

Robots in an Abandoned Warehouse

David Alejandro Lozano Arreola - A01722728, Gerardo Leiva Diaz - A01198479, Jorge Betanzo Carriles - A01198676, Alejandro Hernández Villaseñor - A01571408

Equipo 8, TC2008B.302

Tecnológico de Monterrey,

Monterrey 64700, Mexico,

E-mails: {A01722728, A01109479, A01198676, A01571408}@tec.mx

Resumen – Este problema se enfoca en la solución de un sistema multiagentes para un almacén automatizado mediante robots que se trasladan, reconocen, y transportan objetos de varios pesos a distintos lugares de origen y destino. Mediante esta solución, nosotros simularemos lo antes mencionado en un ambiente digital, de manera que pueda ser aplicado a los robots físicos de manera eficiente.

Índice de términos – Python, Agente, Multiagentes, Sistema Multiagentes, Unity, Simulación, Modelado Gráfico, Vehículo Guiado por Láser (LGV), Estado de carga (SoC).

0. Equipo y logística

Análisis Individual: Identifiquen y describan las fortalezas y áreas de oportunidad de cada miembro del equipo.

Gerardo Leiva - *Fortalezas:* responsabilidad, liderazgo, compromiso. *Áreas de oportunidad:* organización, manejo de tiempos.

David Lozano - *Fortalezas:* liderazgo, conocimiento técnico, organización. *Áreas de oportunidad:* comunicación, repartición de tareas.

Jorge Betanzo - *Fortalezas:* Versatilidad, compromiso y proactividad. *Áreas de oportunidad:* Falta en concentración mental y formación en el ámbito escolar.

Alejandro Hernández - *Fortalezas:* resolución de problemas, implementación de tecnologías emergentes. *Áreas de oportunidad:* Proactividad y compromiso.

Expectativas y Compromisos: Escriban un breve listado de expectativas individuales y grupales para el bloque, además de los compromisos que asumirán como equipo.

Gerardo Leiva - Algunas expectativas que tengo respecto al bloque es poder aprender sobre los sistemas de agentes de una manera que me permita entender sus comportamientos y funciones.

David Lozano - Lo que busco en este curso es poder aprender sobre la Inteligencia Artificial a manera que pueda entender cómo se construyen, cómo responden a diferentes variantes, y sus procesos internos.

Jorge Betanzo - Para este periodo escolar, dentro de la materia de modelación de sistemas, espero aprender la lógica detrás de diversos sistemas inteligentes logísticos, esto me otorga una manera de pensar óptima en el ámbito de programación.

Alejandro Hernández Villaseñor - Quiero aprender más de la conexión entre controlador y máquina y sobre cómo automatizar robots para transformarlos en agentes inteligentes. Me comprometo a cumplir con mis tareas y no ser una carga a los demás.

0-A. Herramientas de trabajo:

Repositorio GitHub:

<https://github.com/Tect0r0/multiagentes>

Mecanismos de Comunicación: Grupo de Whatsapp.

I. Introducción

I-A. Contexto y Problema

Fuimos contactados por una empresa para automatizar y eficientar los traslados de robots llamados “vehículos guiados por láser” (LGV por sus siglas en inglés) a lo largo de sus funciones. Las funcionalidades de los LGVs incluyen: desplazamiento dentro del almacén, traslado de materiales, ciclos de carga, etc. Estos robots tienen capacidades de traslado, tiempos de batería, velocidad, dirección de movimiento, sensores para detectar sus alrededores, entre otros parámetros, que deberán ser monitoreados al igual que manejados para tener la mejor eficiencia al momento de tratar con ellos. Los LGVs pueden también detectar otros a lo largo de su camino, y elegir los caminos más eficientes para evitar choques, congestiones, etc.

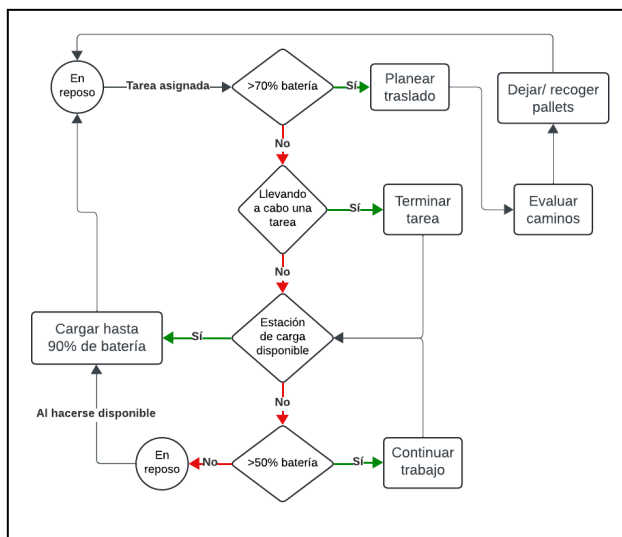


Figura 1: Diagrama de estados del LGV.

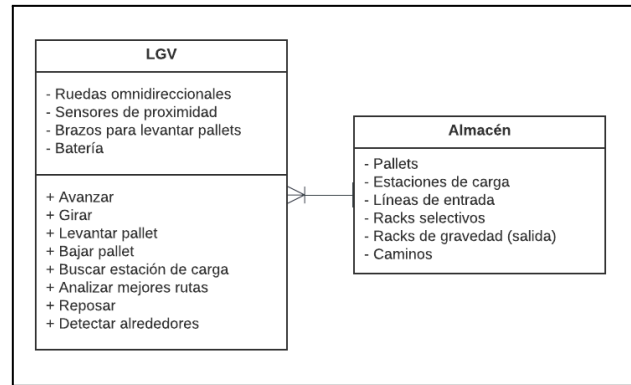


Figura 2: Diagrama de clases.

Adicionalmente, los almacenes cuentan con sus propias especificaciones, como lo son los requerimientos por parte del cliente para el flujo, así como sus características físicas las cuales incluyen la estructura del almacén, las posiciones de las estanterías a llenar, cuántos productos pueden almacenar cada estantería, etc.

I-B. Objetivos generales

En este proyecto se busca lograr varios objetivos mediante la solución. A continuación se mencionan:

- Correctamente simular a un agente, su comportamiento, sus decisiones, de manera que podamos modificar sus comportamientos, en las primeras etapas del proyecto.
- Implementar un sistema de control para decirles a los agentes cómo actuar en base a distintos parámetros, implementado tan pronto como se implementan los agentes individuales.
- Implementación de un sistema multiagentes, en el que interactúen varios agentes al mismo tiempo, pudiendo tomar las mejores decisiones en cuanto a su entorno y las acciones de los otros agentes. Lograr esto después del sistema de control individual.
- Simular todo lo anterior digitalmente para poder tener un sistema completamente funcional para el final del curso.

Modelación de sistemas multiagentes con gráficas computacionales (Gpo 302) TC2008B.302

I-C. Restricciones

Este proyecto cuenta con varias restricciones dada la complejidad de su implementación. Desde el entorno, hasta los propios robots; ambas secciones contienen sus propias limitaciones:

Entorno (flujos):

- Racks de entrada a racks selectivos: 20 pallets por hora.
- Racks selectivos a racks de gravedad (salida): 25 pallets por hora.
- Racks de entrada a racks de gravedad (salida): 10 pallets por hora.

Robots (LGVs):

- Tarda 30 segundos en recoger un pallet desde el rack de entrada.
- Tarda 60 segundos en depositar o recoger un pallet en un rack selectivo.
- Gasta 20% de batería por hora mientras está en movimiento, y 5% por hora cuando está en descanso.
- Se carga en una estación a un ritmo de 20% cada 5 minutos.
- El robot es mandado a cargar (si hay un cargador disponible) cuando su estado de carga (SoC) llega a 70%, y deja de cargarse al llegar a 90%.
- Si un robot llega a un SoC de 50% o menos, no puede llevar a cabo ningún trabajo y se mantendrá en reposo hasta que se haga disponible una estación de carga.
- Un LGV no puede ir a cargar si está en proceso de transportar un pallet.

I-D. Resumen de la solución propuesta

La solución que proponemos consiste en una simulación utilizando la herramienta de Unity 3D para desplegarse, y utilizando Python para la lógica, que tome en cuenta varios parámetros para manejar un sistema multiagentes donde los robots o LGVs interactúen con el entorno mediante el transporte de recursos de un punto a otro. Esta simulación tendrá en

cuenta cosas como las baterías de los robots, los flujos esperados de los pallets, los traslados de los LGVs, y tomas de decisiones en cuanto a las mejores rutas a tomar y problemas emergentes similares.

Los LGVs podrán trasladar los pallets de la línea de entrada a los racks selectivos, de estos a los racks de gravedad/salida, y también de la entrada a la salida directamente con sus flujos designados. Los LGVs también podrán tomar decisiones en cuanto a las rutas que tomen basados en los otros agentes, así como manejar sus flujos de batería.

II. Fundamentos

II-A. Q-learning

Q learning es un algoritmo de aprendizaje por refuerzo, esto le permite a un agente encontrar la acción más favorable en el estado actual para maximizar la recompensa a largo plazo. El algoritmo es de modelo libre, lo que significa que el agente no tendrá que tener la información del entorno de antemano, por lo que el agente recopila información conforme explore.

En cada paso, el agente utiliza la siguiente ecuación para actualizar su estimación de $Q(s, a)$:

$$Q(s, a) \leftarrow Q(s, a) + \alpha(r + \gamma \max_a Q(s', a') - Q(s, a))$$

II-B. Ecuación Bellman

La ecuación de Bellman es una ecuación recursiva, el aprendizaje es por refuerzo, basándose en el valor de estado actual y futuro. Descompone el problema de encontrar la política óptima en partes más manejables.

La ecuación de Bellman para el valor óptimo de un estado s es dado por la siguiente:

$$V(s) = \max(r(s, a) + \gamma V(s'))$$

Modelación de sistemas multiagentes con gráficas computacionales (Gpo 302) TC2008B.302

Donde $V(s)$ es el valor óptimo del estado s , $r(s, a)$ es la recompensa al tomar la acción a en el estado s y $V(s')$ es el valor óptimo del siguiente estado.

II-C. Entornos Multiagente

Los entornos multiagentes se refieren a ellos en los que diversos agentes pueden interactuar, en muchas de las ocasiones causando que las decisiones que uno toma afecte a los demás, por lo general, se tienen estos principales entornos de multiagentes:

- **Cooperativo:** Todos los agentes tienen un objetivo común, por lo que normalmente se coordinan para maximizar la recompensa global.
- **Competitivo:** Los agentes tienen objetivos que se superponen con los de los demás por lo que necesitan anticipar y responder a las estrategias de los demás agentes.
- **Mixto:** Los agentes pueden tener objetivos individuales, los cuales pueden estar alineados o en conflicto con los de otros.

II-D. Política de Acción

Define como el agente selecciona sus acciones en cada estado, hay dos tipos, estas siendo:

- **Determinística:** Cada estado mapea una acción específica.
- **Estocástica:** Distribuye la probabilidad sobre acciones por cada estado, lo que permite una mayor exploración.

El objetivo del agente es aprender la política óptima que le permita maximizar su recompensa a lo largo del tiempo. En Q-learning, la política es implícita en la función Q ya que el agente elige la acción con el valor Q más alto en cada estado.

II-E. Exploración y Explotación

En sistemas multiagentes, el agente debe decidir entre explorar nuevas acciones (exploración) o tomar

acciones conocidas (explotación) para maximizar su recompensa a corto plazo.

- **Exploración:** Le permite al agente hacer nuevas acciones para descubrir si hay mejores opciones que maximicen la recompensa final.
- **Explotación:** El agente, en base a la información ya recopilada, maximiza su recompensa.

Un método muy conocido para balancear el uso de las dos en Q-learning sería el método de epsilon-greedy, en el cual, con una probabilidad de ϵ , el agente alterna y elige entre explorar y explotar.

III. Descripción del Sistema Multiagente

III-A. Modelo de los Agentes

Creencias:

- Estado de carga actual (SoC).
- Ubicación dentro del almacén.
- Estado de ocupación (libre/ocupado transportando).
- Conocimiento de otros LGVs cercanos.

Planes:

- Plan de ruta óptima basado en el lugar de destino.
- Plan de recarga cuando el SoC baja del 70%.
- Plan de evitación de colisiones.
- Planificación de recolección y entrega de pallets.

Cooperación:

- Comunicación de rutas planeadas con otros LGVs.
- Coordinación para evitar congestión en puntos de carga.
- Negociación de prioridades en intersecciones.
- Balanceo de carga de trabajo entre agentes.

Aprendizaje:

Modelación de sistemas multiagentes con gráficas computacionales (Gpo 302) TC2008B.302

- Aprendizaje de rutas óptimas usando Q-learning.
- Adaptación a patrones de tráfico.
- Mejora continua en la estimación de tiempos de entrega.
- Optimización del uso de batería.

III-B. Modelo del Entorno

Describe el entorno de la simulación: observable o parcialmente observable, determinista o estocástico, etc. Mencione cómo se maneja el tiempo y el espacio.

Características:

- **Parcialmente Observable:** Los LGVs solo pueden detectar otros agentes y obstáculos dentro de su rango de sensores.
- **Estocástico:** Hay elementos aleatorios como tiempos de variables de carga/descarga.
- **Secuencial:** Las acciones actuales afectan estados futuros.
- **Dinámico:** El entorno cambia mientras los agentes deliberan.
- **Continuo:** Las posiciones y velocidades varían continuamente.

Manejo del tiempo:

- Simulación en tiempo discreto con intervalos fijos.
- Sincronización de acciones entre agentes.
- Gestión de eventos prioritarios.

Manejo del espacio:

- Grid basado en coordenadas para navegación.
- Zonas definidas: carga/descarga, estaciones de recarga, rutas principales.
- Representación de obstáculos y áreas restringidas(límites).

III-C. Modelo de la Negociación

Tipos de mensajes:

- Solicitudes de paso en intersecciones.
- Avisos de inicio/fin de tareas.
- Alertas de bajo SoC.
- Coordinación de uso de estaciones de carga.

Protocolos de negociación:

- Subastas para asignación de tareas nuevas.
- Votación para resolver conflictos de ruta.
- Negociación basada en prioridades para acceso a recursos compartidos.

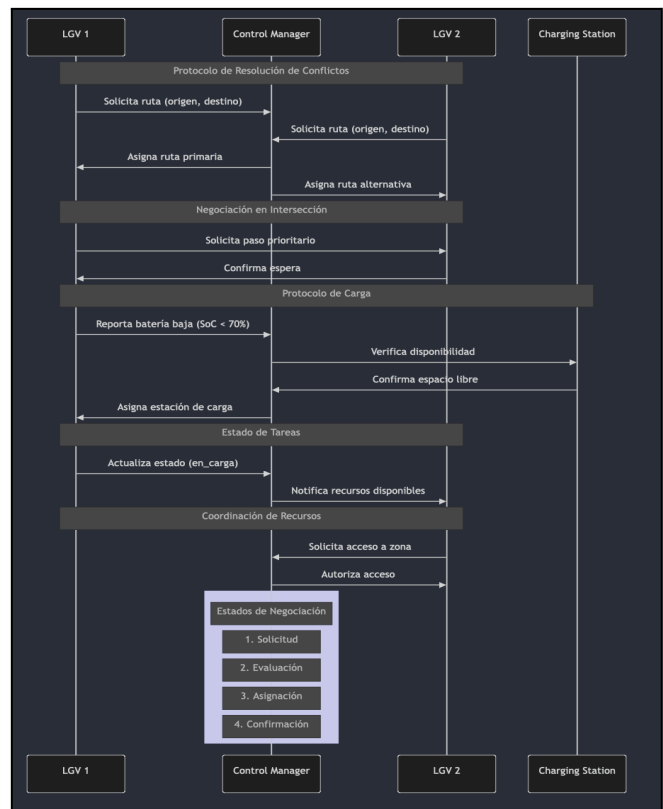


Figura 3: Diagrama de protocolos de información y comunicación entre agentes y el entorno.

III-D. Modelo de la Interacción

El sistema estará conectado con la simulación gráfica hecha en Unity de manera que lo que piense y actúe, se verá reflejado en ésta. Por ejemplo, si un

Modelación de sistemas multiagentes con gráficas computacionales (Gpo 302) TC2008B.302

agente se traslada hacia una estación de carga, el programa informa que la acción desea ser tomada. Después de esto, el programa manda esta información a Unity, y dentro de este, se encarga de desplegar visualmente la acción de alguna manera, como una figura moviéndose de un punto a otro. Eso nos da una manera de poder ver y analizar el comportamiento del agente, así como su interacción con el entorno, ya que Unity se encargará de la parte visual tanto para nosotros, como para el robot.

IV. Descripción del Modelado Gráfico

IV-A. Escena a Modelar

Agentes:

- **Carros de carga:** Agentes autónomos que siguen rutas predeterminadas o generan rutas en tiempo real, evitando obstáculos y otros carros. Cada carro puede estar basado en un algoritmo de navegación.

Entorno:

- **Bodega:** El entorno tiene una estructura de almacén con estantes, zonas de carga, descarga y caminos de tránsito.
- **Obstáculos:** Elementos estáticos y dinámicos dentro de la bodega, como estantes, columnas, otros carros de carga, y áreas restringidas.
- **Rutas:** Definición de caminos principales dentro de la bodega para el movimiento de los carros.

Escena:

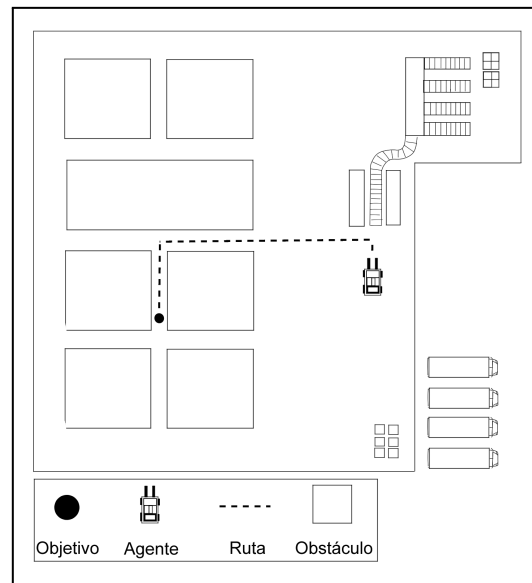


Figura 4: Diagrama de escena (borrador).

Comportamiento de agente:

- **Tareas:** Trasladar carga desde un punto A a un punto B.
- **Decisiones:** Decidir su ruta, tomar decisiones ante obstáculos y seguir las instrucciones del sistema.
- **Comportamiento con otros agentes:** Evitar colisiones, ajustar su velocidad con otros agentes o manejar el viaje a un punto para no crear tráfico.

Ecuación de movimiento:

$$f(n) = a(n) + b(n)$$

Donde $f(n)$ es el costo final estimado para llegar del inicio al objetivo, $a(n)$ es el costo desde el inicio al nodo n , y $b(n)$ es el costo estimado en llegar del nodo n al final.

V. Algoritmos utilizados

[Por agregar]

VI. Administración del proyecto

VI-A. Medios de comunicación

Project backlog:

<https://github.com/Tect0r0/multiagentes/issues>

Project sprint: **[Por agregar]**

VI-B. Plan de trabajo

Tareas pendientes:

- Simulación de la escena (2 - 3 hrs).
- Modelación del cuerpo del agente (2 - 3 hrs).
- Creación de la lógica del agente (3 - 5 hrs).
- Conexión entre los agentes (1 - 2 hrs).
- Manejo de agentes (2 - 3 hrs).
- Implementación de agentes en escena (1 - 2 hrs).

Actividades priorizadas para la primera revisión:

- **Creación del documento:** Todos los integrantes. (11 de Noviembre). [Esfuerzo alto].
- **Inicios de la simulación de escena:** Gerardo Leiva y Jorge Betanzo. (11 de Noviembre). [Esfuerzo moderado].
- **Inicios de la lógica del agente:** David Lozano y Alejandro Hernández. (11 de Noviembre). [Esfuerzo moderado].

VI. Resultados

[Por agregar]

VII. Aprendizaje Adquirido

VII-A. Avance 1

Para este primer avance, nos enfocamos principalmente en el aspecto teórico como lo es aprender sobre lo que son los agentes, lo que son los entornos y las diferentes maneras en las que los

agentes interactúan con el entorno y entre ellos. Adicionalmente, hicimos la base principal de todo, y logramos establecer un plan de trabajo para poder ir desarrollando nuestra solución en torno a lo que vayamos aprendiendo durante el curso.

IX. Conclusión

[Por agregar]

X. Trabajo Futuro

[Por agregar]

Referencias