Revista Internacional de Investigación e Innovación Tecnológica

Página principal: www.riiit.com.mx

Desarrollo de solucionador basado en gemelos digitales para simulación y control de brazo robótico Dorna®

Development of a digital twin-based solver for simulation and control of the Dorna® robotic arm

Elizalde-Vega, Á.D.[1], Bringas-Posadas, D.[2], Cruz-González, C.E.[3], Guzmán-Flores, I.[4]*, Velázquez-Mora, R.[4]

[1] Posgrado interinstitucional CIDESI. Av. Playa, Av. Pie de la Cuesta 702-No. 702, Desarrollo San Pablo, 76125 Santiago de Querétaro, Qro.
[2] Dirección de Ingeniería Especializada. CIDESI. Av. Playa, Av. Pie de la Cuesta 702-No. 702, Desarrollo San Pablo, 76125 Santiago de Querétaro, Qro. ORCID: https://orcid.org/0000-0002-0371-6384.
[3] CIDESI-CENTA. Tecnologías Aeronáuticas. Aeropuerto internacional de Querétaro. 76278 Colón, Querétaro. ORCID: https://orcid.org/0000-0001-7620-9052.
[4] Facultad de Sistemas UAdeC. Ciudad Universitaria, Fundadores Km 13, Zona Centro, 25350 Arteaga, Coah. ORCID: https://orcid.org/0000-0002-1171-8291*, https://orcid.org/0000-0003-3661.
a.elizalde@posgrado.cidesi.edu.mx; dario.bringas@cidesi.edu.mx; ecruz@cidesi.edu.mx; isidroguzman@uadec.edu.mx*; rvelazquezmora@uadec.edu.mx

Innovación tecnológica: Se está desarrollando un marco de trabajo basado en gemelos digitales para planificar actividades de fabricación y optimizar los procesos de toma de decisiones.

Área de aplicación industrial: Sectores de salud públicos y privados centrados en actividades de fabricación.

Recibido: 23 octubre 2025
Aceptado: 22 abril 2026

Abstract

Globally, manufacturing activities are migrating towards more complex strategies, which include digitalization for real-time decision-making in process and plant operations. In this context, current technology is being revolutionized to generate digital twins (DTs) that allow for the modification of manufacturing activities and their analysis for rapid optimization. This is because these activities have a profound impact on a country' seconomy through associated economic sectors such as logistics, technology development, and new business processes. Plant analysis software is generally expensive and depends on licensing, which is prohibitive for many entities, thereby limiting appropriate decision-making and increasing associated costs. In this regard, this work presents the development of a solver for simulation software for flexible production lines based on DTs. This was achieved through the digitalization of a Dorna® robot, the development of a

PyBullet environment, a graphical interface, and a comparison between the digitalized and physical robots through visual inspection and statistical analysis. The results suggest that a concordance between the physical and digital models occurred, which was verified through observations of the interaction between the physical and digital robots. Additionally, a statistical linear regression analysis was performed, revealing a correlation of approximately 99% between the physical and digital robot movements. This suggests a statistical agreement between the physical and digital robots. Therefore, these results indicate that the models, compared visually and statistically, exhibited strong agreement, suggesting an accurate correlation between the physical and digital phenomena. Finally, the development of a solver for simulation software for flexible production lines based on DTs promotes and encourages the creation of low-cost solutions for analyzing manufacturing systems. This will foster solutions for digital manufacturing in open-source software environments and the creation of cutting-edge technology based on digital manufacturing principles.

Keywords: Manufacturing, Digital twin, Robotic arm, Intelligent system.

Resumen

En la actualidad, las actividades de manufactura están migrando hacia estrategias más complejas que incluyen digitalización para la toma de decisiones en tiempo real. En ese sentido, la tecnología actual se está revolucionando para la generación de gemelos digitales (GD) que permitan la modificación de actividades de manufactura y su análisis para el fortalecimiento del mismo. Esto, debido a que dichas actividades tienen un profundo impacto en la economía de un país debido a que genera actividades económicas como logística, desarrollo de tecnología y nuevos procesos. Los softwares de análisis de planta, son por lo general costosos y dependen de licenciamientos que en muchos entes no son permisibles. En ese sentido este trabajo, presenta el desarrollo de un solucionador para un software de simulación para la línea de producción flexible basado en GDs. Lo anterior, mediante la digitalización de un robot Dorna®, el desarrollo de un ambiente PyBullet, una interfaz gráfica y una comparación entre el robot digitalizado y el físico. Los resultados, sugieren que existe una concordancia entre los modelos físico y el digital que pudo verificarse mediante observaciones de la interacción del robot físico y digital. Adicionalmente, se realizó un análisis estadístico de regresión lineal, cuya correlación entre los movimientos del robot físico y digital fueron de alrededor del 99%. Lo anterior, sugiere una concordancia estadística entre el robot físico y el robot digital. Por lo tanto, estos resultados apuntan que los modelos comparados visual/estadísticamente poseen una buena concordancia y sugieren que existe un valor de correlación correcto entre los fenómenos (físico y digital). Finalmente, el desarrollo de solucionador para un software de simulación para la línea de producción flexible basado en GD, promueve e impulsa la creación de soluciones de bajo costo para analizar sistemas de manufactura. Así, impulsar soluciones sobre manufactura digital en ambientes de software libre y la creación de tecnología de punta.

Palabras clave: Manufactura, Gemelo digital, Brazo Robot, Sistema Inteligente.

1.0 Introducción

La economía, es una ciencia social que estudia como las sociedades, personas, compañías interactúan para que, mediante la gestión de recursos limitados, puedan satisfacer sus necesidades (Secretaría de medio ambiente CDMX, s.f.). Una de las actividades con mayor impacto en la economía de un país es la manufactura, ya que transforma materias primas en productos que generar riqueza, empleo e impulsan la innovación tecnológica (ONUDI, 2021). Tal es así, que esta actividad representa alrededor del 20% del Producto Interno Bruto (PIB) de México e impulsa su crecimiento económico de forma sostenida (STATISTA, 2024).

Como se mencionó en el primer párrafo, el uso de recursos de forma eficiente ha llevado generar técnicas de optimización y la manufactura no es la excepción. La optimización de la manufactura busca la mejora continua de los procesos para aumentar la eficiencia, reducir costos y desperdicios (London Consulting Group, 2024). La optimización de proceso sirve para analizar el flujo de trabajo, materias primas, herramientas, maquinaria, eliminando cuellos de botella; así, mejorar los flujos de trabajo y reducir los ciclos de trabajo (Flores, 2024).

En la actualidad, existen los sistemas de manufactura flexible (SMF) que incrementan la productividad ya que permiten una rápida adaptación a cambios en las demandas y optimizando los de recursos empleados. Por lo tanto, se puede esperar un mayor rendimiento, reducción de costos e incremento en la eficiencia de un proceso u organización (Salvagnini, 2024). Los SMF, incluyen estaciones de trabajo en serie, manipulación automatizada y un sistema de control informático que les permiten una reacción veloz. La agilización de un proceso involucra cambios que puede ser graduales o radicales y la participación de un grupo interdisciplinario, no solo de cambios en reformas de negocio (Kosky, Balmer, Keat, & Wise, 2021).

En la actualidad, existe un mercado valuado en 3900 millones de dólares en el desarrollo de software y se proyecta que crecerá hasta el 2027 a un ritmo de alrededor de 6% anual (Ramirez, 2025). La inversión en softwares en actividades de manufactura puede variar de cientos a miles de dólares y sus aplicaciones van desde captura de datos, monitoreo y optimización de procesos (Salado-Morales, s.f.). Con la aparición de la industria 4.0 (I4.0), el 56% de la industria está implementando técnicas de manufactura inteligente, empleando sistemas baja escala, 20% a gran escala y el resto planificando inversiones (Rockwell Automation, 2025). En ese sentido, se puede vislumbrar que el desarrollo de software en procesos de optimización, manufactura digital usando gemelos digitales, por mencionar algunos, podrían ser fuentes de investigaciones relevantes. A continuación, se muestran algunas que se relacionar con esta temática.

(Colledani, Yemane, Lugaresi, Borzi, & Callegaro, 2018), presentaron el desarrollo de un software para el soporte del diseño y reconfiguración de embalaje. Los autores, utilizaron su plataforma para analizar una empresa que fabricaba válvulas hidráulicas. Lo anterior, basándose en seis productos (con diferentes condiciones de proceso) y escenarios de fluctuación (cambios de demanda, nuevos productos, etc.) para simular cambios de producción. Los resultados, sugieren que los cinco escenarios analizados el que mayor demanda de producto, sin necesidad de reconfiguración a lo largo del horizonte temporal generará una diferencia del 35% del mejor (costo mayor). Por lo tanto, el óptimo no es la configuración que minimiza el costo sino la que se adapta mejor a los cambios previstos.

(Lugaresi, Aglio, Folgheraiter, & Matta, 2019), realizaron una validación de modelos digitales en tiempo real para sistemas de manufactura (SM). Los autores, realizaron un diseño experimental para conocer el efecto de las variables estudiadas sobre la respuesta (longitud del lote inicial y tamaño del intervalo) sobre dos casos. Los resultados, sugieren que los errores de contraste fueron de alrededor de 0.0084, estos son considerados pequeños por los autores. Por lo tanto, los modelos propuestos tienen una utilidad adecuada en la aplicación de la simulación de eventos discretos. Finalmente, estos modelos debieran ser utilizados en SM más complejos.

(Lugaresi, Travaglini, & Matta, 2019), construyeron un SM a escala para demostración de conceptos basado en LEGO®. Lo anterior, ya que este tipo de SM digital podría ser clave en el liderazgo de empresas al mejorar la planificación y el control de sistemas de producción son un activo. El sistema, se construyó mediante bloques de tormenta mental LEGO®, palets (rojos y verdes) de madera que representan dos tipos de productos y la transportadora (de bloque EV3 y motor). Los resultados, arrojaron niveles de confianza del 99%, así como valores P <0.001 que sugieren que se cumplen las políticas de producción estipuladas. En ese sentido, el sistema digital fue capaz de tomar una decisión para corregir una tendencia negativa y mejorar su eficiencia en seguimiento a la política programada.

De forma similar (Lugaresi, Alva, & Matta, 2021), desarrollaron modelos a escala de laboratorio de SM para probar tecnologías de simulación en tiempo real y control de producción. En este trabajo, se empleó una celda base LEGO® como la del párrafo anterior y se basó en la norma (ISA-95 Series of Standards: Enterprise-Control System Integration, 2000) así como otras normas industriales. En este trabajo, el sistema a escala reprodujo el flujo de material y lógica de control de producción como en un sistema de manufactura real. Sus resultados basados en caso de estudio sugieren que es posible evaluar modelos de producción y planeación a escala laboratorio. Lo anterior, como soporte rentable en contraste con pruebas en sistemas de manufactura a escala real.

(Fornari, y otros, 2025), publicaron un manifiesto de investigación sobre gemelos digitales en procesos de negocio. Lo anterior, ya que desde su punto de vista gracias al internet de las cosas (IoT), sensores y actuadores es posible monitorear y automatizar procesos físicos que los hacen más eficientes. Con base en lo anterior, se impulsa el paradigma del gemelo digital (GD) como una forma avanzada de gestión de procesos. En ese sentido, combinar dichos procesos físicos (por lo general de negocio) con interacciones en tiempo real promete una nueva forma de guiar las actividades de dicho proceso. Los autores, manifiestan que no es una tarea sencilla, requiere claridad en normatividad e investigaciones rigurosas, pero tendrá un impacto positivo a futuro.

(Bringas-Posadas, y otros, 2025), desarrollaron un GD de un sistema de manufactura a escala laboratorio para planificación de secuencias de producción. Los autores, desarrollaron una celda a escala que, con la incorporación de componentes impresos tridimensionalmente, crearon un ambiente que simula el ensamble de un dispositivo. Lo anterior, con base en los tiempos monitoreados de 10000 equipos que se incorporaron a las rutinas de dos brazos robóticos. Posteriormente, mediante un GD generado con el software CoppeliaSim® se replicaron las rutinas de producción con resultados satisfactorios. Adicionalmente (Bringas-Posadas, Trejo-Morales, Franco-Urquiza, & Cruz-González, 2025), propusieron el uso de técnicas estadísticas

(Shapiro–Wilk y Jarque–Bera) para análisis paramétricos en este tipo de sistemas digital-físico, esto para contribuir con la generación de procesos y normas para la generación de manufactura digital venidera.

Como pudo apreciarse en los párrafos anteriores, existe una tendencia actual para la digitalización de la manufactura, ya que esta es un importante motor económico en las sociedades. Sin embargo, esto se ve acompañado de diversos retos que van desde la normatividad y el desarrollo de los procesos de manufactura digital requeridos para que sean útiles al entorno social. Lo anterior y publicación de leyes (Diputados.Gob, 2025), genera la necesidad nacional para desarrollar software de licencia libre que mitigue gastos para los desarrolladores de tecnología (en su mayoría de centros públicos), por ende, a la sociedad y estos desarrollos puedan transferirse a la pequeña industria nacional (mediante programas y políticas) reduciendo el pago de licencias. Por lo tanto, el objetivo de este artículo es presentar el desarrollo y validación de un solucionador basado en software libre (Python/PyBullet). Adicionalmente, se generó un gemelo digital (GD) de un brazo robótico Dorna® que busca demostrar la viabilidad del desarrollo y su sincronización 1:1 entre el modelo físico y virtual. Así, sentar las bases para su futura integración en sistemas de manufactura flexible más complejos a través de un marco de trabajo digital o framework.

2.0 Metodología para el desarrollo del solucionador.

2.1 Digitalización del brazo robot Dorna®.

Con la finalidad de generar un entorno virtual que pueda ser empleado en un GD y propiciar la manufactura digital, se digitalizará el brazo Dorna® expuesto en los trabajos de (Bringas-Posadas, y otros, 2025). Para la digitalización del brazo Dorna®, se generarán los archivos URDF con base en el ensamble que el fabricante proporciona desde la siguiente referencia (DornaCAD, 2025). Una vez descargado el ensamble, se procederá la asignación de los orígenes y las articulaciones de movimiento. Posteriormente, cada junta se escala (iniciando por la base) mediante un árbol de juntas.

2.2 Desarrollo del entorno mediante software PyBullet.

El motor físico PyBullet versión 3.2.7, es un módulo que se puede cargar en Python versión 3.11.9 se usa en la simulación de eventos físicos para detección de colisiones. En este caso, se aprovechará para la generación del GD del brazo robótico de la celda de manufactura publicada por (Bringas-Posadas, y otros, 2025). Este software, permite la incorporación de archivos de formato de descripción de robot unificado (URDF) que permite enlaces, articulaciones, jerarquía, masas, inercia, colisiones, sensores y motores que son de utilidad en la construcción de los GD.

2.3 Desarrollo de la interfaz gráfica.

Con la finalidad de propiciar el movimiento del robot en el entorno virtual, se desarrollará una interfaz gráfica mediante Phyton. Esta interfaz, será de utilidad para el control del robot Dorna® (modelo físico) y el modelo digital de dicho robot. Además, mediante las pestañas “conexión”, “control individual de juntas” y “trayectorias” se podrá controlar el robot y su ambiente virtual. Lo anterior, mediante la API que el propio robot proporciona.

2.4 Pruebas de contraste del modelo físico y el virtual.

Con la finalidad de propiciar el movimiento se plantea una prueba de trayectoria en la que el modelo virtual y el modelo real ejecutara una secuencia de movimientos como si de un

ciclo de operación se tratase y se evaluaran la concordancia y fluidez entre ambos robots, lo que se busca al hacer esta prueba es que efectivamente el simulador está recreando los movimientos del robot 1:1 esto por la conexión USB del ordenador y el robot Dorna®. En ese sentido, se corrieron tres ejercicios de trayectoria simple a 1, 5 y 10 min de duración. Así, se podrían revisar paros, desviaciones o errores que pudieran ser generados por la programación o desincronización entre el modelo digital y el físico. Adicionalmente, se revisaron las posiciones 0, 120 y 109 entre los modelos físico-digital. Finalmente, se compararon las correlaciones mediante la técnica de regresión lineal para contrastar sus comportamientos y se modelara sus variaciones.

3.0 Resultados.

3.1 Brazo robot Dorna® virtual.

La Figura 1, muestra el brazo Dorna® digitalizado incluyendo sus orígenes, ejes, así como el escalamiento de sus juntas. Con el brazo digitalizado, fue posible comprobar que no hay colisiones. En contraste con un modelo CAD (contiene geometría y malla), este brazo digitalizado ya contiene los archivos URDF con los cuales ya puede operar de forma convincente en un ambiente virtual.

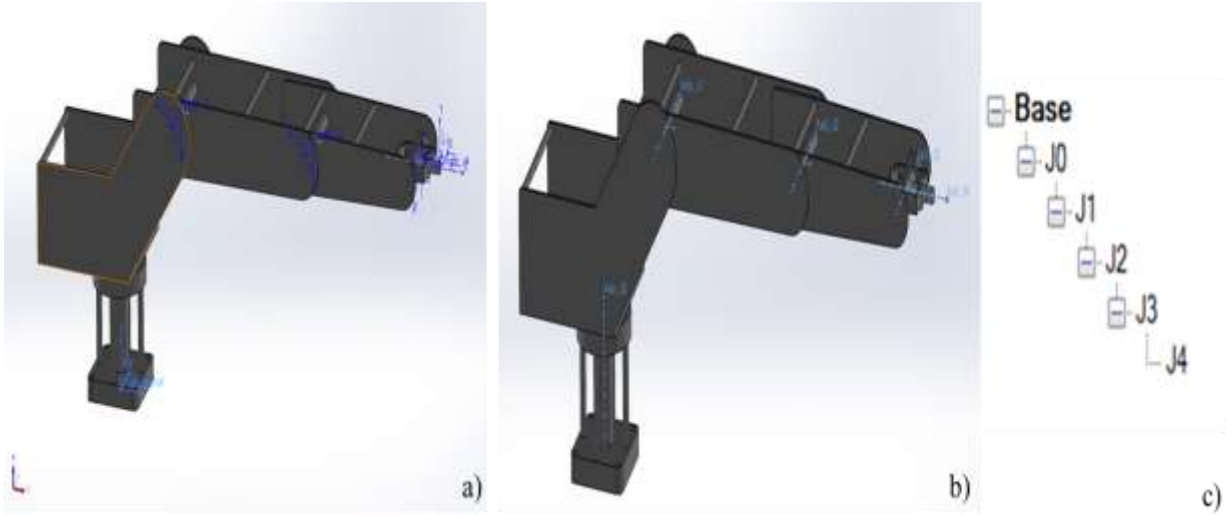


Figura 1. a) Digitalización del brazo robótico orígenes, b) Ejes y c) Escalamiento de las juntas.

3.2 Interfaz gráfica para manipulación del brazo Dorna®.

Previo a cargar el brazo digitalizado en la sección 3.1, se generó el sistema coordenado mostrado en la Figura 2. La Figura 2a, muestra el entorno y el ejemplo básicos de manual para la validación previa a cargar los URDF del brazo robótico. Además, la Figura 2b muestra el eje coordenado asignado por el PyBullet versión 3.2.7 que servirá como referencia al brazo. En ese sentido, la línea azul representa el eje Z, la roja el eje X y la verde el eje Y (todos positivos), mostrando la configuración inicial del PyBullet versión 3.2.7. En dicha configuración, se obtuvo un valor de 9.81 como fuerza al eje Z, simulando la aceleración de la gravedad en el mundo real (aproximadamente 9.81 m/s).

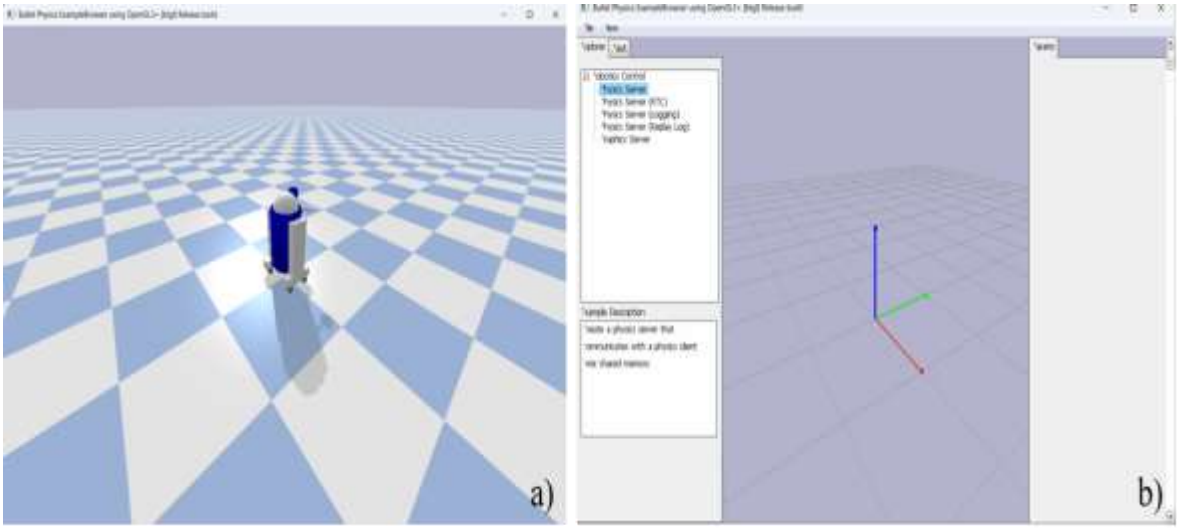


Figura 2. a) Entorno ejemplo básico de manual, b) Sistema coordenado para el desarrollo del brazo robótico Dorna® en PyBullet versión 3.2.7.

Al cargar el URDF de todos los componentes, se construye el modelo virtual mostrado en la Figura 3a, el cual es la representación digital del sistema físico (ver Figura 3b). Las pruebas, demuestran que el robot se mueve mediante la interfaz que constituye de 3 pestañas denominadas conexión, control individual de articulaciones y trayectorias. Cabe mencionar, que la pestaña conexión controla al brazo robot digitalizado mediante su API y su representación fue completa. Esta representación, incluye eslabones rígidos, articulaciones con diferentes tipos de movimiento (como rotacionales o prismáticos) y las propiedades físicas (masa, fricción, límites articulares, etc.).

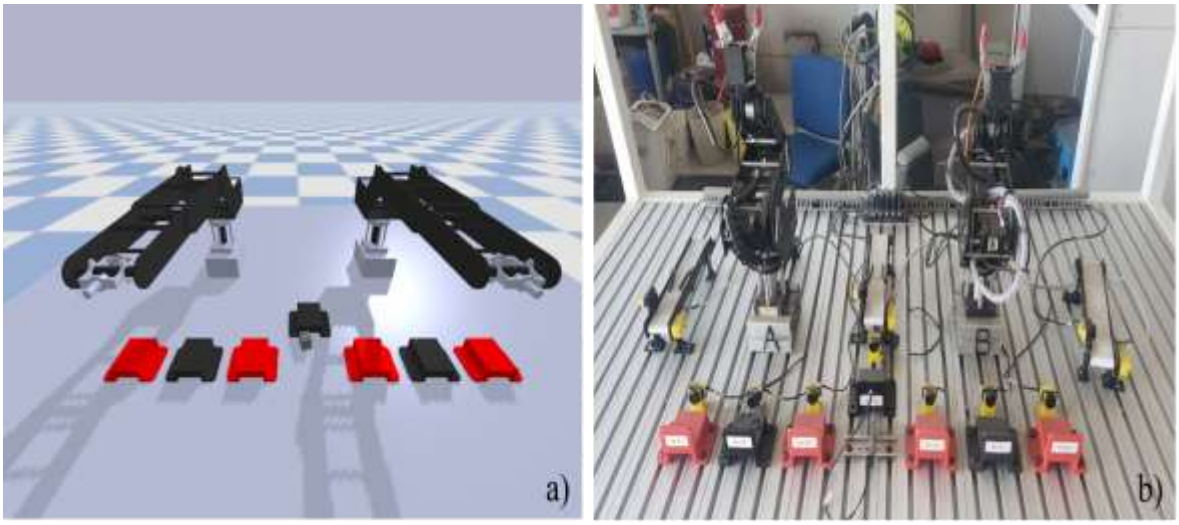


Figura 3. a) GD de dos brazos robot Dorna® y b) Sistema de manufactura a escala laboratorio (sistema físico).

La Figura 4a, muestra la interfaz para el control de los deslizadores, es notorio que al mandar a cero el brazo su representación digital cumplió la instrucción (ver Figura 4b). En ese sentido, los deslizadores establecieron las posiciones objetivo de las articulaciones. Cada deslizador, representó una articulación del robot y permite al usuario definir (en

tiempo real) el valor deseado dentro de un rango específico. Así, es posible explorar las configuraciones articulares y probar movimientos sin necesidad de escribir secuencias de comandos.

El brazo digitalizado y su interfaz, permite que el valor que el usuario ha seleccionado en el deslizador no mueva de forma directa la articulación. En su lugar, se interpreta como una posición deseada, entonces el PyBullet versión 3.2.7 activa un controlador interno —normalmente un controlador de tipo PID (Proporcional–Integral–Derivativo)— para aplicar el torque necesario que lleve la articulación hacia esa posición. En este proceso, se considera de manera realista las restricciones físicas del sistema, incluyendo la gravedad, las colisiones y la dinámica del movimiento.

Durante la simulación PyBullet versión 3.2.7 evaluó continuamente el estado del sistema, en cada paso ajustó la fuerza ejercida por los motores virtuales, minimizando el error entre la posición actual y la posición deseada. De esta forma, el sistema se comporta como si estuviera equipado con motores reales y los deslizadores funcionaron como una interfaz de alto nivel en el control de dichos motores.

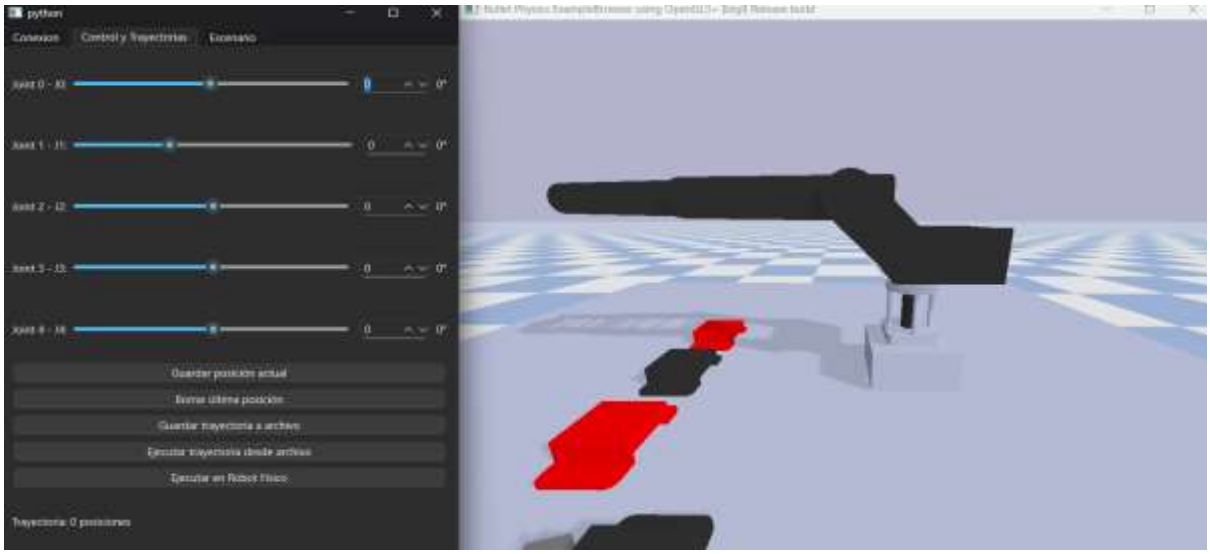


Figura 4. a) Interfaz para el control de deslizadores y control de trayectorias y b) Posición cero de uno de los brazos Dorna®.

Además, en la simulación se tiene una pestaña de “Escenario” en la que se puede visualizar todos los componentes dentro de la simulación que marca cuanto URDF hay, sus ID, y su posición dentro de las variables globales del sistema (ver Figura 5).

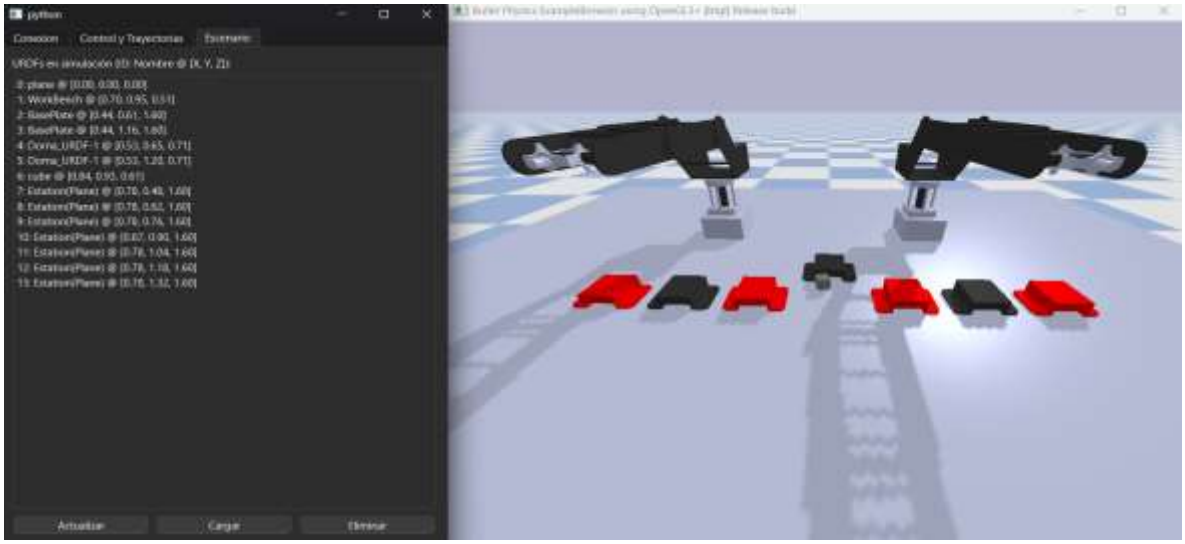


Figura 5. Muestra la pestaña de información sobre URDF activos en la simulación y su posición global con respecto al punto de origen.

3.3 Resultados de comparación de trayectorias entre el robot Físico y el Digital.

La Figura 6, muestra la trayectoria de un minuto de duración que corresponde al brazo robot realizado en este trabajo, se puede ver que los movimientos entre el brazo digital y el físico son similares. Por otro lado, las Figuras 7-8 muestran las trayectorias entre los brazos digital y físico, cuyas duraciones son 5 y 10 min, respectivamente. Es notorio, que entre mayor es el tiempo, mayor serán la cantidad de imágenes y detalle. Así, se esperaría que al comparar las posiciones ocurriese un incremento en los errores entre los brazos robóticos y digital.

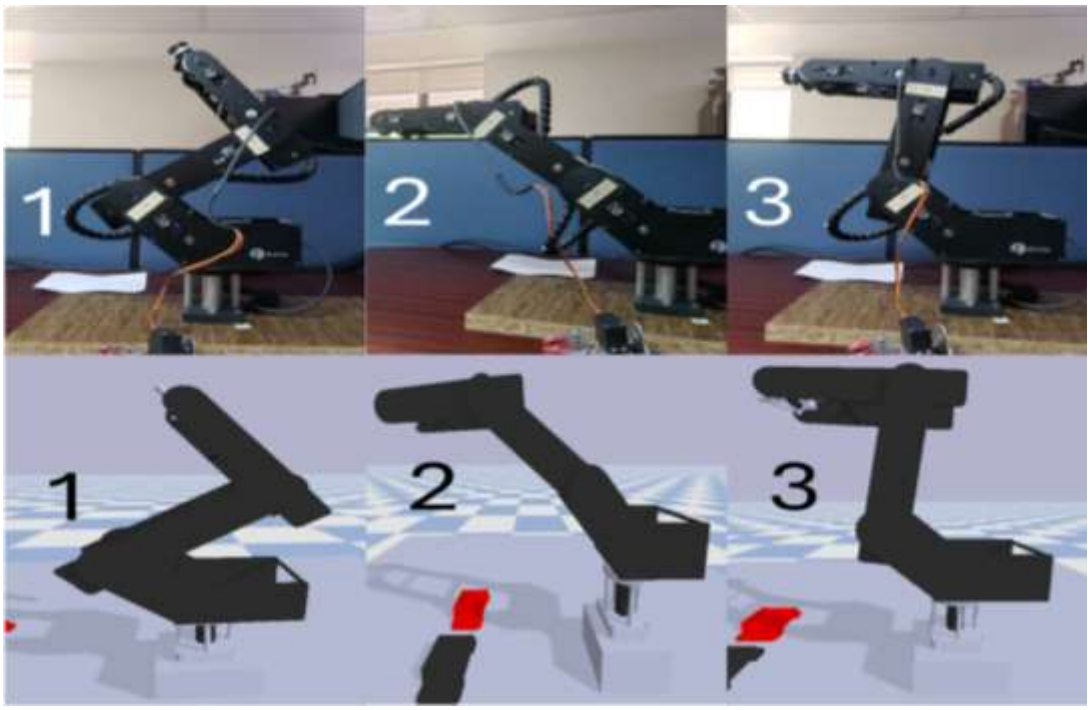


Figura 6. Posiciones de la primera trayectoria (1min) de comparación entre robot digital y físico.

La primera trayectoria (1, 5 y 10 min), generaron 293, 1408 y 2834 datos para cada uno de los tiempos considerados dentro del experimento que representan entre el 5~10 veces respecto a la duración de 1min. En contraste, la trayectoria 2 generaron 291, 1414 y 2836 datos con una tendencia similar a la de la primera trayectoria. Finalmente, los datos generados para la última trayectoria fueron 280, 1411 y 2836 para cada tiempo.

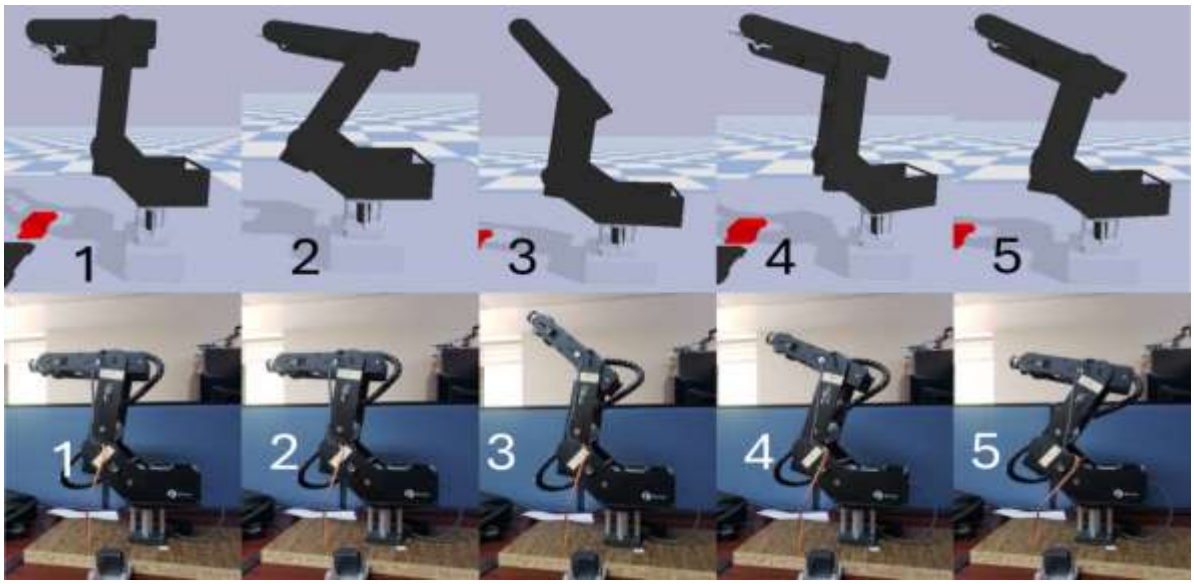


Figura 7. Posiciones de la segunda trayectoria (5min) de comparación entre robot digital y físico.



Figura 8. Posiciones de la segunda trayectoria (10min) de comparación entre robot digital y físico.

Para la prueba se realizaron las tres trayectorias antes definidas en este documento y se repitieron 3 veces con un tiempo diferente, la rutinas se ejecutaron de manera seguida durante 1, 5 y 10 minutos, se programó un script en Python versión 3.11.9 para que se obtuvieran datos de posición de ambos robots (físico y Digital) con un intervalo de tiempo de 0.2 segundos para demostrar que el valor de las juntas del robot físico es igual a su homónimo digital en todo momento durante la simulación, constatando así que el robot Físico y el Real son en efecto un Gemelo Digital. La Figura 9, muestra el comparativo para la trayectoria dos de este trabajo, como puede apreciarse existe una buena correlación lineal entre el sistema físico y el digital. En la sección 7.0, se muestra el script del solucionador de este software como referencia al lector.

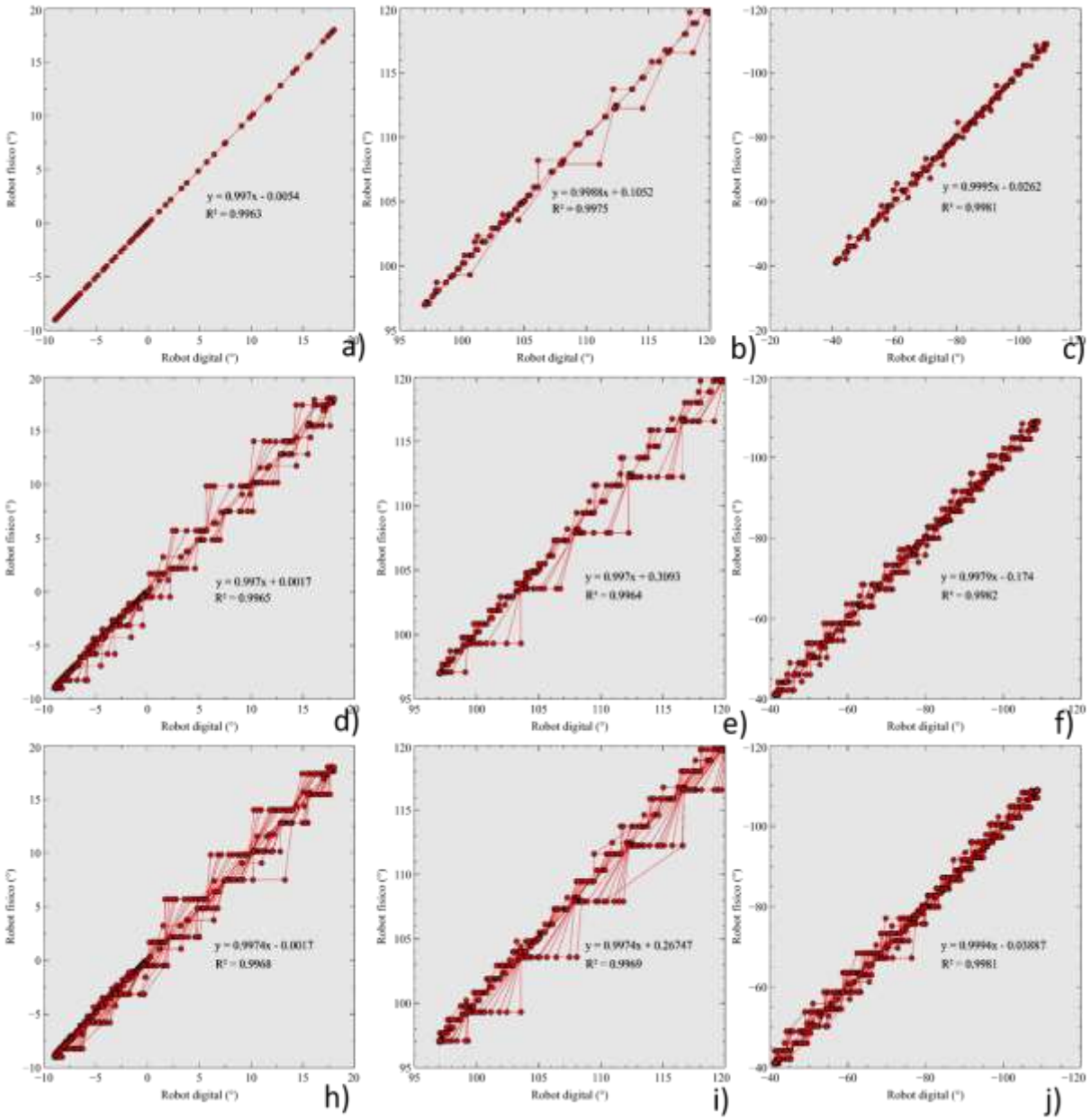


Figura 9. Comparativo en posiciones entre sistema digital y físico para a-c) 1min, d-f) 5min y h-j) 10min para la segunda trayectoria.

La Figura 9, muestra una Figura representativa del comportamiento del brazo robot físico y el digital cuyo comportamiento se rige por la ecuación $Y = m + bx$. La Tabla 1, resume los resultados de las tres trayectorias en los tres tiempos (1, 5 y 10 min) estudiados dentro de este trabajo. Como puede notarse, existe una buena correlación entre los datos del robot físico y el digital (mayor al 90% de correlación).

Tabla 1. Resultados de trayectorias y tiempos para este experimento.

No.	Trayectoria	Tiempo (min)	Posición (°)	Ecuación	R^2
1	**1**	1	-10~20	y = 0.997x - 0.054	0.9963
			95~120	y = 0.9988x + 0.1052	0.9975
			-20~120	y = 0.995x + 0.0262	0.9981
		5	-10~20	y = 0.997x + 0.0017	0.9965
			95~120	y = 0.997x + 0.3096	0.9964
			-20~120	y = 0.9979x - 0.174	0.9982
		10	-10~20	y = 0.9974x - 0.0017	0.9968
			95~120	y = 0.9974x + 0.26747	0.9969
			-20~120	y = 0.9994x - 0.03887	0.9981
3	**3**	1	-10~20	y = 0.3739x - 0.0334	0.9451
			95~120	y = 1.0007x + 0.1899	0.9993
			-20~120	y = 1.0045x + 0.1929	0.9984
		5	-10~20	y = 0.9998x - 0.0016	0.9998
			95~120	y = 0.9992x - 0.0289	0.9995
			-20~120	y = 0.999x + 0.1097	0.9993
		10	-10~20	y = 0.9974x - 0.0017	0.9968
			95~120	= 0.9974x + 0.2674	0.9969
			-20~120	y = 0.9994x - 0.0388	0.9981

4.0 Discusión

En la actualidad, se está trabajando en el desarrollo de tecnología de manufactura virtual que permita mejorar procesos y generar mejores motores para la economía (STATISTA, 2024) (ONUDI, 2021). En ese sentido, el desarrollo de este solucionador basado en Python/PyBullet (software libre) así como su DG, sugirieron una buena verosimilitud estadística ya que su valor de correlación de las trayectorias analizadas fue de alrededor $R^2 \approx 99\%$. Algo similar, fue

reportado por (Lugaresi, Aglio, Folgheraiter, & Matta, 2019) cuyos valores de error fueron de alrededor de 0.0084 entre el modelo digital y el físico. Lo anterior, sugiere que la simulación de eventos discretos entre modelos físicos y digitales es útil, así como promisoria para SM.

Adicionalmente, el desarrollo de sistemas a escala (físicos-digitales) concuerda con los trabajos presentados por (Lugaresi, Alva, & Matta, Parte A, 2021) quienes demostraron que es posible desarrollar modelos a escala (con LEGO® o celdas específicamente diseñadas) para el escalamiento de SM a nivel industrial. En dichos trabajos, se sugirió una sincronización 1:1 entre el modelo digital y el físico. Esto, permitió que el sistema digital pudiera tomar decisiones y validar los esquemas de producción de forma correcta.

También (Bringas-Posadas, y otros, 2025), presentaron un planificador de secuencias basado en CoppeliaSim® que replicó las rutinas de ensamble entre un modelo físico y su respectivo GD. Los autores, emplearon el PyBulltet y controladores PID que permitieron validar la capacidad de simulación; así como la consideración de restricciones físicas como la gravedad, colisiones y el torque. Lo anterior, para generar un desarrollo más robusto y de código abierto para generar un marco de trabajo (framework) de fácil acceso. También, mediante la aplicación de la regresión lineal para validación del sistema, pudo complementarse con la validación previa mediante estadística no paramétrica presentada por (Bringas-Posadas, Trejo-Morales, Franco-Urquiza, & Cruz-González, 2025). Así, fortalecer el desarrollo de normatividad para la generación de manufactura digital estructurada y que complemente el estándar (ISA-95 Series of Standards: Enterprise-Control System Integration, 2000).

Finalmente, tal como indicaron (Fornari, y otros, 2025), al integrar los GD en procesos de negocio y SFM, será posible mejorar la gestión de procesos con base en un paradigma avanzado focalizado en eficiencia. Cabe mencionar que la importancia de estas plataformas, no radica solo en la reducción de costos de operación, ya que es posible reconfigurar y adaptarse a demandas de trabajo variables (Colledani, Yemane, Lugaresi, Borzi, & Callegaro, 2018). Por lo tanto, el desarrollo de este solucionador sienta las bases de un marco de trabajo o framework que facilitará la transición de las pequeñas y medianas empresas (PyMEs) hacia la digitalización de su manufactura con soluciones de bajo costo y funcionales. También, es posible la integración de sensores de corriente en un brazo robótico para calcular esfuerzos en los servomotores y mediante el uso del GD calcular la probabilidad de falla por desgaste y prevenirlo antes de que suceda. Otra posibilidad, es la creación de un entorno de entrenamiento de operadores para programación de un brazo robotizado. Así, sería posible reducir la probabilidad de un daño al equipo y al usuario al evitar colisiones. Esta aplicación, podrá ser utilizada para optimización de trayectorias de producción y control de calidad mediante sistemas de visión artificial. Finalmente, aunque esta aplicación está pensada para pequeñas industrias, no es limitativo su uso a empresas más grandes. Sin embargo, los autores consideran que la puerta de apertura sería las pequeñas empresas, quienes serían beneficiadas en su economía y el desarrollo retroalimentado en un entorno real de operación robusta.

5.0 Conclusiones

Con base en los resultados y discusión, es posible concluir que el desarrollo de un solucionador con base en PyBullet versión 3.2.7 para un brazo robótico Dorna® sugiere

que es posible alcanzar una buena correlación entre el modelo digital y el físico. Esto, podrá sentar las bases para el desarrollo de marcos de trabajo (framework) basados en estadística para generar esquemas de manufactura digital confiables. Dado que el uso de GD generará secuencias de planificación y control de producción eficaces, el uso de este tipo de solucionador permitirá complementar metodologías de validación en tiempo real. Además, la integración de análisis estadístico en la validación de la interacción estos entornos permitirán al fortalecimiento de la normatividad relacionada y el fortalecimiento de la manufactura nacional. Todo lo anterior, sugirió que mediante el uso de software libre es posible desarrollar solucionadores como alternativa técnica válida (sincronización visual y estadística). Así, reducir costos de licenciamiento derivado de las marcas de software especializado y comercial.

6.0 Agradecimientos

A la secretaría de ciencia y humanidades por su apoyo y beca.

7.0 Referencias

Bringas-Posadas, D., Devillet-Castellanos, H. D., Cruz-González, C. E., Vargas-Arista, B., Guzmán-Flores, I., & Franco-Urquiza, E. A. (2025). Development of a digital twin and lab-scale manufacturing system for production sequences planning. *The International Journal of Advanced Manufacturing Technology, 136*, 5515–5530.

Bringas-Posadas, D., Trejo-Morales, A., Franco-Urquiza, E. A., & Cruz-González, C. E. (2025). Exploratory statistical analysis for decision-making in a scaled manufacturing system of medical equipment. *The International Journal of Advanced Manufacturing Technology, 138*, 555–568.

Colledani, M., Yemane, A., Lugaresi, G., Borzi, G., & Callegaro, D. (2018). A software platform for supporting the design and reconfiguration of versatile assembly systems. *Procedia CIRP, 72*, 808-813.

Diputados.Gob. (16 de 04 de 2025). Obtenido de Ley de austeridad republicana artículo 12: https://www.diputados.gob.mx/LeyesBiblio/pdf/LFAR.pdf

DornaCAD. (25 de 07 de 2025). Obtenido de https://drive.google.com/file/d/19kcPUnuemhYqBeXViNxjs8llj91vV8vv/view

Flores, L. (19 de 01 de 2024). *Tres claves para optimizar procesos [Manufactura 4.0]*. Obtenido de Tesselar: https://www.tesselar.mx/blog/tres-claves-para-optimizar-procesos-manufactura-4.0#:~:text=1.,e%20informes%20en%20tiempo%20real.

Fornari, F., Compagnucci, I., De Donato, M. C., Bertrand, Y., Beyel, H. H., Carrión, E., . . . Valderas, P. (2025). Digital Twins of Business Processes: A Research Manifesto. *Internet of Things, 30*, 1-14.

ISA-95 Series of Standards: Enterprise-Control System Integration. (2000). International Society of Automation.

London Consulting Group. (04 de 06 de 2024). Obtenido de Como optimizar los procesos de manufactura: https://londoncg.com/blog/como-optimizar-los-procesos-con-un-sistema-de-manufactura#:~:text=%C2%BFQu%C3%A9%20es%20la%20optimizaci%C3%B3n%20de,de%20producci%

C3%B3n%20en%20una%20empresa.

Lugaresi, G., Aglio, G., Folgheraiter, F., & Matta, A. (2019). Real-time Validation of Digital Models for Manufacturing Systems: a Novel Signal-processing-based Approach. *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)* (págs. 450-455). Vancouver, BC, Canada: IEEE Xplore.

Lugaresi, G., Alva, V. V., & Matta, A. (2021). Lab-scale Models of Manufacturing Systems for Testing Real-time Simulation and Production Control Technologies. *Journal of Manufacturing Systems, 58*, 93-108.

Lugaresi, G., Travaglini, D., & Matta, A. (2019). A LEGO® Manufacturing System as Demonstrator for A Real-Time Simulation Proof of Concept. *Proceedings of the 2019 Winter Simulation Conference* (págs. 1-12). National Harbor, Maryland, USA: IEEE Xplore.

ONUDI. (24 de 07 de 2021). Obtenido de ¿Por que el desarrollo industrial es más importante que nunca?: https://ods9.org/news/448/por-que-el-desarrollo-industrial-es-mas-importante-que-nunca

Ramirez, O. (19 de 03 de 2025). *México, Potencia en Desarrollo de Software en América Latina*. Obtenido de Bambú Mobile: https://bambu-mobile.com/mexico-potencia-en-desarrollo-de-software-en-america-latina/#:~:text=El%20mercado%20de%20desarrollo%20de,del%206.28%25%20anual%20hasta%202027.

Rockwell Automation. (2025). Obtenido de 10.º informe anual sobre la situación de la fabricación inteligente: https://www.rockwellautomation.com/es-mx/capabilities/digital-transformation/state-of-smart-manufacturing.html?utm_source=Google&utm_medium=Paid_Search&utm_campaign=MultiInitiative_MultiIndustry_Global_ES_CMP-07522-S6S3W2&utm_term=industria%204%200&utm_co

Salado-Morales, E. (s.f.). *Los mejores software de Producción y Fabricación en 2025*. Obtenido de SoftDoit: https://www.softwaredoit.es/software-erp/software-produccion.html#:~:text=Un%20ERP%20de%20producci%C3%B3n%20permite,fabricaci%C3%B3n%20para%20optimizar%20la%20planificaci%C3%B3n.

Salvagnini. (2024). Recuperado el 27 de 03 de 2026, de La competitividad no es solo velocidad. Es sobre todo eficiencia.: https://www.salvagninigroup.com/es-ES/empresa/automatizacion-flexible#:~:text=*%20Salvagnini%20ES.%20*%20Empresa.%20*%20Automatizaci%C3%B3n%20flexible.

Secretaría de medio ambiente CDMX. (s.f.). (Gobierno de la ciudad de México) Recuperado el 24 de 07 de 2025, de Glosario definición: https://proyectos.sedema.cdmx.gob.mx/datos/glosario-definicion/Econom%C3%ADa#:~:text=Dado%20que%20los%20recursos%20del,se%20producen%20dentro%20del%20pa%C3%ADs.

STATISTA. (01 de 11 de 2024). Obtenido de La industria manufacturera en México – Datos estadísticos: https://es.statista.com/temas/7853/la-

industria-manufacturera-en-mexico/#:~:text=La%20industria%20manufacturera%20es%20la,9%2C7%20millones%20de%20trabajadores.

8.0 Anexo

Sección del script del solucionador expuesto en este trabajo.



```python
def process_motion_instruction(self, instruction):

    sys_id = instruction.get("system")
    joints = instruction.get("joint_targets", [])
    target = instruction.get("cartesian_target")
    mode = instruction.get("mode", "absolute")
    actuator = instruction.get("actuator", 0)
    delay = instruction.get("delay", 0.0)

    system = self._select_system(sys_id)
    if system is None:
        return None

    ctrl, model, sim, latency = (
        system["controller"],
        system["model"],
        system["simulated"],
        system["latency"]
    )

    is_real = ctrl and ctrl.is_available()
    t0 = time.time()

    try:
        if is_real:
            (ctrl.move_cartesian(target, mode == "absolute") if target
             else ctrl.move_joints(joints))

            if "actuator" in instruction:
                ctrl.set_actuator(bool(actuator))

            while not ctrl.is_available():
                time.sleep(0.02)

        else:
            if target:
                target = self._resolve_target(target, mode, sim)
                joints = model.inverse_kinematics(target)
                if joints is None:
                    return None
                self._simulate_cartesian_motion(sim, model, target)
            elif joints:
                self._simulate_joint_motion(sim, joints)

    except Exception:
        return None

    self._update_actuator(sim, actuator)

    if delay > 0:
        self._apply_delay(delay, is_real)

    elapsed = (time.time() - t0) + delay

    final_joints = joints
    real_pos = [None]*3
    est_pos = self._forward_estimation(model, joints)

    if is_real:
        state = self._safe_get_state(ctrl)
        if state:
            final_joints = state["joints"]
            real_pos = state["position"]

    return {
        "system": sys_id,
        "commanded_joints": joints,
        "final_joints": final_joints,
        "real_position": real_pos,
        "estimated_position": est_pos,
        "actuator": actuator,
        "latency": latency,
        "delay": delay,
        "execution_time": round(elapsed, 3),
    }
```