

# TRABAJO DESEMPEÑADO Y AVANCES LOGRADOS DURANTE LA ANUALIDAD 1

## INVESTIGACIÓN WORKSTATION OPTIMIZATION & LINE REBALANCING

**Propuesta General de Solución: Optimización del Diseño del Puesto de Trabajo Basada en datos en tiempo real bajo la filosofía I3oT y Continuous Twins.**

### **1.-Factorías en la cuarta revolución industrial. Bienvenidos a "la Jungla"**

La gobernanza de las factorías, como pueden ser las factorías del sector de automoción, son extremadamente complejas. Utilizan miles y miles de robots, pinzas, cilindros, cintas transportadoras, etc., cada una con sus componentes, motores eléctricos, engranajes, cadenas, y cada una aplicada a diferentes procesos, como soldadura, estampación, pintura, etc. Además, toda esta maquinaria interactúa con los operarios que intervienen en diferentes fases del proceso como por ejemplo, ensamblando componentes, verificando la calidad de las piezas, y, en algunos casos, con la capacidad de modificar parámetros de las máquinas para garantizar la productividad y calidad de las piezas. El objetivo de la automatización no es otro que tratar de eliminar la dependencia de ese factor humano. Sin embargo, las máquinas no son capaces de adaptarse a situaciones de planta, lo que pone en duda si una automatización completa de una factoría sin la presencia humana sería la más eficiente. Así pues, que los operarios puedan modificar ciertos parámetros de las máquinas responde a una realidad en los procesos de fabricación y no es otra que la variabilidad. Esta variabilidad puede venir desde diferentes fuentes, como por ejemplo que dos máquinas iguales en realidad se comporten de manera diferente, desconocimiento en profundidad del proceso, tecnologías de diferentes generaciones conviviendo en la misma máquina y también en las líneas de montaje, la intervención humana en el proceso de ensamblado.

En la fábrica de Ford Valencia, la producción diaria es de unas 2.000 unidades. Cualquier retraso en el ensamblado o fallo de calidad que genere retrabajos o chatarra, puede generar grandes pérdidas, ya que incrementa directamente el coste de fabricación del producto. Esto genera un importante nivel de presión tanto a los gerentes como al resto de personal de planta ya que, una mala gestión que genere pérdidas puede tener consecuencias. Por todo lo anterior, los empleados llaman a la fábrica con el sobrenombre de "la jungla".

#### **1.1 Línea de montaje**

Una línea de montaje es un sistema de producción en el que un producto se mueve secuencialmente a través de estaciones de trabajo, donde cada operario o máquina realiza una tarea específica para añadir o ensamblar componentes, completando el producto final de manera rápida, eficiente y económica. Es un método clave para la producción en masa,

caracterizado por la división del trabajo y la especialización de tareas para maximizar la velocidad y la uniformidad del producto.

Características clave de una línea de montaje:

- **Movimiento secuencial:** El producto sin terminar se desplaza de forma continua de una estación de trabajo a otra.
- **Tareas específicas y repetitivas:** Cada estación de trabajo se especializa en una tarea simple y repetitiva, lo que aumenta la eficiencia.
- **División del trabajo:** El proceso de fabricación se divide en múltiples fases, cada una realizada por un operario o máquina distinto.
- **Producción en masa:** Es el método más eficaz para producir grandes cantidades de un mismo producto.
- **Mayor velocidad y eficiencia:** La especialización y la repetitividad de las tareas permiten fabricar productos más rápido.
- **Reducción de costos:** Al ser más eficiente, se reducen los costos de producción, lo que se refleja en un precio más bajo para el consumidor.

En la literatura existen multitud de algoritmos para la optimización de líneas de montaje, así como de líneas de fabricación en general. A este problema se le conoce como ALBP (Assembly Line Balancing Problem), ver [1]. Estos algoritmos se utilizan de manera masiva en el diseño de las líneas, pero, ¿Qué pasa después? Pues después, una vez instalada la línea y puesta en marcha, el problema de su gobernanza pasa a manos de los gerentes y los empleados de la fábrica, generando multitud de nuevos problemas como el propuesto en este trabajo de investigación, que no es otro que el rebalanceo de las líneas de montaje (Optimización del Layout) bajo condiciones de fabricación.

## 1.2 IoT,IIoT,I3oT

El Internet de las Cosas (IoT) es una red de objetos físicos con sensores, software y otras tecnologías integradas que se conectan e intercambian datos a través de internet u otras redes de comunicación. Estos dispositivos "inteligentes" abarcan desde artículos domésticos cotidianos como termostatos inteligentes, móviles, wereables, etc, y donde la tendencia es a incrementar exponencialmente su uso en la sociedad. A nivel industrial, el concepto IoT se traduce al IIoT (Internet de las Cosas Industrial). Este se utiliza en entornos industriales para recopilar, analizar y compartir datos en tiempo real, lo que permite mejorar la eficiencia, la productividad y la automatización. Se diferencia del Internet de las Cosas (IoT), orientado al consumidor, al centrarse en los procesos industriales y emplear tecnologías como la IA, el aprendizaje automático y la computación en la nube para optimizar las operaciones y generar beneficios económicos. El principal problema del IIoT es que necesita instalar sensores y extraer la información. Para facilitar la instalación, el mercado ofrece sensores que incluyen baterías (para evitar la conexión a la red eléctrica de la factoría) y con comunicación 5G para extraer el dato sin necesidad de utilizar la red industrial, utilizada para el funcionamiento de la factoría. Esto hace que el coste de estos dispositivos sea muy elevado y no se implemente masivamente a todas las máquinas y componentes que existen en una industria. Esta falta de datos está ralentizando la implementación de tecnologías de la Industria 4.0 como Digital twins, machine learning, computación en la nube, etc y retrasando la tan ansiada llegada de la Industria 5.0, donde aparecen tecnologías como la IA.

En nuestros trabajos anteriores, ver [2] se propone un nuevo concepto, el I3oT (Industrializable Industrial Internet of Things). La idea de este nuevo concepto es utilizar la instalación disponible en las factorías para desarrollar las aplicaciones IIoT a partir de ellas. Las máquinas instaladas en la industria funcionan de manera automática y disponen de sensores que permiten que las máquinas/líneas funcionen de manera automática. Además, las factorías disponen de una red IT/OT donde se comunican las máquinas y se gestiona la eficiencia de las líneas. Bajo este paradigma, las aplicaciones I3oT serían fácilmente extrapolables y escalables al resto de sistemas con un coste y consumo energético muy bajo, permitiendo el asentamiento definitivo de la Industria 4.0 y sus tecnologías en la industria, así como la reducción del consumo energético y la huella de carbono de las aplicaciones.

#### 1.2.1 Aplicaciones I3oT desarrolladas susceptibles de ser usadas.

##### 1.2.1.1 Gestión de la información: CrossPLC:

En uno de nuestros recientes trabajos, ver [3], se desarrolla una herramienta I3oT multiplataforma que permita la extracción eficiente de la información de los sensores existentes, conocida como CrossPLC. Esta herramienta actúa de manera pasiva, es decir, los PLC's o PC's industriales envían la información a través de la red IT (red industrial) si y solo si, el parámetro ha cambiado. CrossPLC actualiza el valor de esa variable en su base de datos. Las aplicaciones I3oT, alojadas en la red OT acceden a esta base de datos para la ejecución de sus funciones. Esta herramienta permite la extracción de la información de una manera eficiente, usando la red IT existente y sin colapsar las comunicaciones de la factoría.

##### 1.2.1.2 Sistema de predicción de averías de componentes y máquinas (Miniterms)

En otro de nuestros trabajos anteriores, ver [4], se utiliza la filosofía I3oT para desarrollar una herramienta de predicción de fallos en componentes de máquinas. Mediante el uso los sensores ya instalados en las líneas de fabricación, se mide el tiempo que les cuesta a los componentes de las líneas en hacer su tarea. Estos tiempos se envían a través del CrossPLC y son analizados en servidores instalados en la red OT. Cuando estos tiempos sufren un aumento significativo, es indicador de que el fin de la vida útil del elemento se acerca.

### 1.3 The twin continuum. From Virtual to Digital twin

El gemelo digital es una tecnología con muchas aplicaciones en la cuarta revolución (Industria 4.0). En general, se trata de disponer de manera virtual de una réplica del proceso o máquina y usar esta para mejorarlo, [7]. Gemelo digital y Gemelo virtual son dos conceptos diferentes. Estos dos conceptos se pueden definir como:

- Virtual Twins: Están fundamentados básicamente en la física conocida y su simulación mediante software a través de métodos numéricos. Suelen tener sistemas complejos y un coste muy alto en recursos, incluido el tiempo de cómputo. Los resultados obtenidos tienen variabilidad e incertidumbre con el proceso real.
- Digital Twins: Están fundamentados en los datos y no en la física. Al estar basado exclusivamente en datos, su funcionamiento es difícilmente explicable y por tanto certificable. Suelen utilizar técnicas de machine learning para aprender un modelo. Su principal problema es que dependen de los datos y necesitan gran cantidad de estos para poder estimar comportamientos adecuados. Desafortunadamente los datos en la industria son altamente costosos. ¿Que datos coger?, ¿dónde están?, y cuando y como cogerlos son preguntas a resolver en estos casos. El modelo obtenido es útil dentro del

rango de datos entrenado, pero extrapolar y trabajar fuera del puede no dar los resultados esperados, es arriesgado.

Entre estos dos extremos, existen puntos intermedios que tratan de resolver los problemas del digital o virtual twin, y es lo que se conoce como “twin continuum”. Los puntos intermedios se pueden dividir en dos grandes bloques, estos son:

- **Physics informed learning:** Utiliza PINNs (Physics-informed neural networks), que son un tipo de aproximadores donde se puede incluir cualquier ley física que pueda ser definida como una PDE (Partial Differential Equation) y que gobierne el proceso de aprendizaje. Este tipo de redes, al contener la física no necesitan tantos datos como el Digital Twin y su funcionamiento es explicable, ya que contiene la física.
- **Physics augmented learning:** En este caso, la idea consiste en tratar de aprender la discrepancia existente entre el virtual twin y el mundo real. A este Gap se le suele llamar “ignorancia” ya que, como en el digital twin no es explicable. Se suelen utilizar técnicas de Machine learning para su aprendizaje. Suele necesitar menos datos ya que solo se pretende aproximar la discrepancia.

## 2.- Visión General de la propuesta

Esta propuesta articula un sistema integral para la **optimización estratégica del diseño de estaciones de trabajo** en líneas de ensamblaje existentes. El sistema se basa en la captura detallada de datos en tiempo real, a ser posible bajo la filosofía I3oT, de la posición en tiempo real y las acciones del operario y el desarrollo de un Continuous twin para modelar y validar configuraciones de la estación de trabajo y emplea una estrategia de optimización híbrida (Algoritmos Genéticos para la búsqueda de configuraciones y Aprendizaje por Refuerzo Profundo para la evaluación realista en simulación) para encontrar el diseño más eficiente. El presente documento se ha redactado pensando en que se dispone de la posición del operario en tiempo real y que los datos han sido generados mediante tracking por visión artificial generado por la empresa DOMOTIK. El objetivo es reducir desplazamientos innecesarios, minimizar tiempos de ciclo dentro de la estación y mejorar la ergonomía y la eficiencia operativa.

### 2.1 Fase 1: Extracción y Procesamiento de Datos de Minitérminos Dinámicos

En cambio, se recibe directamente de la línea de ensamblaje datos de alta precisión consistentes en coordenadas espaciales bidimensionales (x,y)(x,y) del coche y del operario, junto con su identificación y timestamp, capturando así en tiempo real:

- La posición exacta del operario y del coche dentro del área de la estación.
- El recorrido, desplazamientos y estancias que el operario realiza a lo largo del tiempo.
- El lugar y momento preciso donde el operario realiza cada tarea sobre el coche.

Objetivo: Realizar un procesamiento posterior (postprocesado) de estos datos de posición para:

- Transformar coordenadas a la referencia espacial deseada.
- Definir la cuadrícula espacial que segmenta el área intraestación en casillas discretas.
- Extraer y estructurar los **minitérminos**, entendidos aquí como las unitarias acciones o movimientos del operario entre casillas de la cuadrícula, o la duración de estancias específicas dentro de casillas, que representan tareas o pausas.

- Asociar tiempos y desplazamientos a cada minitérmino observado.

### Componentes y Actividades Clave:

#### Procesamiento de Visión por Computadora (CV):

- **Postprocesado de Posiciones y Tiempos:**
  - Transformación de coordenadas  $(x,y)$  del operario y coche a un sistema de referencia realista.
  - Segmentación en casillas discretas conformando una cuadrícula intraestación.
  - Identificación de desplazamientos (desplazamiento entre casillas consecutivas) y estancias (tiempo en una casilla sin desplazamiento).
  - Construcción de perfiles de minitérminos unión de estos desplazamientos y estancias temporales.
- **Identificación de Tareas vs. MinTérminos:**
  - Enfoque simplificado y robusto: considerar directamente los desplazamientos y estancias como minitérminos, sin la necesidad de entrenar explícitamente modelos de reconocimiento de tareas.
  - Importancia creciente de diferenciar, si es relevante, qué tarea se realiza en cada estancia o desplazamiento, para entender el contexto operativo.
  - En caso de no poder identificar la tarea automáticamente, registrar la posición y duración para análisis y simulaciones posteriores.

#### Componente Gráfico y Dashboard:

- Se diseñará una **interfaz gráfica de usuario tipo dashboard** que permita la visualización **en tiempo real y retroalimentación histórica** de los datos extraídos.
- Esta interfaz incluirá:
  - **Mapas de calor dinámica** que muestran la intensidad y frecuencia de movimientos dentro de la cuadrícula, destacando zonas de mayor actividad, fatigabilidad o posibles cuellos de botella espaciales.
  - Visualización de **rutras típicas** y trayectorias recorridas por los operarios, para identificar patrones recurrentes, ineficiencias o riesgos ergonómicos.
  - Estadísticas visuales y gráficos de series temporales para seguir la evolución de la duración y desplazamiento de cada minitérmino.
- La importancia del dashboard radica en proporcionar a ingenieros, ergonomistas y responsables de producción una herramienta interactiva para:
  - Diagnosticar de forma visual y rápida las ineficiencias espaciales y temporales.
  - Validar y contextualizar la información obtenida del análisis automático.
  - Facilitar la toma de decisiones para optimizaciones posteriores (fase de simulación y optimización).

#### Salida Esperada:

- **Base de datos estructurada con registros de minitérminos dinámicos:**
  - Coordenadas, tiempos, desplazamientos.
  - Id. operario y coche.
  - Contexto o etiquetas de tarea si se pudiesen identificar.

- Visualizaciones gráficas (mapas de calor, rutas, histogramas) que apoyen la interpretación operativa y la mejora continua.
- Infraestructura sólida para alimentar con precisión las fases posteriores de simulación y optimización.

#### Tecnologías/Librerías Python:

- **OpenCV:** Procesamiento de imágenes, calibración.
- **TensorFlow / PyTorch:** Modelos de detección, seguimiento, pose y reconocimiento de actividad.
- **DB:** Para almacenar los datos estructurados de minitérminos y patrones de movimiento.

#### Datos de Entrada para el Rebalanceo en la Literatura

##### 1. Datos de la Línea de Montaje y Estaciones:

1. **Tipo de línea:** Recta, en U o de dos lados (recta, en U, de dos lados). [9]
2. **Número de estaciones:** Cantidad de estaciones de trabajo disponibles en la línea. [9]
3. **Configuración de líneas paralelas:** En sistemas PAL (Parallel Assembly Lines), se considera la disposición de múltiples líneas en paralelo. [10]
4. **Estaciones comunes:** Identificación de estaciones compartidas entre líneas paralelas. [10]

##### 2. Datos de Tareas:

1. **Lista de tareas:** El conjunto de todas las operaciones que deben realizarse. [12]
2. **Relaciones de precedencia:** El orden en que las tareas deben ejecutarse. [10,12]
3. **Tiempos de tarea:**
  - **Determinísticos:** Tiempos fijos para cada tarea. [10]
  - **Estocásticos/Inciertos:** Tiempos de tarea modelados con una distribución de probabilidad o como un intervalo de valores. [13] Para el caso de incertidumbre, se usan el tiempo nominal y el límite superior del intervalo, así como la desviación máxima. [14]
  - **Dinámicos:** Tiempos de tarea que pueden cambiar con el tiempo debido a factores como el aprendizaje o la fatiga del trabajador. [14]
  - **Dependientes del trabajador:** El tiempo de ejecución de una tarea puede variar según el trabajador que la realice [11,12]

##### 3. Datos de Trabajadores:

1. **Número de trabajadores:** Cantidad total de personal disponible. [10]
2. **Habilidades de los trabajadores (heterogeneidad):** Considera que los trabajadores tienen diferentes capacidades y eficiencias para realizar tareas. [10,12,16]
3. **Curvas de aprendizaje y olvido (L&F):** Modelan cómo la eficiencia del trabajador cambia con la experiencia y el tiempo. [9]
4. **Velocidad de caminata del trabajador:** La velocidad a la que cada trabajador se desplaza físicamente. [15]
5. **Restricciones de asignación:** Por ejemplo, un subconjunto de trabajadores que no pueden ser asignados a estaciones comunes. [10]

##### 4. Datos de Espacio y Movimiento (para "Walking Times"):

1. **Tiempos de caminata entre líneas adyacentes:** El tiempo que un trabajador tarda en desplazarse entre diferentes líneas paralelas, especialmente en estaciones comunes. [12]

2. **Distancia entre líneas adyacentes:** Factor que influye en los tiempos de caminata. [14]

### **Diferenciación de la Data del Estado del Arte para el Rebalanceo de Líneas (ALRBP):**

El problema que abordamos es el Rebalanceo de Líneas de Ensamblaje (ALRBP), lo que implica que estamos trabajando con líneas ya operativas que requieren ajustes dinámicos. La literatura existente sobre ALRBP, aunque reconoce la necesidad de dinamicidad, a menudo se basa en datos agregados, pre-definidos o modelados teóricamente (como tiempos de tarea estocásticos con distribuciones preestablecidas o modelos de fatiga y aprendizaje). Muy pocos estudios incorporan la riqueza de datos dinámicos y en tiempo real recopilados directamente del comportamiento del operario en el puesto de trabajo, como lo propone esta solución.

Además, nuestra propuesta se diferencia sustancialmente al capturar y analizar en tiempo real los desplazamientos, movimientos y tiempos del operario **dentro de una misma estación de trabajo**, utilizando para ello una cuadrícula espacial que segmenta el área de operación en posiciones discretas.

Esta focalización en el nivel intraestación permite medir con precisión los **minitérminos**, los cuales representan los desplazamientos microescalares y tareas elementales que el operario realiza durante un ciclo de producción.

A diferencia de los enfoques tradicionales que manejan tiempos de caminata solo entre estaciones o líneas paralelas, en nuestro caso el análisis del walking time se refiere a los desplazamientos **entre casillas dentro de la cuadrícula intraestación**, proporcionando una granularidad y realismo significativo para la simulación y la optimización.

Este enfoque es especialmente relevante porque estudios recientes (como Baykasoglu et al., 2023) ya destacan la importancia de considerar tiempos y distancias de caminata en líneas paralelas o multi-línea, pero nuestra propuesta profundiza y complementa esta visión hacia el interior de la estación, facilitando la construcción de simulaciones más fidedignas y caminos de mejora concretos en ergonomía y eficiencia operativa a nivel micro.

### **2.2 Fase 2: Plataforma de Simulación (Gemelo Digital de la estación de Trabajo)**

Esta fase establece un "gemelo digital" **de la estación de trabajo piloto**, que se utilizará como un entorno de laboratorio virtual para evaluar y validar teóricamente las configuraciones de diseño propuestas.

**Objetivo:** Crear un modelo de simulación preciso y dinámico del puesto de trabajo que pueda replicar el comportamiento del operario y responder a la variabilidad de los minitérminos observados bajo diferentes configuraciones de layout.

#### **Componentes y Actividades Clave:**

##### **1. Modelado del Puesto de Trabajo (Conceptualización basada en Redes de Petri):**

- **Representación del Layer de Minitérminos:** El núcleo del modelo del puesto de trabajo se construirá como una Red de Petri.

- Las transiciones de esta Red de Petri representarán directamente los minitérminos identificados y medidos en la Fase 1 (ej., "Tomar componente A", "Ensamblar pieza", "Moverse a zona de herramientas").
- Los lugares de la Red de Petri representarán los estados intermedios o las precondiciones/postcondiciones de estos minitérminos (ej., "Operario en posición inicial", "Herramienta en mano", "Pieza A lista para ensamblar").
- Los tokens representarán las entidades que fluyen a través del proceso (ej., la pieza en proceso, la atención del operario).
- **Configuración del Layout:** El modelo de la Red de Petri será parametrizable para incorporar la disposición física de los elementos (herramientas, cestas, superficies de trabajo). La ubicación de estos elementos influirá directamente en los desplazamientos asociados a las transiciones (minitérminos).
- Variabilidad de Minitérminos (Parametrización desde Fase 1):
  - El modelo de simulación utilizará las distribuciones de tiempo y desplazamiento (promedios, desviaciones estándar, o incluso distribuciones completas) obtenidas de la Fase 1 para asignar duraciones y distancias estocásticas a cada transición (minitérmino) de la Red de Petri. Esto asegura que el gemelo digital refleje la variabilidad observada en la operación real.

## 2. Motor de Simulación (Implementación de la Red de Petri):

- El motor de simulación (ej., basado en SimPy o código Python personalizado) actuará como un "ejecutor" de la Red de Petri que describe el puesto de trabajo.
- Será capaz de simular el flujo de trabajo del operario a través del disparo de las transiciones (minitérminos) de la Red de Petri, respetando sus pre-condiciones y actualizando los tokens en los lugares.
- Calcula los desplazamientos del operario: Para cada disparo de una transición (minitérmino), el simulador calculará los desplazamientos asociados en función de la posición de los elementos en el layout. Estos cálculos utilizarán los patrones de desplazamiento espacial obtenidos en la Fase 1 para cada mini-término.
- Permite la inyección de la variabilidad de los minitérminos en las duraciones de las actividades (transiciones), utilizando las distribuciones derivadas de la Fase 1.

## 3. Métricas de Rendimiento:

- Calcula métricas clave para evaluar el rendimiento de un diseño de puesto de trabajo propuesto:
  - **Tiempo de Ciclo Promedio del Puesto:** Tiempo total para que el operario complete su secuencia de tareas en la estación.



- **Desplazamiento Total del Operario:** Suma de las distancias recorridas por el operario (o sus manos) durante un ciclo de trabajo.
- **Eficiencia Ergonómica:** (Opcional, pero valioso) Podría integrarse un modelo simplificado para estimar la tensión o el esfuerzo basado en los movimientos.
- **Utilización de Recursos:** Tiempo activo vs. inactivo del operario.
- **Tiempos de Búsqueda/Acceso:** Tiempo que el operario tarda en alcanzar herramientas o componentes.

#### 4. Capacidad de Evaluación de Layouts:

- La simulación debe poder recibir una configuración de layout del puesto de trabajo como input y devolver sus métricas de rendimiento simuladas.

#### Salida Esperada:

- **Gemelo Digital Funcional de la estación de Trabajo:** Un modelo de simulación interactivo y configurable que puede simular el comportamiento del operario para cualquier diseño de puesto de trabajo.
- **Reportes de Rendimiento Simulados:** Para cada configuración de layout del puesto evaluada, la simulación proporcionará un conjunto de métricas de rendimiento detalladas.

#### Tecnologías/Librerías Python: (Tentativo)

- **SimPy:** Librería principal para el motor DES.
- **Custom Python Code:** Para implementar la lógica del puesto y las interacciones espaciales, siguiendo la estructura de lugares, transiciones y tokens de una Red de Petri, y utilizando los datos de la Fase 1 para parametrizar las transiciones.
- (Opcional: Librerías específicas para Redes de Petri como SNAKES, si se desea una implementación más formal del modelo PN dentro del simulador, aunque SimPy permite construir la lógica de forma más flexible).

#### Diferenciación de la Data y el Modelado frente al Estado del Arte:

La literatura tradicional sobre simulación de puestos de trabajo y balanceo de líneas a menudo se basa en modelos con parámetros estáticos o distribuciones de probabilidad teóricas (ej., tiempos de tarea con distribuciones normales o uniformes predefinidas, tiempos de desplazamiento calculados con fórmulas simples) [7,8].

Nuestra aproximación se distingue por el uso de **datos dinámicos de mini-términos** (Fase 1) para nutrir el gemelo digital, lo que nos permite operar en un "**Twin Continuum**" más avanzado [7,8]

- **Modelado de la Variabilidad Basado en Observación Real:** A diferencia de la parametrización teórica, nuestro gemelo digital se alimenta directamente de las distribuciones de tiempo y desplazamiento reales y observadas de los mini-términos. Esto permite que la simulación refleje con una fidelidad sin precedentes la variabilidad intrínseca del comportamiento humano y las condiciones operativas, incluyendo efectos

de fatiga, aprendizaje y micro-interrupciones, que son inherentemente dinámicos y difíciles de capturar con modelos fijos [15].

- **Enfoque en el "Physics-augmented Learning" para el Gemelo Digital:** Nuestro gemelo digital se alinea con el concepto de "Physics-augmented Learning". Si bien el modelo de Red de Petri proporciona una base "física" o de "reglas" claras (cómo fluyen las tareas, las precedencias), los datos dinámicos de la Fase 1 nos permiten:
  - **Aprender la Discrepancia (Ignorancia):** El gemelo digital, aunque formalmente estructurado (Red de Petri), utiliza datos empíricos para modelar la "ignorancia" o la discrepancia entre un modelo puramente teórico y el comportamiento real y variable del operario [15].
  - **Mejora Continua del Modelo:** Los datos de mini-términos se pueden usar para calibrar y refinar continuamente los parámetros estocásticos del gemelo digital, haciendo que sus predicciones sean cada vez más precisas y representativas del mundo real. Esto es fundamental para la credibilidad de las optimizaciones futuras y para potenciar la confiabilidad de un sistema de optimización.

### 2.3 Fase 3: Optimización Offline del Diseño del Puesto de Trabajo y Validación

Esta fase utiliza algoritmos de optimización para explorar diferentes configuraciones de diseño del puesto de trabajo, evaluándolas rigurosamente con el gemelo digital (Fase 2) y seleccionando la más óptima para su aplicación en la línea existente.

**Objetivo:** Encontrar la configuración física del puesto de trabajo (ubicación de cestas, herramientas, etc.) que optimice las métricas de rendimiento clave (ej., minimizar desplazamiento del operario, minimizar tiempo de ciclo), considerando la variabilidad dinámica de los minitérminos y el comportamiento adaptativo del operario.

#### Componentes y Actividades Clave:

##### 1. Algoritmos Genéticos (AG) para la Búsqueda de Configuraciones de Puesto:

- **Propósito:** El AG actúa como el motor de búsqueda y exploración para el vasto y complejo espacio de posibles diseños de puestos de trabajo. Dada la naturaleza combinatoria y a menudo no lineal de las relaciones entre la disposición de elementos y el rendimiento del operario, el AG es excepcionalmente adecuado para:
  - **Explorar Múltiples Escenarios:** Generar y evaluar un sinnúmero de combinaciones de ubicaciones de cestas, herramientas, y otros elementos dentro del puesto. Esto incluye explorar soluciones no intuitivas que podrían pasarse por alto en un diseño manual o con heurísticas simples.
  - **Manejar la Alta Dimensionalidad:** A medida que el número de elementos a posicionar y sus posibles coordenadas (X,Y) aumentan, el espacio de soluciones crece exponencialmente. El AG puede navegar eficientemente este espacio de alta dimensionalidad.

- **Optimización Multi-Objetivo:** Si se busca optimizar simultáneamente el tiempo de ciclo y el desplazamiento del operario (o incluso la ergonomía), los AG pueden encontrar un conjunto de soluciones de compromiso (frentes de Pareto).
- **Input:**
  - **Perfiles de Mini-Términos Dinámicos:** El AG utiliza las características estadísticas y las distribuciones de probabilidad de los mini-términos (obtenidas de la Fase 1, que capturan su naturaleza dinámica y variabilidad) para parametrizar el gemelo digital (Fase 2). Esto asegura que cada diseño de puesto propuesto por el AG sea evaluado en la simulación bajo condiciones realistas de variabilidad, donde el DRL interactuará con esa dinámica.
  - **Restricciones Físicas:** Dimensiones del puesto, zonas de acceso, zonas prohibidas, alturas mínimas/máximas, etc.
  - **Objetivos:** Definir qué se busca optimizar (ej., minimizar tiempo de ciclo, minimizar desplazamientos, mejorar ergonomía).
- **Ejecución:** El AG se ejecuta *offline* cuando se necesita optimizar el diseño de un puesto de trabajo.

## 2. Rol del Aprendizaje por Refuerzo Profundo (DRL) en la Evaluación de Aptitud:

- **Propósito:** El DRL se utiliza *dentro del simulador* (Fase 2) para realizar la evaluación de la función de aptitud del AG de la manera más realista y precisa posible.
- **Entrenamiento:** Para cada configuración de puesto de trabajo propuesta por el AG, un agente DRL es entrenado en el simulador para aprender la **política óptima de cómo el operario debería realizar la secuencia de mini-términos en ese layout específico**. Esto incluiría aprender la secuencia de movimientos más eficiente, cómo manejar la variabilidad de los mini-términos, etc.
- **Evaluación:** El rendimiento que el DRL logra en el simulador (ej., el mínimo desplazamiento total que puede lograr el operario, el tiempo de ciclo más rápido) se convierte en la métrica clave para evaluar la aptitud del diseño propuesto por el AG. Esto simula un comportamiento "óptimo" del operario bajo esa configuración del puesto.

### Salida Esperada:

- **Diseño de Puesto de Trabajo Óptimo:** Un conjunto de recomendaciones validadas para la disposición física de elementos en un puesto de trabajo.
- **Reporte de Rendimiento Proyectado:** Predicciones detalladas de cómo funcionará el puesto optimizado en la realidad (basado en el gemelo digital).
- **Mejora Continua del Proceso:** Un sistema iterativo para la optimización y validación de diseños de puestos de trabajo individuales.

### Tecnologías/Librerías Python:

- **AG: DEAP, PyGAD.**
- **DRL: Stable Baselines3, Ray RLlib.**
- **Herramientas de Visualización:** Para analizar los resultados del AG y del simulador (ej., Matplotlib, Plotly, herramientas de visualización 3D para layouts).

### **Alternativa: Optimización Directa con Aprendizaje por Refuerzo Profundo (DRL)**

En esta alternativa, el DRL asume el rol principal en la búsqueda y optimización de las configuraciones de diseño del puesto de trabajo.

#### **1. Agente DRL para Diseño de Puestos de Trabajo:**

- **Agente:** Se desarrolla un agente de Aprendizaje por Refuerzo Profundo (ej., utilizando algoritmos como PPO, SAC, o variantes adaptadas para espacios de acción discretos/combinatorios). Este agente será el encargado de aprender a generar y modificar diseños de puestos de trabajo.
- **Entorno de Entrenamiento:** El **simulador de la Fase 2 (Gemelo Digital del Puesto de Trabajo)** se convierte en el entorno de aprendizaje para el agente DRL. Es aquí donde el agente interactúa, propone diseños y recibe feedback.
- **Estado (State):** Una de las partes más críticas y desafiantes. El estado debe representar de manera efectiva la configuración actual del puesto de trabajo. Esto podría incluir:
  - Una representación espacial de la ubicación de todos los elementos (cestas, herramientas, superficies de trabajo).
  - La asignación de tareas al puesto y sus precedencias.
  - Métricas de rendimiento observadas hasta el momento en el diseño actual.
  - Perfiles de mini-términos relevantes para el puesto.
  - Esto a menudo requiere el uso de arquitecturas de redes neuronales avanzadas como **Redes Neuronales de Grafos (GNNs)** o representaciones basadas en cuadrículas/imágenes para capturar la estructura espacial y las relaciones entre los elementos.
- **Entrenamiento:** El agente DRL interactúa con el simulador, tomando acciones que modifican el diseño del puesto. Después de cada acción (o una secuencia de ellas, que define un diseño completo), el simulador ejecuta una simulación del puesto, y el agente recibe una recompensa basada en el rendimiento simulado. A través de millones de estas interacciones, el agente aprende una política que lo lleva a generar diseños de puestos de trabajo que maximizan la recompensa acumulada.

#### **2. Desafíos de la Optimización Directa con DRL:**

- **Diseño Complejo del Espacio de Estado y Acción:** Representar eficazmente la geometría y las relaciones espaciales de un layout como un estado DRL y definir

acciones que permitan modificaciones de diseño significativas es un desafío de investigación activo y requiere una ingeniería de características considerable.

- **Recompensas Escasas y Retrasadas:** La recompensa (el rendimiento general del puesto) solo se conoce después de una simulación completa del diseño propuesto, lo que dificulta significativamente el aprendizaje del agente.
- **Coste Computacional Elevado:** El entrenamiento del agente DRL puede ser extremadamente intensivo en recursos computacionales, ya que cada acción o secuencia de acciones requiere una evaluación en el simulador, lo que puede llevar mucho tiempo.
- **Manejo de Restricciones Duras:** Asegurar que el agente DRL genere diseños *válidos y factibles* (ej., no superponer objetos, mantener el acceso a herramientas) es difícil. A menudo se recurre a penalizaciones en la función de recompensa, pero esto puede ralentizar el aprendizaje o llevar a soluciones subóptimas.
- **Interpretabilidad:** Entender *por qué* el agente DRL propone un diseño particular puede ser un desafío, lo que dificulta la validación por parte de los expertos humanos y la confianza en la solución.
- **Generalización:** La capacidad del agente para optimizar diseños de puestos de trabajo que son significativamente diferentes de los que vio durante el entrenamiento es un reto clave.

#### Salida Esperada:

- **Diseño de Puesto de Trabajo Óptimo:** Un conjunto de recomendaciones validadas para la disposición física de elementos en un puesto de trabajo.
- **Reporte de Rendimiento Proyectado:** Predicciones detalladas de cómo funcionará el puesto optimizado en la realidad (basado en el gemelo digital).
- **Mejora Continua del Proceso:** Un sistema iterativo para la optimización y validación de diseños de puestos de trabajo individuales.

#### Tecnologías/Librerías Python:

- **DRL: Stable Baselines3, Ray RLlib, PyTorch / TensorFlow.**
- **Herramientas de Visualización:** Para analizar los resultados del DRL y del simulador (ej., Matplotlib, Plotly, herramientas de visualización 3D para layouts).

#### Diferenciación con el Estado del Arte en Optimización de Layouts y Rebalanceo:

Históricamente, la optimización de layouts de puestos de trabajo y problemas similares de asignación ha recurrido ampliamente a metaheurísticas como los Algoritmos Genéticos (AG). Estos son ampliamente reconocidos por su efectividad en explorar espacios complejos de soluciones y manejar múltiples restricciones, siendo una opción robusta para la búsqueda de configuraciones óptimas [14,15,16,17].

Más recientemente, el Aprendizaje por Refuerzo Profundo (DRL) ha emergido como una alternativa potente, especialmente para problemas dinámicos y secuenciales. Estudios recientes evidencian que las estrategias basadas en DRL pueden superar a los heurísticos convencionales, incluyendo AG, en productividad, reducción del tiempo de ciclo y mejor balance de cargas, sin necesidad de reentrenamiento constante para cada nueva condición [16].

Sin embargo, la aplicación directa y autónoma de DRL como optimizador offline de layouts presenta desafíos importantes:

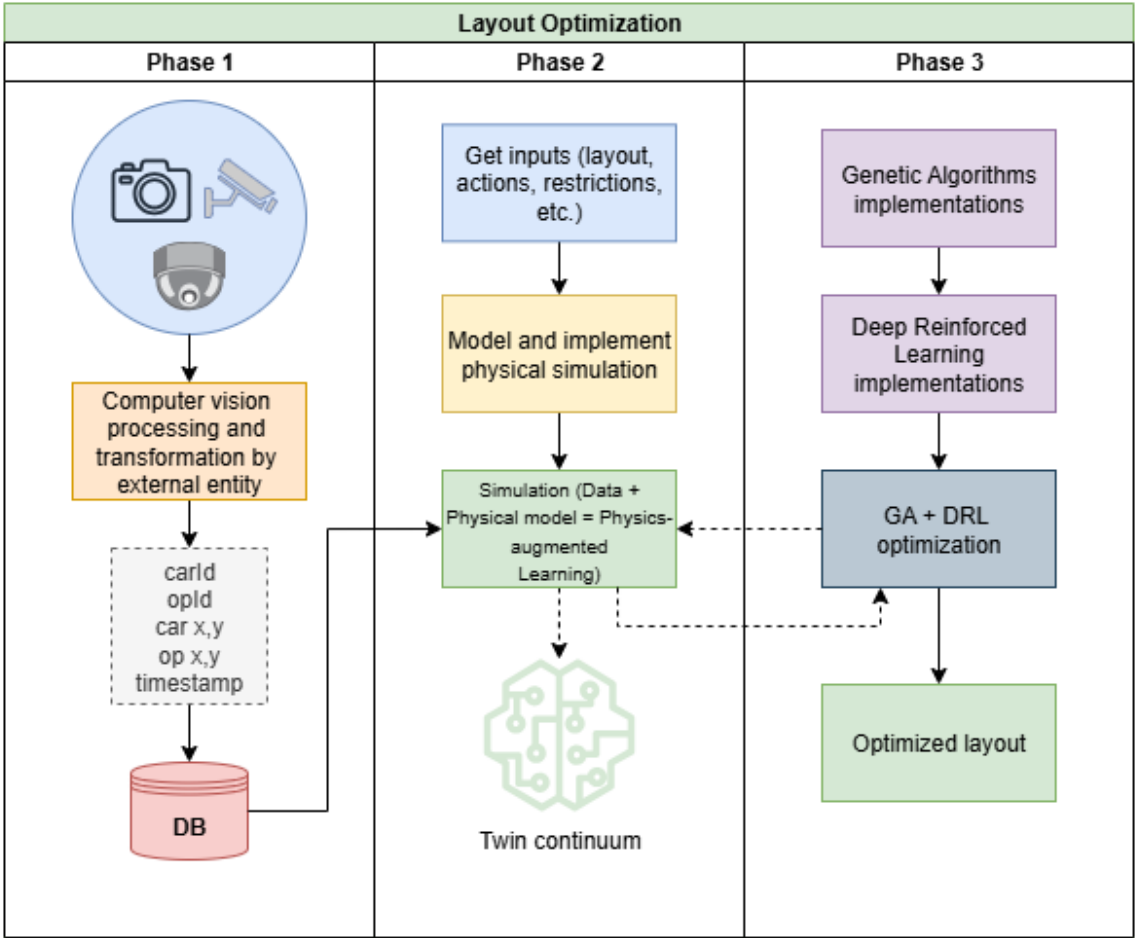
- La complejidad del espacio de diseño y la dificultad de representar el layout como estado para DRL, con un espacio de acciones extremadamente grande e intrincado [13].
- Elevados costos computacionales durante el entrenamiento, debido a la necesidad de simular exhaustivamente cada modificación de diseño propuesta [13].
- La dificultad en manejar restricciones duras para generar diseños válidos y factibles dentro del aprendizaje DRL [12].

Nuestra propuesta híbrida AG + DRL está diseñada para explotar las fortalezas de ambos métodos y mitigar sus limitaciones. Los AG funcionan como exploradores robustos de configuraciones posibles en un espacio combinatorio de alta dimensión, buscando soluciones óptimas y creativas [6]. Mientras tanto, el DRL se integra en el gemelo digital para evaluar con realismo la aptitud de cada diseño generado por el AG, aprendiendo la política óptima para operar cada layout considerando la variabilidad dinámica de los minitérminos observados [10,11].

Además, el DRL simula el comportamiento adaptativo del operario, capturando cómo un sistema inteligente reaccionaría ante cambios y condiciones reales, superando las limitaciones de las funciones de aptitud heurísticas tradicionales de los AG [12].

De este modo, la propuesta ofrece una solución más robusta y aplicable para la optimización offline del layout que cualquiera de los métodos por separado, con un balance entre exploración y evaluación dinámica [10,12].

1.4 Esquema



## Referencias

- [1] Ahmad, A. , Osman, S.A., Azhar M., Jamaludin, M.H., Abu Bakar, H.A., Rahman,M.I.A.R., Kiat, T.S., ***A Comprehensive Review: Analysing the Pros and Cons of Assembly Line Balancing Methods***. Journal of Advanced Research in Applied Sciences and Engineering Technology 44, Issue 2 (2025) 72-88.
- [2] Peinado-Asensi, I., Montés, N., Ibañez, D., García, E. Industrializable Industrial Internet of Things (IIoT) for a massive implementation of industry 4.0 applications: a press shop case example, Vol 63, Is 12 2025
- [3] Lacasa, A.; Llopis, J.; Montes, N.; Peinado-Asensi, I.; Garcia, E. **Cross-PLC: An IIoT Cross Platform to Manage Communications for Applications in Real Factories. Sensors. 25 (10): 2973- - 2025-05-08 25(10),**
- [4] E Garcia, N Montés, J Llopis, A Lacasa. **Miniterm, a novel virtual sensor for predictive maintenance for the industry 4.0 era. Sensors 2022, 22(16), 6222.**
- [5] Fink, C., Bodin, U., & Schelén, O. (2025). Why decision support systems are needed for addressing the theory-practice gap in assembly line balancing. *Journal of Manufacturing Systems*, 79, 515–527. <https://doi.org/10.1016/j.jmsy.2025.01.019>
- [6] Aguilar, H., García-Villoria, A., & Pastor, R. (2020). A survey of the parallel assembly lines balancing problem. *Computers and Operations Research*, 124, 105061. <https://doi.org/10.1016/j.cor.2020.105061>
- [7] Álvarez-Miranda, E., Pereira, J., & Vilà, M. (2023). Analysis of the simple assembly line balancing problem complexity. *Computers & Operations Research*, 159, 106323. <https://doi.org/10.1016/j.cor.2023.106323>
- [8] Yang, C., Gao, J., & Sun, L. (2013). A multi-objective genetic algorithm for mixed-model assembly line rebalancing. *Computers & Industrial Engineering*, 65(1), 109–116. <https://doi.org/10.1016/j.cie.2012.09.006>
- [9] Çimen, T., Baykasoğlu, A., & Demirkol Akyol, S. (2022). A detailed review and analysis of assembly line rebalancing problems. *Assembly Automation*, 42(6), 742–760. <https://doi.org/10.1108/AA-02-2022-0031>
- [10] Llopis, J., Lacasa, A., Garcia, E., Montés, N., Hilario, L., Vizcaíno, J., Vilar, C., Vilar, J., Sánchez, L., & Latorre, J. C. (2022). Manufacturing Maps, a Novel Tool for Smart Factory Management Based on Petri Nets and Big Data Mini-Terms. *Mathematics*, 10(14), 2398. <https://doi.org/10.3390/math10142398>
- [11] Pérez-Wheelock, R. M., Ou, W., Yenradee, P., & Huynh, V.-N. (2022). A Demand-Driven Model for Reallocating Workers in Assembly Lines. *IEEE Access*, 10, 80300–80320. <https://doi.org/10.1109/ACCESS.2022.3194658>
- [12] Falcó, A., Hilario, L., Montés, N., Mora, M. C., & Nadal, E. (2021). Towards a Vector Field Based Approach to the Proper Generalized Decomposition (PGD). *Mathematics*, 9(1), 34. <https://doi.org/10.3390/math9010034>
- [13] Çimen, T., Baykasoğlu, A., & Akyol, S. D. (2022). Assembly line rebalancing and worker assignment considering ergonomic risks in an automotive parts manufacturing plant.



International Journal of Industrial Engineering Computations, 13(3), 363–384.  
<https://doi.org/10.5267/j.ijiec.2022.2.001>

[14] Çimen, T. (2023). Rebalancing Manufacturing Lines by Meta-Heuristic Methods [Ph.D. thesis, Dokuz Eylül University]. <https://ssrn.com/abstract=5183856>

[15] Nikkerdar, M., ElMaraghy, W., & ElMaraghy, H. (2025). Smart adaptable assembly line rebalancing based on reinforcement learning. The International Journal of Advanced Manufacturing Technology. <https://doi.org/10.1007/s00170-025-15780-9>

[16] Baykasoglu, A., Himmetoglu, S., & Delice, Y. (2023). Robust parallel assembly line balancing with heterogeneous workers considering walking times: A mathematical model and hybrid artificial bee colony algorithm. Computers & Operations Research, 159, 106323. <https://doi.org/10.1016/j.cor.2023.106323>

[17] Dalle Mura, M., & Dini, G. (2023). Sensitivity analysis and validation of a genetic approach to enhance ergonomics in assembly lines. Production Engineering. <https://doi.org/10.1007/s11740-023-01219-1>

*Alicia López Castellano*

oct.-08-2025

*Carlos Moliner*

oct.-08-2025

*Nicolas Montes*

oct.-02-2025



oct.-07-2025