

Simulation Tool for Hybrid AGVs based on IEC-61131

V. Ruiz and J.E. Sierra-García

Abstract—AGVs or automatic guided vehicles are unmanned battery-powered trucks used to transport loads in logistics and production. They are equipped with a controller, normally an industrial programmable logic controller (PLC), that processes the information of the guiding sensor and computes the signals to control the motors with the aim of following the desired trajectory. The tracking error, the traveled distance, and the time to execute a task depend on the parameters of the controller. Indeed, parameters that work in a circuit may not work in a different one. Thus, the fine-tuning of the controller is a key factor. In order to improve this process, in this work, an IEC61131-based simulation tool has been developed to evaluate the performance of the controller with different trajectories and search the configuration that provides the minimum guide error. The tool can run on a computer or on the AGV's PLC. The simulation tool can execute the same code that is running in the AGV.

Index Terms— AGVs, IEC-61131, PLC, simulation tool, control, fine-tuning.

I. INTRODUCCIÓN

LA robótica móvil es una rama de la robótica y una línea de investigación en continua evolución centrada principalmente en el control de robots autónomos [1]. Dentro de esta tipología de robots se encuentran los AGV, siglas tomadas del inglés Automatic Guided Vehicles.

Los AGV son vehículos de transporte alimentados por baterías que utilizan diferentes sistemas de guiado para trasladarse de forma autónoma. Se emplean comúnmente en aplicaciones industriales en sustitución de las carretillas manuales y vehículos industriales tripulados [2]. Han tenido una gran aceptación en la industria y su uso sigue en aumento debido a la creciente tendencia a la digitalización y automatización de procesos. Los AGV mejoran los procesos productivos y logísticos aportando mayor flexibilidad que las soluciones de transporte tradicionales.

Los AGV están equipados con un controlador, normalmente un PLC, que recibe la información del sensor de guiado, toma las decisiones de control apropiadas y genera las señales de control de los motores necesarias para seguir las rutas de transporte establecidas minimizando el error, el tiempo empleado y la distancia recorrida. El error cometido al seguir la trayectoria y el tiempo empleado dependen en gran medida de los parámetros de ajuste del controlador.

V. Ruiz, Universidad de Burgos, Burgos, España (e-mail: vrml1001@alu.ubu.es).

J. E. Sierra-García, Departamento de Ingeniería Electromecánica, Área de Ingeniería de Sistemas y Automática, Universidad de Burgos, Burgos, España (email: jesierra@ubu.es).

Unos parámetros que funcionan bien para una trayectoria puede que no funcionen para otra, por lo que realizar una sintonía adecuada del controlador es fundamental. Para facilitar este trabajo se ha desarrollado una herramienta que permite evaluar el rendimiento del controlador con diferentes trayectorias y encontrar la combinación de parámetros que proporcionan un menor error de guiado. Esto permite reducir los tiempos de puesta en marcha de estos sistemas y mejorar su rendimiento.

El primer artículo sobre el diseño y funcionamiento de sistemas AGV fue elaborado por Maxwell y Muckstadt en el año 1982 [4], desde entonces la investigación en este campo no se ha detenido. Habitualmente estos estudios centran su interés en subsistemas específicos de los AGV. La mayoría están enfocados en cinemática y control, sin embargo, otros se orientan hacia los algoritmos de localización, almacenamiento de energía y sistemas de carga.

Veiga et al. presentan un modelo multicuerpo de un AGV desarrollado en Modelica [5]. En [6], se muestra el análisis cinemático y dinámico de un robot móvil. En [7] se presenta el proceso de optimización de un sistema de suspensión para un robot móvil. En [8] se comparan diferentes técnicas de control para la realización de maniobras con un AGV. En [9] se propone un controlador en modo deslizante para el seguimiento de trayectorias de un AGV. En [10] se propone una estrategia de control de tracción para un vehículo eléctrico de cuatro ruedas basada en observadores no lineales. En [11] se presenta una arquitectura funcional para la descripción de AGV industriales. En [12] se estudia el efecto de la latencia en el control remoto de AGV.

Un extenso estado del arte sobre tecnologías de localización de AGV se presenta en [13]. Biswas desarrolla un sistema de localización de robots móviles para entornos en interior basado en cámara de profundidad [14]. En [15] se desarrollan unidades de sensorización basadas en infrarrojos con bajo tiempo de respuesta para controlar robots móviles de forma remota. Un sistema de localización de robots móviles autónomos basado en métodos de aprendizaje máquina y sonar omnidireccional se presenta en [16]. Un algoritmo de planificación de rutas probabilístico para múltiples robots con incertidumbre en el movimiento se describe en [17]. En [18] proponen un índice de calidad para evaluar datos supervisados para localización de AGV basada en redes neuronales convolucionales.

Este trabajo se centra en el diseño y desarrollo de una herramienta de simulación de AGV basada en IEC-61131 que nos permita evaluar el rendimiento del controlador ante diferentes trayectorias e identifique la combinación de

parámetros que minimiza el error de seguimiento ante una trayectoria dada. La implementación siguiendo el estándar IEC61131 nos permite correr la herramienta en un PLC virtual o en un PLC real. Además, el código empleado en el controlador puede ser exactamente el mismo que se ejecuta en el AGV real.

Actualmente con la herramienta se pueden simular AGV de tipo diferencial o AGV híbridos que combinan un triciclo y un diferencial. Siendo estos últimos de gran aceptación en la industria. La distancia entre las ruedas y la distancia del sensor al eje de las ruedas, es decir los dos parámetros que determinan el comportamiento del AGV, son configurables, esto permite realizar simulaciones con diferentes AGV. Por ello se da la posibilidad al usuario de introducir diferentes parámetros en una pantalla HMI. Por otro lado, la herramienta se ha desarrollado de forma modular, de esta forma se puede ampliar fácilmente a un AGV de tipo triciclo con tan solo cambiar uno de los módulos sin alterar el resto del código.

Existen otras alternativas para realizar el ajuste del controlador como Matlab/Simulink. Sin embargo, tienen la desventaja de ejecutar el controlador simulado en un entorno diferente al entorno de ejecución del controlador real. Por lo que los resultados obtenidos en simulación pueden ser diferentes que los resultados obtenidos cuando se implementa en el controlador real.

Para resolver este problema, en este trabajo la herramienta de simulación se ha implementado en un lenguaje de programación de autómatas bajo la norma IEC-61131. El estándar IEC-61131 es un conjunto de normas publicado por la Comisión Electrotécnica Internacional con el objetivo de armonizar los lenguajes de programación de los autómatas programables [3]. Actualmente está soportado por la mayoría de los PLC. De esta forma la herramienta desarrollada puede correr en un PLC virtual en un ordenador, o puede correr en el PLC real del AGV. Además, nos permite ejecutar exactamente el mismo programa de control que está corriendo en el PLC del AGV minimizando así las diferencias entre los resultados obtenidos en simulación y en el entorno real.

En conclusión, las principales contribuciones de este trabajo son las siguientes:

- Desarrollo de una herramienta modular para la simulación de AGV híbridos.
- Implementación de la herramienta en el standard de programación de autómatas IEC-61131.
- Diseño de la herramienta para que pueda ejecutarse en un ordenador o directamente en el autómata.
- Búsqueda de la configuración de constantes del controlador que minimiza el error de guiado.

La estructura del artículo es la siguiente. En la sección II se describe el modelo cinemático y el diagrama de control del sistema. En la sección III se detalla el algoritmo de simulación implementado en la herramienta. A continuación, se describe la herramienta de simulación en la sección IV. Acto seguido, los resultados de los diferentes ensayos son analizados en la sección V. El documento finaliza con las conclusiones y líneas de trabajo futuras.

II. DESCRIPCIÓN DEL SISTEMA

En términos cinemáticos, un AGV híbrido es un robot móvil cuya configuración cinemática es una combinación de una configuración diferencial y una de tipo triciclo. La cabeza tractora constituye un AGV diferencial, y el cuerpo se comporta como un triciclo unido a la unidad de tracción por un eje sobre el cual pivota (Fig. 1) [11-12, 19].

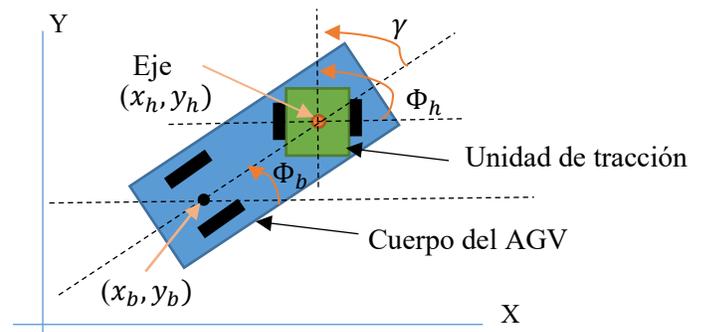


Fig. 1. Cinemática del AGV híbrido.

El sensor de guiado puede variar en función del tipo de referencia que se emplee: magnético, óptico, filoguiado, etc. Esta referencia se coloca físicamente en el entorno y depende del sensor: una línea pintada en el suelo, una banda magnética colocada en el suelo, etc. Normalmente en estos AGV el sensor se coloca en la unidad de tracción o cabeza tractora. De esta forma para estudiar el seguimiento de la referencia podemos centrarnos en el movimiento que describe la unidad de tracción.

El movimiento de la cabeza tractora viene definido por las ecuaciones de un robot diferencial (1-3).

$$\dot{x}_h = \frac{v_L + v_R}{2} \cdot \cos(\Phi_h) \quad (1)$$

$$\dot{y}_h = \frac{v_L + v_R}{2} \cdot \sin(\Phi_h) \quad (2)$$

$$\dot{\Phi}_h = \frac{v_R - v_L}{L_h} \quad (3)$$

Donde \dot{x}_h , \dot{y}_h y $\dot{\Phi}_h$ son las componentes de velocidad x e y (m/s) y la velocidad angular (rad/s) del robot diferencial. v_L y v_R son las velocidades longitudinales de la rueda izquierda y derecha respectivamente (m/s). L_h es la distancia entre el centro de cada rueda (m) y finalmente Φ_h es la orientación en un sistema de referencia inercial (rad).

Como se puede apreciar en (1-3) si la velocidad de la rueda derecha es mayor que la de la izquierda la unidad de tracción gira hacia la izquierda y el ángulo Φ_h crece. En el caso opuesto gira hacia la derecha y el ángulo Φ_h decrece.

Las ruedas están sujetas a restricciones no holonómicas, es

decir, el AGV no es controlable en determinadas direcciones de forma instantánea, al no permitir el deslizamiento lateral del mismo.

La Fig. 2 presenta una aproximación al diagrama de control del sistema. Se puede observar cómo el robot recibe del controlador las consignas de velocidad de las ruedas izquierda y derecha (v_L, v_R). A través del modelo cinemático directo (MCD) estas velocidades de las ruedas se transforman a velocidades en coordenadas cartesianas [20]. En función de la posición del robot (x_h, y_h, Φ_h), y la trayectoria prefijada en el suelo a través de la referencia, se obtiene el error de guiado, err_{gui} , esto es la distancia de la unidad de tracción a la referencia. El error es medido por el sensor de guiado, err_m , y se lleva al controlador. El objetivo del controlador es mantener el error de guiado lo más próximo a cero.

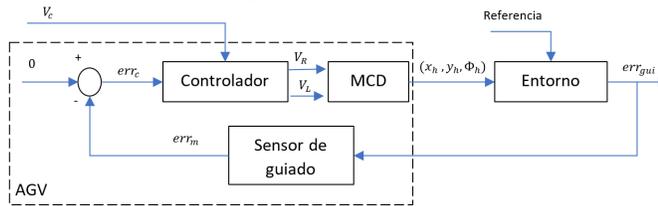


Fig. 2. Diagrama de control del sistema.

III. DESCRIPCIÓN DEL ALGORITMO DE SIMULACIÓN

La herramienta ejecuta un algoritmo que simula el comportamiento del AGV al seguir una trayectoria elegida por el usuario. Este algoritmo de simulación se ejecuta para diferentes valores de las constantes del PID. El usuario establece el rango de las constantes del PID que serán evaluadas y el intervalo, y la herramienta establece todas las combinaciones posibles. Al finalizar cada simulación se calculan ciertas métricas que nos informan del rendimiento del controlador para cada combinación de las constantes del controlador: el MSE; la distancia total recorrida y el tiempo empleado.

Este algoritmo debe ser capaz de simular todos los bloques de la Fig. 2. Debe calcular el error de guiado a partir de la posición y la referencia, debe calcular las consignas de las ruedas en función del error de guiado, y también debe actualizar la posición del robot, en función de las velocidades de las ruedas.

El diagrama de flujo del algoritmo de simulación implementado en la herramienta se muestra en la Fig. 3. Primero se actualiza la posición del robot. A partir de los puntos en el sistema de coordenadas del robot se calculan los puntos en el sistema inercial S0. Con estos puntos se calcula la proyección de la recta que describe el sensor de guiado de la cabeza tractora. A partir de esta recta se calcula el punto de intersección entre la proyección del sensor y la referencia. Este punto de intersección se emplea para calcular el error de guiado. A partir del error de guiado se obtiene la referencia de velocidad angular. Empleando la consigna de velocidad angular y la consigna de velocidad longitudinal se calculan las consignas de velocidad de las ruedas. La frecuencia de ejecución de cada ciclo de trabajo es ajustable. En nuestro caso se ha fijado a 10ms

porque es una frecuencia que se utiliza típicamente en este tipo de AGV.

Seguidamente se expone de forma detallada los cálculos mostrados en el diagrama de flujo.

1) Cálculo de posición actual del robot

En cada ciclo de trabajo se realiza la actualización de las variables de estado de posición y orientación del AGV (4-6) [21]. Estas variables se obtienen integrando las ecuaciones diferenciales (1-3) y se encargan de describir el estado del sistema durante la simulación.

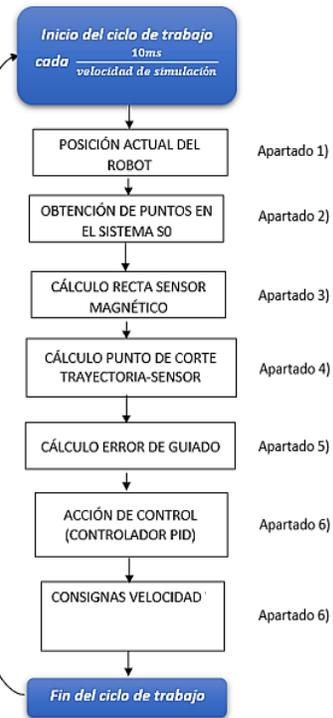


Fig. 3. Diagrama de flujo: Algoritmo de simulación implementado.

$$x_{h_i} = x_{h_{i-1}} + \frac{v_{L_{i-1}} + v_{R_{i-1}}}{2} \cdot \cos(\Phi_{h_{i-1}}) \cdot \Delta t \quad (4)$$

$$y_{h_i} = y_{h_{i-1}} + \frac{v_{L_{i-1}} + v_{R_{i-1}}}{2} \cdot \sin(\Phi_{h_{i-1}}) \cdot \Delta t \quad (5)$$

$$\Phi_{h_i} = \Phi_{h_{i-1}} + \frac{v_{R_{i-1}} - v_{L_{i-1}}}{L_h} \cdot \Delta t \quad (6)$$

Donde x_h e y_h son las coordenadas x e y del centro de la cabeza tractora (m) y Δt es el incremento de tiempo (s).

La herramienta se ha desarrollado de forma modular, uno de los módulos es el encargado de hacer la actualización de la posición del robot, por lo que si se necesitara simular un AGV de tipo triciclo las ecuaciones (4-6) podrían cambiarse por las

ecuaciones de un triciclo simplemente cambiando este módulo sin alterar el resto del código [11].

2) Conversión entre sistemas de referencia

Para el desarrollo del algoritmo el punto de partida es la traslación de los puntos del sistema S_1 (sistema de referencia en el AGV) al sistema S_0 (sistema de referencia inercial), ver Fig. 4.

De este modo, se parte de los puntos P_1 (esquina superior izquierda), P_2 (esquina superior derecha) y P_3 (punto medio superior). Los mismos se pueden apreciar en la Fig. 5 del sistema S_1 del AGV y sus valores en forma matricial en (7).

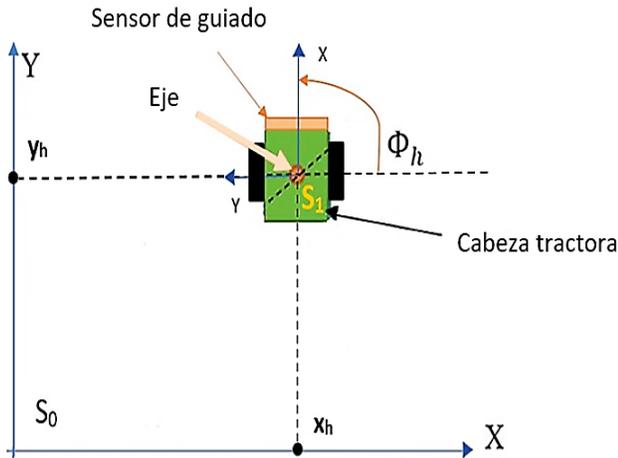


Fig. 4. Sistemas de referencia.

$$P1_{S_1} = \begin{pmatrix} L_h/2 \\ L_h/2 \\ 1 \end{pmatrix}; P2_{S_1} = \begin{pmatrix} L_h/2 \\ -L_h/2 \\ 1 \end{pmatrix}; P3_{S_1} = \begin{pmatrix} L_h/2 \\ 0 \\ 1 \end{pmatrix} \quad (7)$$

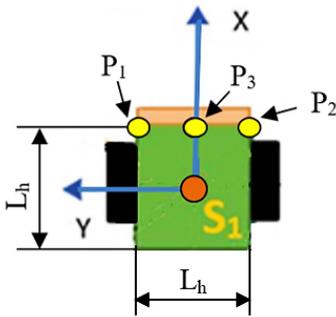


Fig. 5. Puntos a considerar en la conversión entre sistemas.

Se obtienen los respectivos puntos referenciados al sistema S_0 haciendo uso de la ecuación (8)

$$P_{S_0} = M_{transformación} \cdot P_{S_1} \quad (8)$$

Donde P_{S_0} es el punto en el sistema S_0 , $M_{transformación}$ es la matriz de transformación para obtener el cambio entre los sistemas de referencia y P_{S_1} es el punto en el sistema S_1 .

Con respecto a la matriz de transformación [22], se realiza el cálculo (9) sabiendo que se produce una traslación (teniendo en cuenta x_h e y_h del AGV) y una rotación entre los sistemas (teniendo en cuenta Φ_h).

$$M_{transformación} = T(x_h, y_h) \cdot R(\Phi_h) = \begin{pmatrix} 1 & 0 & x_h \\ 0 & 1 & y_h \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos(\Phi_h) & -\sin(\Phi_h) & 0 \\ \sin(\Phi_h) & \cos(\Phi_h) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (9)$$

Donde $T(x_h, y_h)$ es la matriz de traslación y $R(\Phi_h)$ es la matriz de rotación.

3) Recta descrita por la proyección del sensor

Para obtener la recta descrita por la proyección del sensor de guiado (10), es preciso obtener un mínimo de dos puntos. En nuestro caso, se realiza con los puntos P_1 y P_2 (esquinas superior e inferior respectivamente) del sistema S_0 . Con estos puntos se representa la recta descrita por el sensor mediante la expresión (10).

$$y = mx + b = \frac{P_1 \cdot y - P_2 \cdot y}{P_1 \cdot x - P_2 \cdot x} x + \frac{P_2 \cdot y \cdot P_1 \cdot x - P_1 \cdot y \cdot P_2 \cdot x}{P_1 \cdot x - P_2 \cdot x} \quad (10)$$

Donde m es la pendiente de la recta y b es el punto de corte en el eje Y.

4) Puntos de corte

Seguidamente, se calculan las intersecciones o puntos de corte entre las trayectorias prefijadas (11-13) y la recta que está describiendo el sensor de guiado (10). Se han establecido cuatro tipos de trayectorias líneas rectas, circunferencias, elipses y polilíneas. La combinación de estas trayectorias permite implementar la mayoría de los layouts empleados en las aplicaciones industriales.

En primer lugar, si la trayectoria a seguir es una línea recta se parte de las dos ecuaciones características de dos rectas. Una de ellas es la recta descrita por la proyección del sensor y la otra, la definida por la trayectoria.

$$\begin{cases} y = m_1 x + b_1 \\ y = m_2 x + b_2 \end{cases} \quad (11)$$

En el caso de la circunferencia para el cálculo de las intersecciones se considera la ecuación canónica de la misma (12).

$$(x - h)^2 + (y - k)^2 = r^2 \quad (12)$$

Donde h es la coordenada x del centro de la circunferencia (m), k es la coordenada y del centro de la circunferencia (m) y r es el radio de la circunferencia (m).

En el caso de la elipse se examinan dos posibles escenarios ya que esta puede ser horizontal (13.a) o vertical (13.b).

$$\begin{cases} \frac{(x-h)^2}{a^2} + \frac{(y-k)^2}{b^2} = 1 & a) \\ \frac{(x-h)^2}{b^2} + \frac{(y-k)^2}{a^2} = 1 & b) \end{cases} \quad (13)$$

Donde h es la coordenada x del centro de la elipse (m), k es la coordenada y del centro de la elipse (m), a es el semieje mayor de la elipse (m) y b es el semieje menor de la elipse (m).

Por último, en lo referente a la polilínea, al tratarse esta trayectoria de un conjunto de líneas rectas, para cada uno de los segmentos se consideran las restricciones planteadas previamente en el caso de la línea recta.

5) Error de guiado

Con respecto al error de guiado (Fig. 6), se obtiene considerando el punto de corte de la recta descrita por el sensor de guiado y la trayectoria. Para hallar el mismo (14), se toma como referencia el punto de corte con respecto al punto P_3 (punto medio superior).

$$err_{gui} = \sqrt{(x_{PC} - x_{P_3})^2 + (y_{PC} - y_{P_3})^2} \quad (14)$$

Donde x_{PC} es la coordenada x del punto de corte (m), y_{PC} es la coordenada y del punto de corte (m) en el sistema S_0 , y err_{gui} es el error de guiado del AGV (m).

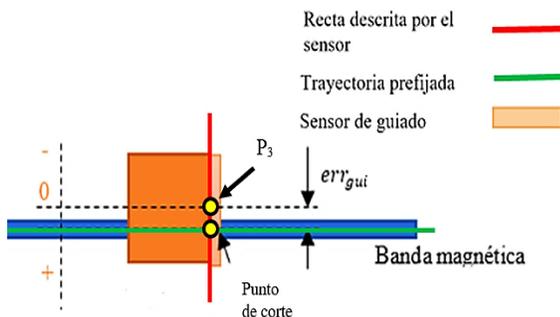


Fig. 6. Error de guiado.

Se fija un criterio para establecer si el error es positivo o negativo, teniendo en cuenta las distancias d_3 (distancia entre el punto P_1 hasta el punto de corte) y d_2 (distancia entre el punto P_2 hasta el punto de corte). Esto se representa en la Fig. 7.

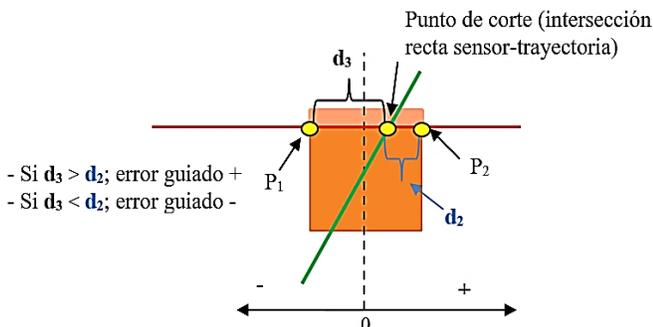


Fig. 7. Criterio de error de guiado.

De igual modo, se determina hacia donde tiene que girar el AGV estableciendo un criterio en función del error de guiado. Es decir, si el error es positivo el sentido de giro será horario y antihorario en caso contrario.

6) Cálculo de la acción de control

El controlador se encarga de regular y adecuar la respuesta del AGV en función de la desviación respecto de la referencia, y por tanto su objetivo es minimizar las oscilaciones que se puedan originar y la distancia total recorrida. Se ha implementado un controlador PID por ser el más utilizado en este tipo de AGV, aunque se podría haber escogido una estrategia de control más avanzada.

De modo que, para controlar el movimiento de la cabeza tractora se emplean las señales de control V_c y w_c . La velocidad de traslación de referencia V_c viene dada por la aplicación, mientras que la velocidad angular de consigna w_c se ajusta mediante el controlador PID para corregir la desviación del AGV. Por ello, lo primero es el cálculo de w_c (15).

$$w_c = K_P \cdot err_c + K_D \cdot \dot{err}_c + K_I \cdot \int err_c dt \quad (15)$$

Donde w_c es la velocidad angular de consigna del AGV (rad/s), K_P es la constante de control proporcional, K_D es la constante de control derivativo y K_I es la constante de control integral. El usuario establece el rango de las constantes del PID que serán evaluadas.

En la herramienta se ha implementado un controlador PID porque es uno de los controladores más utilizados en la industria y es el que se utiliza normalmente en este tipo de soluciones. Sin embargo, la herramienta se ha diseñado expresamente para que el controlador pueda sustituirse fácilmente por otro controlador más avanzado.

7) Consignas de velocidad

A partir de lo descrito anteriormente, se calculan las velocidades de consigna de la rueda izquierda (V_{Lc}) y derecha (V_{Rc}), mediante las expresiones (16-17) [12].

$$V_{Lc} = \frac{2V_c - w_c \cdot L_h}{2} \quad (16)$$

$$V_{Rc} = \frac{2V_c + w_c \cdot L_h}{2} \quad (17)$$

Donde V_c es la velocidad lineal de consigna del robot (m/s), V_{Lc} es la velocidad de consigna de la rueda izquierda (m/s) y V_{Rc} es la velocidad de consigna de la rueda derecha (m/s).

IV. DESARROLLO DE LA HERRAMIENTA EN IEC 61131

A. Entorno de desarrollo de software

El entorno de software elegido es *TwinCAT* versión 3 [23]. Asimismo, se emplean las funciones integradas de HMI y Measurement de este software. Mediante TWINCAT podemos convertir cualquier PC compatible en un controlador en tiempo real con un sistema multi-PLC, posibilitando el desarrollo eficaz de interfaces hombre-máquina, así como la posibilidad de representación gráfica de variables con unos tiempos de muestreo fijados por el usuario. Dentro de sus características cabe destacar el soporte a la programación en el estándar IEC 61131-3.

Por lo tanto, este entorno de desarrollo de software nos permite simular en tiempo real el comportamiento del PLC que controla al AGV, obteniendo un funcionamiento análogo a la implementación de la solución planteada en un PLC real.

La pantalla principal o *Desktop.view* es el punto de partida para poder visualizar los diferentes contenidos presentes en la herramienta. Posee los elementos de control necesarios para arrancar y parar el proceso, así como los controles para la introducción y visualización de datos previos a la simulación. Esta pantalla principal se representa en la Fig. 8.

Esta pantalla tiene una zona central (recuadro naranja en Fig. 8) en la que se muestra diferente información a elección del usuario: la selección del perfil de velocidad, la evolución de los KPI del proceso y las variables de estado del sistema.

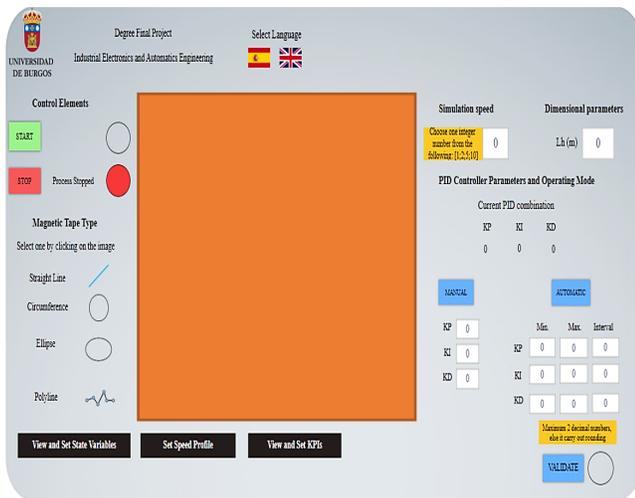


Fig. 8. Pantalla principal o *Desktop.view*.

Uno de los contenidos destacados, es la lista de indicadores clave de rendimiento o KPI. En esta pantalla se permite al usuario fijar el número de vueltas a ejecutar por el robot y visualizar cuantas vueltas ha efectuado en el caso de una trayectoria cerrada, así como el MSE, el RMSE, el error medio, el tiempo transcurrido y la distancia recorrida por vuelta. Asimismo, si la trayectoria es una polilínea es posible ver los puntos alcanzados en la simulación. Esta pantalla se muestra en la Fig. 9.

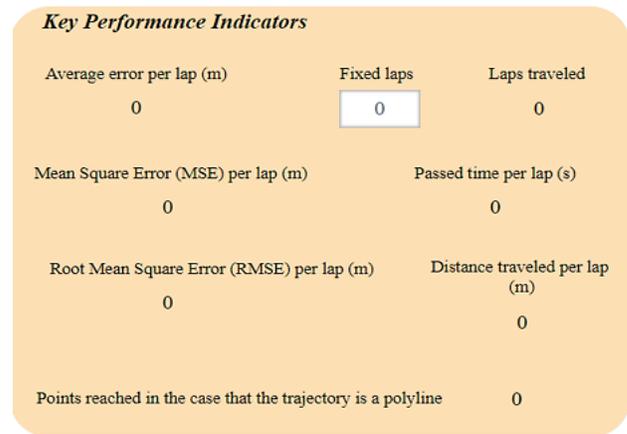


Fig. 9. Contenido de KPI.

B. Configuración de la interfaz implementada

En la Fig. 10 se muestran las opciones que puede configurar el usuario en la interfaz.

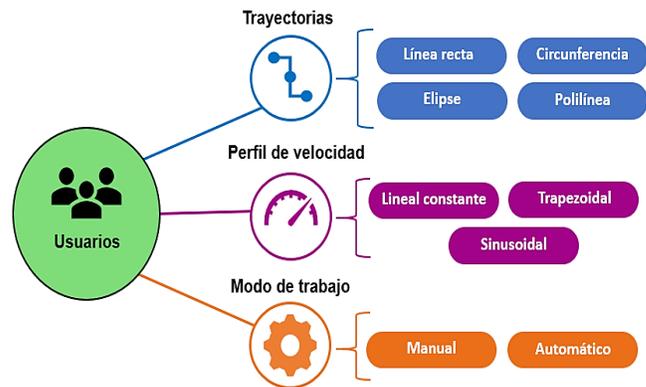


Fig. 10. Configuración de la interfaz HMI.

1) Trayectoria a ejecutar

Es posible simular cuatro tipos de trayectoria prefijada: línea recta, circunferencia, elipse y polilínea.

2) Perfil de velocidad

Se han implementado diferentes tipos de perfiles de velocidad: constante, trapezoidal y sinusoidal. Todos ellos, servirán para evaluar el comportamiento del AGV ante diferentes escenarios de simulación.

- **Constante:** En ejecución la velocidad se mantiene constante a no ser que el usuario presione el pulsador de paro de emergencia, en ese caso se frena el AGV, cayendo la velocidad a 0 m/s.
- **Trapezoidal:** La velocidad aumenta siguiendo una pendiente de aceleración (m_{ac}) desde el inicio de la ejecución hasta que alcanza un valor estable y se mantiene en el mismo hasta que el usuario presiona el pulsador de paro o se alcanzan las vueltas prefijadas en simulación. En este momento la velocidad empezará a disminuir

progresivamente siguiendo la pendiente de frenado (m_d) hasta llegar a 0 m/s (18).

$$\begin{cases} V_{c_i} = V_{c_{i-1}} + (m_{ac} \cdot \Delta t) & V_{c_{i-1}} < v_f \wedge p_{off} = 0 \\ V_{c_i} = v_f & V_{c_{i-1}} \geq v_f \wedge p_{off} = 0 \\ V_{c_i} = V_c - (m_d \cdot \Delta t) & p_{off} = 1 \end{cases} \quad (18)$$

- **Sinusoidal:** Viene definido por una componente continua (B) y una alterna (A, f) para formar una onda sinusoidal (19).

$$V_c(t) = A \cdot \sin(2 \cdot \pi \cdot f \cdot t) + B \quad (19)$$

Donde A es la amplitud de la onda (m/s), B es la componente continua de la onda, es decir, el desplazamiento vertical u offset (m/s), f es la frecuencia (Hz) y t es el tiempo transcurrido (s).

3) Modo de trabajo

En modo manual se establecen unas constantes del controlador (K_P, K_I, K_D) y nos permite evaluar el comportamiento del AGV ante un caso de simulación propuesto. En modo automático nos permite evaluar diferentes combinaciones de parámetros ante el mismo caso de simulación fijado. Al finalizar una de las combinaciones arranca la siguiente desde condiciones iniciales sin tener que seleccionar la misma configuración previa. Finalizada la simulación en modo automático, se muestra por pantalla la configuración con menor error cuadrático medio.

V.RESULTADOS

Para validar el funcionamiento de la herramienta se ha empleado como referencia el modelo Easybot Standard de la empresa ASTI Mobile Robotics, y se han realizado diferentes ensayos en modo manual y en modo automático. En esta sección se muestran algunos de los resultados. Las gráficas representadas se han obtenido mediante el *Scope View* de la herramienta *TwinCAT Measurement* [23].

A. Ensayos en modo manual

En el siguiente experimento, se establecen unas constantes $K_P = 50$, $K_I = 5$ y $K_D = 2$ y un perfil de velocidad sinusoidal con una componente continua y amplitud de 0.7 m/s y una frecuencia de 226 Hz. La trayectoria a seguir en este caso es una elipse vertical de eje mayor 6 m y eje menor 4 m con centro en [0,0] m. Se fijan unas condiciones iniciales de $x_h = 0$ m, $y_h = -3$ m y $\Phi_h = 0$ rad y un valor de $Lh = 0.5$ m.

La Fig. 11 representa el movimiento descrito por el AGV al seguir la trayectoria elíptica. Se puede observar que las características de la elipse coinciden con las prefijadas en la trayectoria. Por otro lado, en la Fig. 12 se muestra la posición y

orientación del AGV, así como el error de guiado. En este caso se aprecia una oscilación mantenida en el error de guiado a lo largo de toda la trayectoria. Esto puede ser debido a que el radio de curvatura no es constante. Sin embargo, el mayor error se produce cuando la componente y llega al máximo y al mínimo.

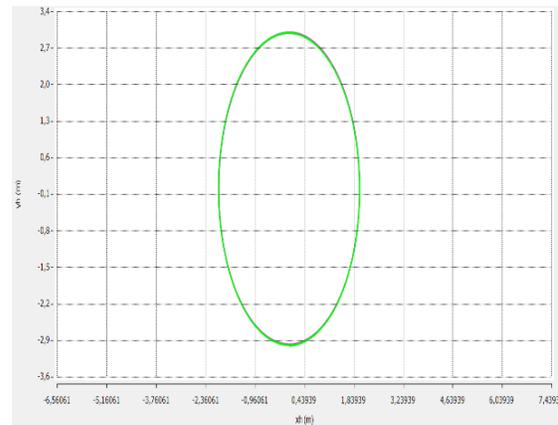


Fig. 11. Movimiento descrito por el AGV cuando sigue una trayectoria elíptica.

B. Ensayos en modo automático

Por último, se establece un perfil de velocidad lineal de $V_c = 0.7$ m/s y un valor de $Lh = 0.5$ m. La trayectoria descrita en este caso es una polilínea con nueve puntos de paso: (6,0), (7,-1), (7,-3), (6,-4), (0,-4), (-1,-3), (-1,-1), (0,0) y (6,0). Se parte de unas condiciones iniciales de $x_h = 6$ m, $y_h = 0$ m y $\Phi_h = 5.3$ rad, de modo que, el punto final de la polilínea coincide con el inicial por lo que el AGV podrá dar vueltas completas. Se fijan tres vueltas para evaluar el rendimiento del controlador.

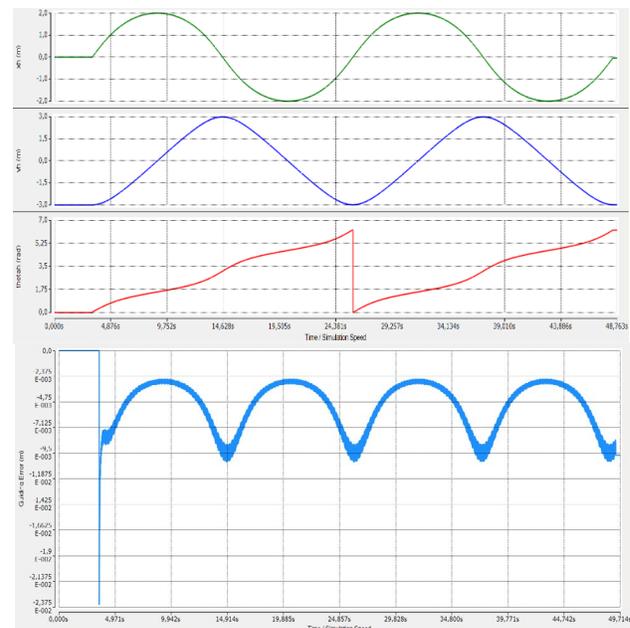


Fig. 12. Variables de posición, orientación y error de guiado en la trayectoria elíptica.

De acuerdo con este modo de funcionamiento, se introducen los rangos de búsqueda de las constantes del controlador, y el

sistema notifica al final de la ejecución la combinación con la que se obtiene el menor error cuadrático medio, siendo esta [50, 12, 2] (Fig. 13).

	Min.	Max.	Interval
KP	30	50	10
KI	0	12	6
KD	1	3	1

Maximum 2 decimal numbers, else it carry out rounding

VALIDATE

Minor error with combination [50,12,2]

Fig. 13. Rangos de búsqueda de constantes y resultado obtenido.

En la Fig. 14 se representa el movimiento descrito por el AGV cuando la trayectoria descrita es una polilínea. En general el AGV sigue correctamente la trayectoria con alguna oscilación en los extremos de los segmentos.

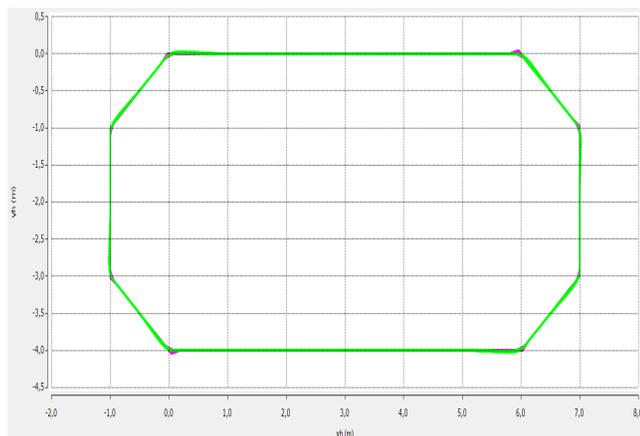


Fig. 14. Movimiento descrito por el AGV cuando sigue una polilínea

Seguidamente, en la Fig. 15 se muestra la evolución de las variables de estado y el error de guiado a lo largo del tiempo. Cuando el AGV pasa por los extremos de los segmentos que definen la polilínea se producen discontinuidades en la curvatura de la trayectoria. Estos cambios bruscos se reflejan en escalones en la orientación y picos en el error de guiado.

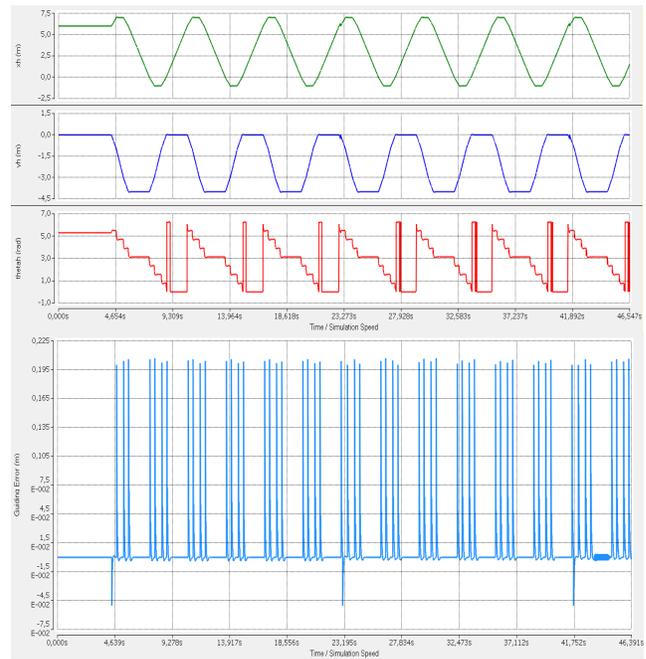


Fig. 15. Variables de posición, orientación y error de guiado cuando el AGV describe una polilínea.

Finalmente, en la Fig. 16 se muestra el error MSE cuando la herramienta realiza la búsqueda de la mejor configuración de parámetros del controlador. Es posible observar como el MSE cambia con cada combinación de constantes y la mejor configuración encontrada es la 26 ($K_p = 50, K_I = 12, K_D = 2$).



Fig. 16. MSE acumulado por vuelta asociado a las combinaciones.

VI. CONCLUSIONES

En este trabajo se ha desarrollado bajo el estándar IEC-61131 una herramienta de simulación que permite evaluar el rendimiento del controlador de un AGV híbrido ante diferentes trayectorias y encontrar la configuración que minimiza el error de guiado. Al implementarla bajo IEC-61131 la herramienta puede correr en un ordenador o en el PLC del AGV. Esto nos ayuda a garantizar que el controlador implementado en la herramienta de simulación ejecuta exactamente el mismo código que el controlador del AGV, minimizando las diferencias entre la simulación y la ejecución real.

La herramienta posee un HMI que permite modificar la configuración del AGV para adaptarse a distintos modelos. Además, permite establecer diferentes perfiles de velocidad y diferentes tipos de trayectorias.

Como líneas de trabajo futuras cabe destacar, la incorporación en el modelo de los componentes eléctricos del AGV: motores, batería y cargador; la inclusión del modelo

cinemático del triciclo, la implementación de saturación y anti-windup en el PID, y el desarrollo de controladores más avanzados basados en lógica difusa y redes neuronales.

REFERENCIAS

- [1] P. Moubarak, and P. Ben-Tzvi, “Modular and reconfigurable mobile robotics,” *Robotics and Autonomous Systems*, vol. 60, no. 12, pp. 1648-1663, 2012.
- [2] F. Espinosa, C. Santos, and J.E. Sierra-García, “Transporte multi-AGV de una carga: estado del arte y propuesta centralizada,” *Revista Iberoamericana de Automática e Informática industrial*, vol. 18, no. 1, pp. 82-91, 2021.
- [3] Programmable controllers, IEC-61131, 2013. [Online]. Available: <https://webstore.iec.ch/publication/4552>
- [4] W.L. Maxwell and J.A. Muckstadt, "Design of automatic guided vehicle systems," *IIE Transactions*, pp. 114-124, 1982.
- [5] J. Veiga, et al. “Modeling of Dynamic Behavior of AGV systems,” in *6th International Conference on Control, Decision and Information Technologies (CoDIT)*, Paris, France, 2019, pp. 1307-1312.
- [6] M. Bonilla, F. Reyes, and M. Mendoza, “Modelling and simulation of a wheeled mobile robot in configuration classical tricycle,”. In *Proceedings of 5th WSEASA International Conference on Instrumentation, Measurement, Control, Circuits and Systems*, Cancún, México, 2005.
- [7] Aguirre, J. A., Gutiérrez-Frías, O., & Sossa-Azuela, H. (2021). Optimization of a Passive Parallelogram Suspension System for a Planetary Rover Using Differential Evolution. *IEEE Latin America Transactions*, 19(8), 1366-1374.
- [8] J. Villagra, and D. Herrero-Pérez, “A comparison of control techniques for robust docking maneuvers of an AGV,” *IEEE Transactions on Control Systems Technology*, vol. 20, no. 4, pp. 1116-1123, 2011.
- [9] Y. Han, Y. Cheng, and G. Xu, “Trajectory tracking control of AGV based on sliding mode control with the improved reaching law,” *IEEE Access*, vol. 7, pp. 20748-20755, 2019.
- [10] D. Aligia, G. Magallán, and C. De Angelo, “Traction control of an electric vehicle based on nonlinear observers,” *Revista Iberoamericana de Automática e Informática industrial*, vol. 15, no. 1, pp. 112-123, 2017.
- [11] J.E. Sierra-García, and M. Santos, “Mechatronic Modelling of Industrial AGVs: A Complex System Architecture,” *Complexity*, vol. 2020, Article ID 6687816, 2020.
- [12] J.E. Sierra García, and M. Santos, “Influencia de la latencia en el control de AGVs a través de redes 5G,” In *XL Jornadas de Automática*, La Coruña, Spain, Universidade da Coruña, Servizo de Publicacións, pp. 611-616, 2019.
- [13] H. Durrant-Whyte, D. Rye, and E. Nebot, “Localization of autonomous guided vehicles,” In *Robotics Research*, Springer, London, pp. 613-625, 1996.
- [14] J. Biswas, and M. Veloso, “Depth camera based indoor mobile robot localization and navigation,” In *2012 IEEE International Conference on Robotics and Automation*, Minnesota, USA, 2012, pp. 1697-1702.
- [15] F. Espinosa Zapata, J. Lázaro Galilea, and J. Olivares Bueno, “Contributions to Optimizing Remote Robot Guidance in Intelligent Spaces,” *Revista Iberoamericana de Automática e Informática industrial*, vol. 15, no. 4, pp. 416-426, 2018.
- [16] J.S. Almeida et al, “Localization system for autonomous mobile robots using machine learning methods and omnidirectional sonar,” *IEEE Latin America Transactions*, vol. 16, no. 2, 872, pp. 368-374, 2018.
- [17] J. Wang, R. Tai, and J. Xu, “A Bi-Level Probabilistic Path Planning Algorithm for Multiple Robots with Motion Uncertainty,” *Complexity*, vol. 2020, Article ID 9207324, 2020.
- [18] S. Ito et al, “Quality Index of Supervised Data for Convolutional Neural Network-Based Localization,” *Applied Sciences*, vol. 9, no. 10, 2019.
- [19] Sanchez, R., Sierra-García, J. E., & Santos, M. (2021). Modelado de un AGV híbrido triciclo-diferencial. *Revista Iberoamericana de Automática e Informática industrial*.
- [20] Peralta, R. M., Gómez, E. Z., & Azuela, J. H. S. (2019). Efficient FPGA hardware implementation for robot manipulator kinematic modeling using rational trigonometry. *IEEE Latin America Transactions*, 17(09), 1524-1536
- [21] G. Oriolo, “Control of Nonholonomic Systems,” University of Rome, 1995. [Online]. Available: https://www.dis.uniroma1.it/~oriolo/cns/cns_slides.pdf
- [22] A. Barrientos, L.F. Peñín, C. Balaguer and R. Aracil, “Fundamentos de Robótica,” 2nd ed, McGraw-Hill, pp. 49-92, 2007.
- [23] Beckhoff Automation TwinCAT 3 Software, 2021. [Online]. Available: https://www.beckhoff.com/media/downloads/information-media/beckhoff_twinCAT3_e.pdf



Víctor Ruiz Molledo nacido en Burgos, España. Es graduado en Ingeniería en Electrónica Industrial y Automática por la Universidad de Burgos (UBU) en el año 2021. Sus principales líneas de investigación son: automatización, PLC, AGVs, modelado y simulación.



Jesús Enrique Sierra García nacido en Burgos, España. Es Ingeniero en Electrónica e Ingeniero de Telecomunicación por la Universidad de Valladolid (UVA), Máster en Ingeniería de Sistemas y Control (UNED), y Doctor en Ingeniería Informática por la Universidad Complutense de Madrid (UCM). Desde 2012 trabaja en la Universidad de Burgos, donde actualmente es profesor del área de Ingeniería de Sistemas y Automática. Es fundador de la unidad mixta de investigación ASTI-UBU en Tecnologías Aplicadas a Vehículos Autónomos, Robótica Móvil y AGV. Ha sido investigador principal en 15 proyectos de I+D relacionados con robótica móvil. Sus principales líneas de investigación son: control inteligente, robótica, AGV, simulación y energía eólica.