el estándar internacional IEC 61499 [3, 4], el cual ha sido desarrollado conjuntamente con el estándar IEC 61131-3 que define aspectos de los FB [5]. Existen

aplicaciones comunes usando FB, como el diseño de sistemas distribuidos, control de nivel en tanques de un proceso, sistemas de control industrial basado en FB [6], diseño de prototipos en diferentes niveles de control [7, 8], y arquitecturas de desarrollo de soporte a ingeniería usando FB [9].

A. Estándar Internacional IEC 61131-3

El estándar internacional IEC 61131 es una colección completa de estándares referentes a controladores programables y sus periféricos asociados [10]. Este estándar internacional es utilizado en los lenguajes de programación de controladores lógicos programables (PLC).

El estándar 61131-3 define los bloques de función, los cuales proveen un mejor diseño, flexibilidad y eficiencia en los proyectos de programación de PLC [10]. Adicionalmente, define la modularidad de los FB logrando una estructura que genera un contexto más simple y también la habilidad de integrar escenarios tipo planta (como el de esta investigación) y modelaje en una sola ejecución y marco de desarrollo [11].

Sin embargo, en los escenarios tipo planta se presentan algunas dificultades cuando se programa con este tipo de herramientas como la falta de relojes globales, ausencia de estados globales y dificultad para sincronizar procesos [12]; existen herramientas de diseño que pueden mejorar sustancialmente los inconvenientes descritos [13] como por ejemplo MATLAB®.

III. MATLAB® Y SIMULINK® PLC CODER

MATLAB® es uno de los software de cálculo matemático más utilizado en las industrias a nivel global, teniendo en cuenta que con su uso se facilita el cálculo de sistemas complejos y notaciones matemáticas indispensables en un proceso industrial.

Simulink® PLC Coder, es una herramienta incluida en el paquete de instalación del software de cálculo matemático MATLAB® la cual permite generar código de texto estructurado compatible con el estándar internacional IEC-61131, a partir de modelos de Simulink®. Este software aporta el enfoque de

diseño basado en modelos en el dominio y desarrollo del PLC permitiendo que los programadores puedan dedicar más tiempo a los algoritmos y modelos finos de ajuste a través de la creación de prototipos y la experimentación rápida de sistemas complejos, ahorrando tiempo en la codificación de estos sistemas en el PLC [14].

Como se mencionó anteriormente Simulink® PLC Coder posee herramientas modernas que pueden generar de manera sencilla código textual o generar código con el estándar IEC 61131 compatible para la programación de PLC [15]. Esta herramienta y método de generar código compatible con los PLC es de gran utilidad a la hora de enfrentarse a sistemas complejos con notaciones matemáticas que involucren uno o más variables de un sistema de control industrial.

A. Método de transformación del modelo

La sintaxis y semántica de MATLAB® y Simulink® asociados a los bloques de función debe entenderse de manera formal, para lo cual se realizan algunas consideraciones entre ambas aplicaciones antes de la transformación de los modelos. El comportamiento de los modelos debe ser el mismo aun frente a cambios de plataforma, como el PLC, esto garantiza la equivalencia semántica y un mapeo formal entre los dos idiomas, que se establece al crear un bloque de función asociado a una función descrita en el lenguaje de programación del software de cálculo matemático MATLAB® [10].

Simulink® y los bloques de función son objetos que utilizan una notación matemática para modelar un sistema de control automático. El método de transformación del modelo garantiza la equivalencia entre ambos después de relacionar una notación matemática hacia un objeto o bloque de función insertado en Simulink®.

Para reducir el conflicto natural de ejecución o transformación entre dos modelos, una de las mejores opciones es la implementación de bloques de función modelados con escenarios tipo planta a través de Simulink® [16, 17].

En la Fig. 1 [10], se muestra un método de transformación de equivalencia semántica del modelo

FB generado con un modelo original Simulink®; el bloque de función se conforma de $N+1$ subsistemas internos en él.

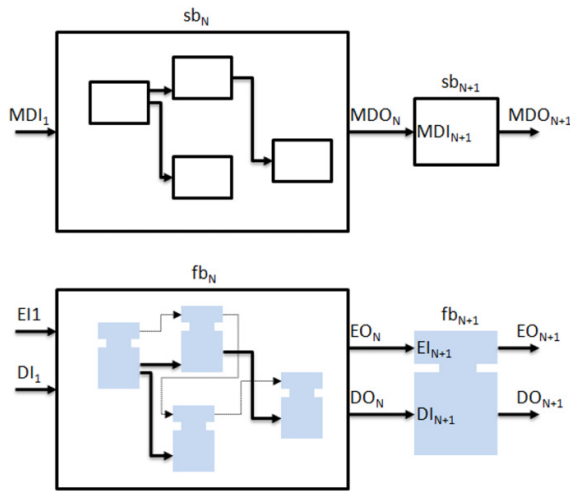


Fig. 1. Bloque de función con $N+1$ sistemas, Adaptado de [10].

B. Lenguaje de control estructurado SCL

SCL es un lenguaje de programación de alto nivel disponible en SIMATIC y definido por la norma IEC 1331-3, este se programa en PASCAL y crea una fuente con la extensión .scl, la cual al ser compilada puede ser utilizada para la creación del programa principal (OB) y los bloques de función (FB) o bloques de función sin variables estáticas (FC) [17].

El lenguaje SCL es un lenguaje de programación de alto nivel que permite: el uso de expresiones matemáticas complejas, fórmulas, tipos de datos definidos por el usuario, manejo de tablas, arreglos, matrices, programación de bucles y flujos condicionados, y también es apropiado para gestionar gran cantidad de datos.

Los bloques de función FB tienen un DB de instancia (Bloque de Datos) asociado únicamente al bloque FB. Por otro lado, los bloques de función FC están asociados al DB General, es decir, los datos empleados y generados por un bloque FC son almacenados en el DB General, y están disponibles para todo bloque del programa.

Este escenario motiva a realizar una evaluación de los resultados de programación en alto nivel empleando

lenguaje SCL y el uso del tradicional lenguaje Ladder en la programación de un PLC.

IV. MÉTODOS Y MATERIALES

Se presenta un problema convencional en los procesos industriales para comparar los resultados obtenidos en las herramientas SCL Siemens y PLC Coder de MATLAB®.

El sistema consta de un tanque de almacenamiento y el problema corresponde a la medición indirecta del volumen de agua contenido en el tanque a partir de la medición de nivel.

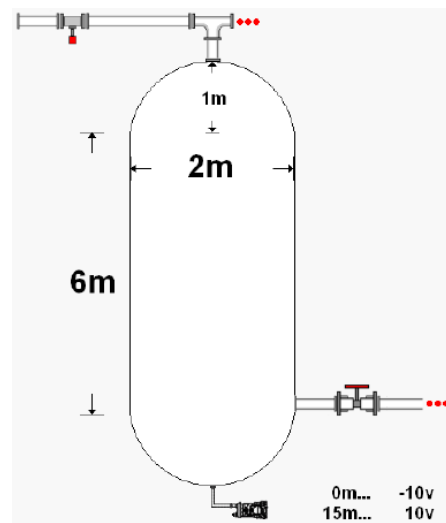


FIG. 2. Tanque del Proceso

El tanque se divide en tres secciones según la forma geométrica de cada una de estas. Los casquetes superior e inferior del tanque poseen forma de media esfera, con un metro (1m) de radio, la sección central del tanque es de forma cilíndrica con 6 metros de altura y diámetro de 2 metros.

El sensor de nivel tiene un rango de medida entre 0 y 15 metros y una escala de voltaje de salida de -10 Voltios a 10 Voltios. El sensor está conectado a un PLC SIEMENS S7-400 con CPU referencia 416-2 DP.

El problema es analizado y modelado matemáticamente según estas especificaciones y su solución es diseñada empleando Bloques de Función (FB). En primer lugar, se generan los Bloques FB con programación

tradicional en Lenguaje Ladder, y segundo, mediante la herramienta Simulink® PLC Coder de MATLAB®.

A. Análisis y Modelamiento matemático del problema

Se debe describir la relación entre la altura total del tanque, el rango de medida del sensor y el rango de la entrada analógica del PLC.

La ecuación (1) define matemáticamente la medida de nivel (altura) de agua en el tanque.

$$h = \left(\frac{V_s - V_{\min}}{V_{\max} - V_{\min}} * (h_{\max} - h_{\min}) \right) + h_{\min} \quad (1)$$

Donde,

V_s : Valor medido por el Sensor
 V_{\min} : Voltaje mínimo de entrada PLC
 V_{\max} : Voltaje Máximo de entrada PLC
 h_{\max} : Altura máxima del rango del Sensor
 h_{\min} : Altura mínima del rango del Sensor

Los valores de V_{\min} , V_{\max} , h_{\max} y h_{\min} están especificados en los datos técnicos del PLC SIEMENS S7-400 y del Sensor de Nivel. El rango de valores cuantificados por la entrada analógica del PLC inicia en -27648.0 hasta 27648.0, y el rango del sensor va de 0 a 15 metros.

La ecuación (2) se obtiene al reemplazar estos valores en (1):

$$h = \left(\frac{V_s + 27648}{27648 + 27648} * (15 \text{ mts}) \right) \quad (2)$$

Parametrizada la medición de nivel o altura del agua en el tanque, es necesario realizar el cálculo indirecto del volumen de agua contenido en él. Para lo cual se describe matemáticamente el volumen del tanque de acuerdo a las secciones que lo conforman: casquete inferior, sección central y casquete superior.

Empleando las ecuaciones de volumen de un cilindro (3) y una esfera (4) se modela el volumen en cada sección con respecto a la altura.

$$V = \pi r^2 * h \quad (3)$$

$$V = \frac{1}{3} * \pi r^2 * h^2 * (3r - h) \quad (4)$$

El volumen total del tanque está definido por la suma del volumen en las tres secciones: V_1 para el casquete inferior (5), V_2 para la sección central (6) y V_3 para el casquete superior (7). Cada ecuación es relativa a la altura total del tanque y las secciones inferiores y los términos conocidos como el radio han sido reemplazados.

$$V_{1(m^3)} = \frac{\pi}{3} * h^2 * (3 - h) \quad (5)$$

$$V_{2(m^3)} = \pi * (h - 1) \quad (6)$$

$$V_{3(m^3)} = \frac{\pi}{3} * (h - 7) * (3 - (h - 7)^2) \quad (7)$$

B. Implementación y Pruebas

La implementación de la solución se divide en dos: la aplicación del modelo matemático empleando lenguaje Ladder en el software SIMATIC Manager de Siemens y en segundo lugar, la aplicación en Simulink® PLC Coder de MATLAB® para la generación de un bloque FB que se ejecutará en SIMATIC Manager.

Como alternativa, la librería de herramientas de Siemens provee el bloque FC105 para escalar valores e ingreso del modelamiento matemático de la ecuación (1) y (2).

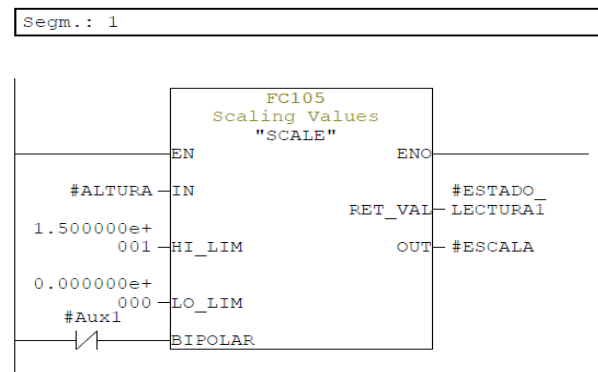


FIG. 3. Bloque FC105 para escalar valores de entrada.

Para las pruebas, se emplea la ejecución de los programas obtenidos en el software S7-PLCSIM de Siemens. S7-PLCSIM permite medir el tamaño en líneas de ejecución de cada uno de los programas, el uso de recursos y otros datos que permiten identificar la solución más eficiente en la generación de programas de PLC.

La comparación de los resultados se realiza empleando como indicadores el consumo de memoria de carga (datos), memoria de trabajo (programa) requerida, y la longitud de código de acuerdo al estándar MC7 [20].

V. RESULTADOS Y ANÁLISIS

A. Bloque de Función FB mediante lenguaje LADDER

La creación de un bloque FB empleando Lenguaje LADDER es muy común en los programadores de controladores lógicos programables.

El cálculo de la función representada por las ecuaciones (1) y (2) se implementa mediante el bloque de función FC105 disponible en la librería de SIMATIC Manager. El bloque FC105 es ilustrado en la Figura 3.

Posteriormente se realiza la implementación de las ecuaciones (5), (6) y (7), las cuales se encargan de calcular el volumen del tanque mostrado en la Figura 3 se emplean bloques de operaciones matemáticas disponibles en SIMATIC Manager, como sumadores (ADD_R), restadores (SUB_R), multiplicadores (MUL_R), y comparadores (COMP ==R, COMP >=R), de números reales, entre otros.

Los bloques de adición, substracción y multiplicación tienen como objetivo realizar las operaciones de cada uno de los términos de las ecuaciones (5), (6) y (7). Los bloques comparadores se emplean para identificar en que sección del tanque se encuentra la lectura del sensor (casquete inferior, sección central o casquete superior) y qué ecuaciones deben aplicarse y sumarse.

La totalidad del código Ladder es almacenado en un bloque de función FB llamado FB2. En total el bloque FB2 se compone de 20 Bloques FB y 11 segmentos de programación en Ladder. En la Figura 4 se muestra el bloque de función FB2 insertado en el programa principal OB1.

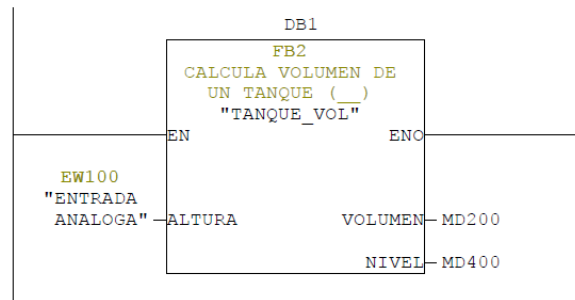


FIG. 4. Bloque de función FB programado en Lenguaje LADDER.

B. Bloque de Función FB implementado con Simulink® PLC Coder

La implementación de este bloque de función FB se realiza con el software de cálculo matemático MATLAB® y la herramienta de simulación Simulink®. Simulink® permite emular el comportamiento del bloque antes de exportarlo con la herramienta Simulink® PLC Coder a un lenguaje de programación entendible por el software SIMATIC de SIEMENS, para el caso, el lenguaje IEC 61131-3.

A continuación se presenta el algoritmo que representa el código implementado en MATLAB®.

ALTURA: Medida del sensor
 h: altura en metros
 V1: Volumen seccion 1
 V2: Volumen seccion 2
 V3: Volumen seccion 3
 Volumen: Volumen total del liquido
 Nivel: Nivel del liquido

$$h = ((ALTURA + 27648.0) / (27648.0 + 27648.0)) * 15.0)$$

$$V1 = ((1/3 * \pi) * (h^2) * (3-h))$$

$$V2 = (\pi * (h-1))$$

$$V3 = ((1/3 * \pi) * (h-7) * (3 - (h-7)^2))$$

Si $h \geq 1$

R1 = 2.09

Si $h \geq 7$

R2=18.85

R3=V3;

Sino

R2=V2

R3=0;

Fin-si

Sino

R1=V1;

R2=0;

R3=0;

Fin-Si

Volumen = R1+R2+R3

Nivel = h

El algoritmo implementado es asignado a un Bloque de Función en Simulink®, el cual permitirá su simulación y validación antes de ser exportado.

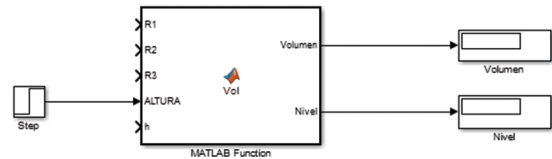


FIG. 5. Bloque de función matemática insertada en Simulink®.

Empleando la herramienta Simulink® PLC Coder se exporta la versión en lenguaje IEC 61131-3 para el algoritmo y bloque de función creado.

El código generado desde Simulink® es implementado en el PLC creando un bloque de función FB1 (Figura 6) e importando el código IEC 61131-3 obtenido.

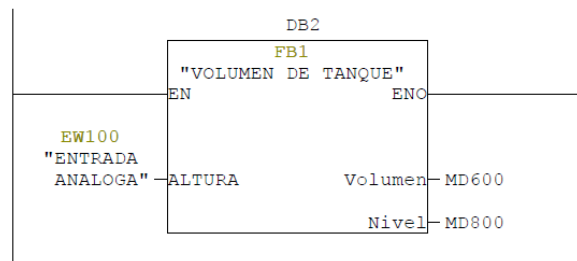


FIG. 6. Bloque de función FB, generado importado en SIMATIC generado con Simulink® PLC Coder.

C. Pruebas y validación

Para la prueba de los bloques de función FB, se empleó un proyecto en SIMATIC Manager, con una función principal OB que invoca a los dos bloques FB1 y FB2. Al ejecutar el programa en la herramienta de simulación S7-PLCSIM se permite la visualización de los resultados de cada bloque. La tabla 2 agrupa los resultados obtenidos para los dos bloques.

TABLA 1.
RESULTADOS PARA LA MEDICIÓN Y CÁLCULO DE NIVEL Y VOLUMEN EN LADDER Y SIMULINK PLC CODER.

Valor Sensor	Nivel FB1 (m)	Volumen FB1 (m ³)	Nivel FB2 (m)	Nivel FB2 (m ³)
-27648	0.0	0.00	0.0	0.00
-25805	0.5	0.65	0.5	0.65
-23962	1.0	2.09	1.0	2.09
-20275	2.0	5.24	2.0	5.24
-16589	3.0	8.38	3.0	8.38
-12902	4.0	11.5	4.0	11.5
-9216	5.0	14.6	5.0	14.6
-5530	6.0	17.8	6.0	17.8
-1843	7.0	20.9	7.0	20.9
0	7.5	22.4	7.5	22.4
1843	8.0	23.0	8.0	23.0

Se observa que en un nivel de precisión de 2 cifras, no existen diferencias entre los valores obtenidos por el bloque FB1 y FB2.

desde las propiedades de cada bloque de función el uso de memoria de carga (datos) y memoria de trabajo (programa) requerida. Además el parámetro MC7 indica la longitud del código. La Tabla 2 presenta los resultados obtenidos.

D. Comparación de los resultados

Una vez implementado los dos códigos (Ladder e IEC 61131-3) en SIMATIC Manager, es posible observar

TABLA 2.
COMPARACIÓN DE USO DE MEMORIAS DE LOS BLOQUES FB.

	MC7 (bytes)	Memoria de Carga	Memoria de Trabajo
FB1 (SCL de Simulink PLC Coder)	668 B	814 B	704 B
FB2 (Ladder)	898 B	1078 B	934 B

El bloque FB1 realizado con MATLAB® y la herramienta Simulink® PLC Coder, genera un código IEC 61131-3 que emplea menos recursos que el bloque de Función FB2 implementado con Ladder.

La comparación relativa de estos resultados arroja la siguiente información.

TABLA 3.
COMPARACIÓN RELATIVA DE SIMULINK® PLC CODER VS. LADDER.

	MC7	Memoria de Carga	Memoria de Trabajo
FB1 (SCL de Simulink PLC Coder)	-25%	-24.4%	-24.6%

La Tabla 3 permite identificar que la implementación de un programa en Simulink® PLC Coder y el código

IEC 61131-3 generado resulta un 25% más eficiente

que la solución del mismo problema mediante Lenguaje Ladder.

VI. CONCLUSIONES

En este artículo se presenta la creación de un bloque de función FB que calcula el volumen y nivel de un tanque empleando dos herramientas de programación, Lenguaje LADDER y Simulink® PLC Coder de MATLAB®. Este último, facilita la inserción de una notación matemática que represente un proceso industrial.

El software de cálculo matemático MATLAB® y la herramienta Simulink® PLC Coder, constituyen una solución adecuada para el modelamiento y cálculo de variables en procesos industriales controlados por Autómatas Programables.

La generación de código IEC 61131-3 mediante Simulink® PLC Coder permite la programación de un Autómata Programable o PLC, y además, reduce el tiempo de desarrollo de programación, el tamaño de memoria de carga, de memoria de trabajo y la longitud de código MC7 del programa.

Como trabajo futuro se sugiere llevar a cabo implementaciones de problemas y modelos matemáticos de procesos industriales más complejos que permitan realizar otro tipo de comparaciones con Simulink® PLC Coder y así llegar a resultados globales.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Y.-C. Wu y C.-F. Fan, «Automatic test case generation for structural testing of function block diagrams», *Inf. Softw. Technol.*, vol. 56 (10), pp. 1360-1376, oct. 2014.
- [2] A. Mader, «A Classification of PLC Models and Applications», en *Discrete Event Systems*, vol. 569, R. Boel y G. Stremersch, Eds. Springer US, 2000, pp. 239-246.
- [3] L. Wang, G. Adamson, M. Holm, y P. Moore, «A review of function blocks for process planning and control of manufacturing equipment», *J. Manuf. Syst.*, vol. 31 (3), pp. 269-279, 2012.
- [4] International Electrotechnical Commission, «IEC 61499-1: Function Blocks - Part 1 Architecture», *Int. Stand. First Ed.* Geneva, vol. 1, 2005.
- [5] I. E. Commission, «Programmable Controllers - Part 3: Programming languages», vol. 3, 2003.
- [6] N. Völker y B. J. Krämer, «Automated verification of function block-based industrial control systems», *Sci. Comput. Program.*, vol. 42 (1), pp. 101-113, 2002.
- [7] C. Yuan y P. Ferreira, «An integrated rapid prototyping environment for reconfigurable manufacturing systems», en *ASME 2003 International Mechanical Engineering Congress and Exposition*, 2003, pp. 737-744.
- [8] C. Yuan y P. Ferreira, «An integrated environment for the design and control of deadlock - free flexible manufacturing cells», en *ASME 2004 International Mechanical Engineering Congress and Exposition*, 2004, pp. 471-481.
- [9] K. Thramboulidis y C. Tranoris, «An architecture for the development of function block oriented engineering support systems», en *Computational Intelligence in Robotics and Automation*, 2001. *Proceedings 2001 IEEE International Symposium on*, 2001, pp. 536-542.
- [10] C. Yang y V. Vyatkin, «Transformation of Simulink models to IEC 61499 Function Blocks for verification of distributed control systems», *Control Eng. Pract.*, vol. 20 (12), pp. 1259-1269, dic. 2012.
- [11] V. Vyatkin, *IEC 61499 function blocks for embedded and distributed control systems design*. ISA-Instrumentation, Systems, and Automation Society, 2007.
- [12] A. D. Kshemkalyani y M. Singhal, *Distributed computing: principles, algorithms, and systems*. Cambridge University Press, 2008.
- [13] J. Peltola, J. Christensen, S. Sierla, y K. Koskinen, «A migration path to IEC 61499 for the batch process industry», en *Industrial Informatics, 2007 5th IEEE International Conference on*, 2007, vol. 2, pp. 811-816.
- [14] MathWorks, «Simulink PLC Coder». [En línea]. Disponible en: <http://www.mathworks.com/products/sl-plc-coder/>. [Accedido: 02-dic-2014].
- [15] M. Bonfè, C. Fantuzzi, y C. Secchi, «Design patterns for model-based automation software design and implementation», *Control Eng. Pract.*, vol. 21 (11), pp. 1608-1619, 2013.

- [16] C. Yang y V. Vyatkin, «Model transformation between MATLAB simulink and Function Blocks», en Industrial Informatics (INDIN), 2010 8th IEEE International Conference on, 2010, pp. 1130-1135.
- [17] C. H. Yang y V. Vyatkin, «Automated Model Transformation between MATLAB Simulink/ Stateflow and Function Blocks», en 13th IFAC Symposium on Information and Control in Manufacturing (INCOM'09), Moscow, 2009, vol. 13, pp. 205-210.
- [18] Siemens, «Structured Control Language (SCL) for S7-300/S7-400 Programming». [En línea]. Disponible en: https://cache.automation.siemens.com/dnl/Dc1NzU5OQAA_1137188_HB/SCLV4_e.pdf. [Accedido: 02-dic-2014].
- [19] L. R. Peñalver, D. G. Arroniz, A. R. Canales, M. J. Buendía, y J. M. Martínez, «Sistema de automatización y telecontrol mediante microcontroladores para la gestión del riego de cultivos en maceta con lisímetros de pesada.», presentado en VII Congreso Ibérico de Agroingeniería y Ciencias Hortícolas, Madrid, 2013, p. 52.
- [20] H. Berger, Automating with SIMATIC: Controllers, Software, Programming, Data. John Wiley & Sons, 2012.