UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

EXCAVACIÓN AUTÓNOMA CON LHD UTILIZANDO APRENDIZAJE REFORZADO PROFUNDO

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN ELÉCTRICA

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

RODRIGO ANDRÉS SALAS OSORIO

PROFESOR GUÍA:
JAVIER RUIZ DEL SOLAR

PROFESOR CO-GUÍA:
FRANCISCO LEIVA CASTRO

MIEMBROS DE LA COMISIÓN:
MARCOS ORCHARD CONCHA
EDUARDO MORALES MANZANARES

Este proyecto ha sido parcialmente financiado por Proyecto FONDECYT 1201170, Proyecto ANID-PIA AFB220002 y Proyecto ANID-PIA AFB230001

SANTIAGO DE CHILE
2024

RESUMEN DE LA TESIS PARA OPTAR AL GRADO
DE MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN ELÉCTRICA
Y MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO
POR: RODRIGO ANDRÉS SALAS OSORIO
FECHA: 2024
PROF. GUÍA: JAVIER RUIZ DEL SOLAR

EXCAVACIÓN AUTÓNOMA CON LHD UTILIZANDO APRENDIZAJE REFORZADO PROFUNDO

La transición hacia una minería subterránea cada vez más profunda requiere automatizar la maquinaria para poder operar en condiciones demasiado peligrosas para operadores humanos. Este trabajo aborda el problema de excavación de material con máquinas *Load Haul Dump* (LHD) mediante el uso de aprendizaje reforzado profundo. El controlador es entrenado utilizando el algoritmo DDPG, únicamente en simulación y sin demostraciones previas ejecutadas por expertos. El diseño de la recompensa busca incentivar al agente a cargar la mayor cantidad de material evitando el resbalamiento de las ruedas. Se propone una simulación de bajo costo computacional basada en un modelo analítico, que calcula las fuerzas ejercidas sobre el balde durante un carguío, para entrenar al agente. Múltiples experimentos en el mundo real muestran que la política aprendida alcanza resultados iguales o mejores en cantidad de material cargado y resbalamiento de ruedas, comparado con carguíos realizados por teleoperación y un algoritmo experto. Además, los resultados muestran que el diseño del sistema y la simulación utilizada proveen al agente de robustez frente a perturbaciones en observaciones del ambiente y a cambios en la granulometría del material.

Qué sabes tú de las piedras.

Agradecimientos

Quiero agradecer a mi familia que me han apoyado incondicionalmente toda mi vida y me han acompañado en todo momento. Les agradezco por guiarme y darme las herramientas para llegar a ser la persona que soy hoy, por ser un pilar de cariño y felicidad para alegrar todos los días.

A mi pareja Isabel, que ha estado conmigo toda mi carrera, que me ha acompañado en todo momento y me ha ayudado a superar cada obstáculo, haces que cada momento sea más lindo y que todo este tiempo haya sido lo mejor de mi vida.

Quiero agradecer al profesor Javier Ruiz del Solar por haberme recibido en el laboratorio y guiarme durante todo mi paso por la universidad, culminando en este trabajo. A Francisco e Isao por ayudarme en múltiples proyectos realizados siempre con la mejor disponibilidad y a todos los integrantes del laboratorio de robótica de campo del AMTC por alegrar los días de sacar resultados.

A mis cabros, Niko, Pulga, Noizy, Chuckles, Palli, Pit, Buti, Juan, JC, Nacho, Nils y Mulet, gracias por todos los buenos momentos que pasamos y poder contar con ustedes como una familia más.

A los del laboratorio de robótica, tanto aquellos que me recibieron y la nueva generación que me sigue, Gio, Lukas, Nico M, Cris, JP, Ulises, Pablo, Pipe, Gonzalo, Matti, Rona y Nacho, muchas gracias por todos los momentos compartidos disfrutando el laboratorio.

A mis amigos de la universidad, Iván, Cata, Flo, Robert, Martín, Mati, Coni, Crístian, Isa B, Alex, Pichún, Joaquín, Willy gracias por hacer de esta etapa en mi vida espectacular y por compartir tantas buenas historias.

Por último, quiero agradecer a los proyectos FONDECYT 1201170, ANID-PIA AFB220002 y ANID-PIA AFB230001 por brindar fondos que hicieron posible este trabajo.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción

1.1. Motivación

La demanda global por metales ha aumentado estos últimos años debido a factores como el crecimiento constante de la población y la transición a una economía mundial de bajo consumo de carbono [1, 2]. Este aumento en la demanda de metales, junto al agotamiento de los yacimientos superficiales y la búsqueda de una minería con menor impacto ambiental, son factores que impulsan a la minería hacia yacimientos subterráneos cada vez más profundos [3]. Extraer material de yacimientos más profundos genera un mayor riesgo para el personal que trabaja en terreno. El riesgo proviene principalmente de que se trabaja en lugares aislados y de difícil acceso, con maquinaria de gran tamaño en espacios reducidos y en ambientes con alta concentración de partículas suspendidas en el aire y gases resultantes de combustión [4].

Un Load-Haul-Dump (LHD) es un vehículo diseñado para transporte de material utilizado en la mayoría de las minas subterráneas a nivel global [5]. La Figura 1.1 [6] muestra un LHD utilizado en la minería. Son similares a los cargadores frontales utilizados en superficie pero con adaptaciones específicas para funcionar bajo tierra y tienen tres principales modos de operación: presencial, remoto y autónomo. La operación presencial corresponde a que el LHD es operado por un conductor ubicado dentro de la máquina. La operación remota corresponde a que el LHD es operado por un conductor ubicado fuera de la máquina, generalmente en una sala de operaciones donde tiene los controles y visión del entorno del LHD mediante múltiples cámaras montadas en la máquina. La operación autónoma corresponde a que el LHD es controlado por un programa especializado sin la intervención de una persona.

El diseño de cada parte del LHD está enfocado hacia la productividad y pensado para transportar material principalmente en cortas distancias, por lo que el ciclo de carga y descarga debe ser tomado en cuenta al momento de desplegar estas máquinas. En comparación con los cargadores frontales, los LHDs están diseñados para tener un perfil más largo, angosto y de baja altura, lo que complejiza la maniobrabilidad pero permite aumentar la capacidad del balde [5]. Un LHD tiene tres partes principales las cuales se pueden visualizar en la Figura 1.2 (adaptada de [7]): (i) el chasis trasero, (ii) el chasis delantero y (iii) el brazo que sostiene el balde. En el chasis trasero van montados la mayoría de los componentes del LHD como el motor, la transmisión, eje trasero de ruedas y la

Figura 1.1: Máquina LHD modelo R2900G producida por Caterpillar.

cabina del piloto. El chasis frontal sostiene el eje delantero de ruedas y el brazo hidráulico que sostiene el balde. Ambas partes están conectadas con una articulación hidráulica.

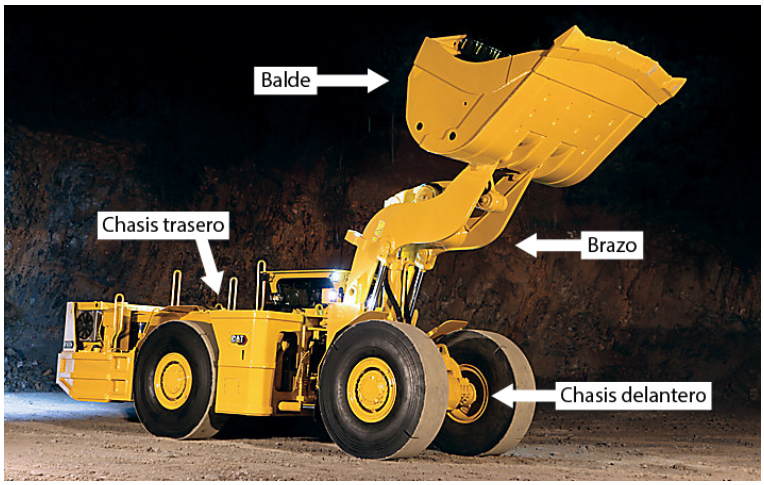

Figura 1.2: Diagrama de un LHD, adaptado de [7].

Como se detalla en [5], la mantención de las ruedas de un LHD representa generalmente hasta el 20 % de los costos operacionales de la máquina. Las ruedas están pensadas para tener un tiempo de vida de alrededor de 1000 horas. Este tiempo de vida se ve principalmente determinado por el desgaste que sufren los neumáticos durante la operación, el cual proviene principalmente del exceso de derrape y por pasar sobre rocas que generan cortes en el neumático. Se utilizan neumáticos macizos ya que permiten la reparación de los surcos superficiales y así mantener una mejor tracción durante la operación. Los LHDs cuentan además con tracción en las cuatro ruedas.

El control del balde es realizado mediante dos cilindros hidráulicos que controlan respectivamente el volteo y el levante. El brazo tiene una configuración de pantógrafo, por lo que al levantar el brazo, el ángulo de volteo se mantiene constante siempre y cuando no se acciones el volteo. Levantar y voltear el balde hacia arriba mientras se tiene la pala dentro de la pila es una técnica utilizada para aumentar la tracción de las ruedas y evitar el resbalamiento. Otra técnica para evitar el derrape es bajar la velocidad hasta que todas las ruedas dejen de derrapar, pero esta disminuye el momentum de la máquina y podría perjudicar el penetramiento en el punto de extracción.

Existen tanto LHDs eléctricos como LHDs con motores de combustión interna (normalmente motores diesel), estos últimos siendo los más utilizados en la industria. Aquellos LHDs con motores

diesel son utilizados cuando las distancias de acarreo necesitan ser mayores, puesto que brindan una mejor autonomía. Debido a que son utilizados principalmente en la minería subterránea (i.e. en espacios cerrados), los LHDs deben contar con una gran cantidad de elementos para tratar los gases de escape de la máquina y el lugar de trabajo debe seguir múltiples normas para mantener la seguridad de los operadores (e.g. el espacio debe ser ventilado, ciertos gases no deben superar límites de seguridad).

Los LHDs y cargadores frontales tienen un ciclo de operación comúnmente llamado *V-cycle* [8, 9], ilustrado en la Figura 1.3. Este nombre proviene de la trayectoria realizada por una máquina al ir hacia un punto de extracción, cargar material y luego depositarlo en una zona objetivo (e.g. camión de transporte, cinta transportadora). El ciclo de operación de un LHD puede ser divido en las siguiente etapas principales:

1. Navegar al punto de extracción.
2. Analizar el punto de extracción y escoger de dónde excavar material.
3. Ataca el punto de extracción y excavar material.
4. Navegar a la zona objetivo.
5. Depositar el material en el objetivo final.

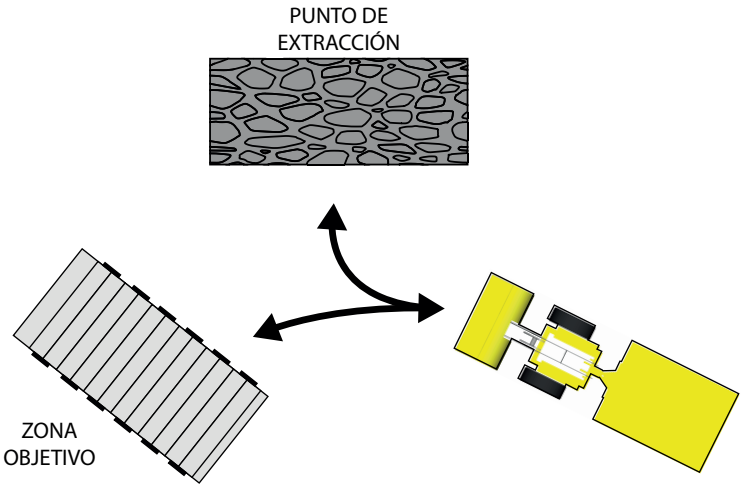

Figura 1.3: *V-cycle* de carguío con LHD.

La distancia de acarreo que debe realizar el LHD durante el ciclo de carguío determina el tamaño del balde [5]. Para realizar trayectos más largos, el tamaño del balde se puede incrementar acordemente. No obstante, aumentar la distancia a recorrer aumenta los costos de operación principalmente debido al desgaste de los neumáticos y al uso de combustible. Este balance entre distancia y tamaño del balde es uno de los principales factores que determinan las características del LHD a utilizar.

Para Chile, la minería es un pilar fundamental de su economía, representando el 13,6 % del PIB del país [10]. Chile es uno de los principales actores dentro de la producción mundial de cobre, siendo el mayor productor y exportador del mundo. Además, ocho de las veinte minas de cobre más grandes del mundo están ubicadas en Chile [11]. Entre ellas está Chuquicamata, una de las principales minas de cobre del país que comenzó una transición hacía la minería subterránea [12],

y el Teniente que es “el yacimiento de cobre subterráneo más grande del planeta” [13]. Por lo tanto, todos los avances en tecnologías y procesos mineros son relevantes para el país y su economía.

La operación autónoma responde tanto a la necesidad de suplir la creciente demanda al incrementar la productividad, como a la necesidad de mantener la seguridad de los trabajadores. Además, la automatización de procesos y equipos mineros es una de las soluciones para alcanzar la minería integrada e inteligente, como es indicado en [14]. Esta puede ser aplicada en cualquier etapa de la cadena de valor de la minería, debido al alto nivel tecnológico implementado en las diferentes partes del proceso. Múltiples empresas líderes en el rubro de producción de maquinaria minera ofrecen máquinas con algún tipo de automatización [15, 16, 17, 18, 19]. A pesar de esto, la industria minera es considerada de los rubros con menor nivel de digitalización en la industria [20].

En el último tiempo, ha habido un fuerte desarrollo de nuevos métodos y algoritmos basados en inteligencia artificial y aprendizaje de máquinas, impulsado por el avance en tecnologías que permiten el procesamiento de una mayor cantidad de datos, produciendo así algoritmos más potentes. Estos nuevos sistemas son alternativas atractivas para abordar la automatización debido a su potencial de superar a los sistemas tradicionales, que necesitan de conocimiento experto para la configuración de sus parámetros.

El aprendizaje reforzado o *reinforcement learning* (RL) es una rama del aprendizaje de máquinas que se enfoca en entrenar agentes que interactúan con un ambiente con el fin de maximizar una función de recompensa. La función de recompensa está diseñada para guiar al agente hacia un comportamiento adecuado. Una ventaja de esta metodología es que no necesita conocimiento experto para la generación de demostraciones debido a que el agente explora para llegar a una solución. Se tienen como ejemplo algunas soluciones basadas en RL que han resuelto exitosamente problemas en distintos campos de estudio, tales como la manipulación [21], navegación [22] o su aplicación en juegos como el Go [23].

Se necesita generalmente una gran cantidad de interacciones entre el agente y el ambiente durante el entrenamiento para llegar a una política capaz de resolver la tarea adecuadamente. Además, si se entrena al agente en el mundo real, las interacciones pueden poner en riesgo tanto al agente como su entorno, debido a la necesidad de ejecutar acciones exploratorias durante el entrenamiento. Con esto, el entrenamiento en el mundo real es poco práctico para la mayoría de los casos y generalmente se usan simuladores para entrenar los agentes. Si bien el uso de simuladores permite acelerar el entrenamiento, estas experiencias adquiridas en el simulador son diferentes a las experiencias adquiridas en la realidad, generando así el problema del *reality-gap*. Esta brecha puede ocurrir cuando la dinámica de las interacciones simuladas entre el ambiente y el agente difiere de las interacciones reales, o cuando existen diferencias entre las observaciones y acciones entre el mundo real y el ambiente simulado.

Si bien el problema de cargar autónomamente utilizando agentes RL ha sido abordado en el pasado (e.g. [24, 25, 26]), estos trabajos o están enfocados en otra maquinaría como los cargadores frontales o retroexcavadoras, o solo se hacen pruebas en ambientes simulados. En este contexto, este trabajo busca aportar al problema de carguío autónomo utilizando RL, mediante la creación de un agente capaz de controlar un LHD para excavar material desde un punto de extracción. La simulación de la interacción entre maquinaria y material es especialmente difícil debido a las distintas dinámicas involucradas, por lo que evaluar distintos métodos de simulación de interacciones es importante para la literatura. Además, el agente propuesto es evaluado extensivamente en el

mundo real para determinar si es que el *reality-gap* fue superado exitosamente y evaluar cómo se desempeña comparado con excavación teleoperada y sistemas de control tradicionales.

1.2. Hipótesis

Este trabajo aborda la siguiente hipótesis:

Utilizando aprendizaje reforzado profundo, se puede entrenar una política robusta capaz de excavar material con un LHD desde un punto de extracción, tanto en simulación como en la realidad, con un desempeño similar a la excavación mediante teleoperación o a la excavación mediante sistemas autónomos diseñados por expertos.

1.3. Alcance

El problema de excavación de material ha sido reducido a extraer material desde un punto de extracción ubicado frente a la máquina. Como se mencionó anteriormente, todo el proceso de carguío de material incluye otra fases previas como el navegar hasta el punto de extracción y posicionarse frente a la pila, y fases posteriores como depositar el material en un punto objetivo. Todas estas fases son omitidas en este trabajo puesto que solo se aborda la fase de excavación.

Para realizar las pruebas experimentales en el mundo real, se utilizan un LHD y un punto de extracción, ambos a escala y diseñados para funcionar en conjunto. Tanto el LHD como el punto de extracción son simulados para entrenar la política con aprendizaje reforzado.

1.4. Objetivos

1.4.1. Objetivo general

El objetivo general de este trabajo es resolver el problema de excavación autónoma con un LHD desde puntos de extracción mediante el uso de aprendizaje reforzado. Esto consiste en el diseño, implementación y validación de las soluciones tanto en un entorno simulado como real. Debido a que las soluciones deben ser desplegadas en el mundo real, los supuestos considerados para entrenar el agente en simulación, en particular aquellos referentes a los sensores y capacidad de cómputo, deberán apegarse a la realidad.

1.4.2. Objetivos específicos

Se establecen los siguientes objetivos específicos para cumplir con el objetivo general anterior y verificar la hipótesis de este trabajo:

- Efectuar una revisión de la literatura sobre aprendizaje reforzado profundo y excavación autónoma de material enfocado en LHDs.
- Formular e implementar un sistema de excavación autónoma de material basado en aprendizaje reforzado profundo.

- Validar experimentalmente que los sistemas desarrollados puedan excavar material tanto en simulación como en el mundo real y comparar su desempeño con otros sistemas disponibles.

1.5. Estructura de la tesis

Este trabajo se estructura de la siguiente forma:

- En el Capítulo 3 se presentan los antecedentes relacionados con este campo de estudio. Se introducen distintas metodologías de control, tanto clásicas como basadas en aprendizaje de máquinas para el control de LHDs. Además, se presentan algunas metodologías de simulación para la interacción entre la máquina y el material.
- En el Capítulo 2 se presentan los distintos antecedentes generales necesarios para abordar los contenidos de este trabajo, particularmente las máquinas LHD y su funcionamiento, los procesos de decisión de Markov, el aprendizaje reforzado profundo y el modelo utilizado para la simulación de material.
- En el Capítulo 4 se presenta el diseño e implementación de un sistema de excavación autónoma para máquinas LHD basado en aprendizaje reforzado profundo.
- En el Capítulo 5 se presentan los resultados obtenidos, tanto en simulación como en la realidad, en los carguíos ejecutados con el sistema propuesto en este trabajo.
- En el Capítulo 6 se revisan y analizan los distintos resultados obtenidos.
- En el Capítulo 7 se presentan las conclusiones derivadas del trabajo realizado y se entregan posibles direcciones para trabajos futuros.

Capítulo 2

Marco teórico

La interacción entre la máquina y la pila de material durante la excavación es modelada como un proceso de decisión de Markov. Esta formulación permite diseñar una solución basada en aprendizaje reforzado para entrenar la política que realiza de forma autónoma la excavación. En esta sección, se introduce en primer lugar los procesos de decisión de Markov, para luego describir su implementación en una solución basada en aprendizaje reforzado profundo.

2.1. Procesos de decisión de Markov

Los procesos de decisión de Markov (*Markov Decision Processes* (MDP) [27]) son procesos estocásticos que tienen la propiedad de Markov, descrita por la Ecuación (2.1). Un proceso estocástico con esta propiedad implica que es un proceso sin memoria, es decir, cada estado solo depende del estado anterior. Los MDP son una extensión de las cadenas de Markov. Tanto el espacio de estados, como el tiempo pueden ser descritos de forma continua o discreta dependiendo del problema abordado.

$$\mathbb{P}(X = x_{n+1}|X = x_n, X = x_{n-1}, ..., X = x_0) = \mathbb{P}(X = x_{n+1}|X = x_n) \tag{2.1}$$

Un MDP se define por la tupla $(\mathcal{S}, \mathcal{A}, \mathcal{T}, r)$, en donde cada componente corresponde a:

- $\mathcal{S}$ es el conjunto de estados que puede tener el proceso.
- $\mathcal{A}$ es el conjunto de acciones que se puede en el proceso.
- $\mathcal{T}(s, a, s')$ es la función de transición de estados.
- $r(s, a)$ es la función de recompensa.

Como se detalla en [28], los MDPs permiten formular matemáticamente el problema de aprender a partir de interacciones para lograr un objetivo. Aquello que toma las decisiones y aprende una política se define como el agente. Aquello con lo que el agente interactúa se define como el ambiente. Considerando un MDP, el ciclo de interacción entre el agente y el ambiente se desarrolla con el agente observando un estado s_t en el cual se ejecuta la acción a_t, lo que lleva al agente a un siguiente estado s_{t+1} siguiendo la función de transición de estados $\mathcal{T}(s, a, s')$ y recibe la recompensa r_t entregada por la función de recompensa $r(s, a)$. La Figura 2.1 (adaptada de [28]) muestra la dinámica de la interacción entre el agente y el ambiente.

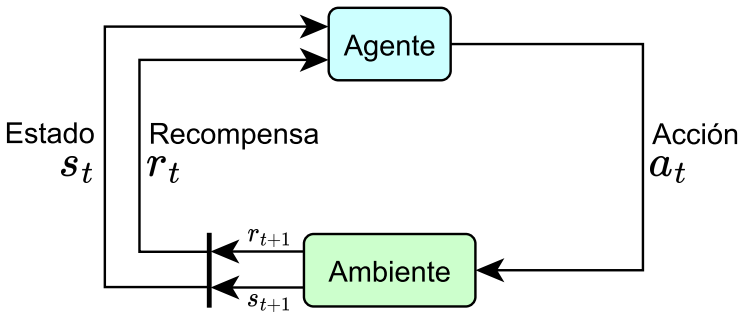

Figura 2.1: Interacción Agente-Ambiente en un MDP (adaptado de [28]).

La función de transición de estados determina la dinámica de la interacción entre el agente y el ambiente. Esta función define la probabilidad de pasar a un estado s' desde un estado s dado que se toma la acción a. La Ecuación (2.2) define la función de transición de estados para un MDP.

$$\mathcal{T}(s, a, s') = p(s_{t+1} = s' | s_t = s, a_t = a) \tag{2.2}$$

La función de recompensa determina la recompensa esperada para cada transición de estados. Esta función entrega un valor según el estado actual del ambiente s_t y la acción ejecutada por el agente a_t. La Ecuación (2.3) define la función de recompensa para un MDP.

$$r(s, a) = \mathbb{E}[r_t | s_t = s, s_t = a] \tag{2.3}$$

El agente no siempre puede observar la totalidad del estado del ambiente, por lo que observa estados incompletos o con información ruidosa. Esto transforma el proceso a un MDP parcialmente observable (*Partially Observable Markov Decision Problem* (POMDP)). Un POMDP se define por la tupla $(\mathcal{S}, \mathcal{A}, \mathcal{T}, r, \Omega, \mathcal{O})$, en donde cada componente corresponde a:

- $(\mathcal{S}, \mathcal{A}, \mathcal{T}, r)$ es el MDP.
- Ω es el conjunto de observaciones que se obtienen del proceso.
- $\mathcal{O}(s', a, o)$ es la función de observación.

La función de observación determina la probabilidad de obtener una observación o dado que el agente tomó una acción a y llegó al estado s'. La Ecuación (2.4) define la función de observación para un MDP.

$$\mathcal{O}(s', a, o) = p(o_{t+1} = o | s_{t+1} = s', a_t = a) \tag{2.4}$$

2.2. Aprendizaje reforzado

El aprendizaje reforzado es una de las tres principales líneas de desarrollo del aprendizaje de máquinas, los otros dos siendo el aprendizaje supervisado y el aprendizaje no supervisado. El aprendizaje reforzado es una metodología para resolver problemas de toma de decisiones secuenciales, los cuales son modelados generalmente mediante un MDP. De esta forma, el agente final va a interactuar con el ambiente ejecutando una acción a_t determinada por una política $\pi(a_t|s_t)$ (abreviada

como π) y percibir una recompensa r_t. El objetivo principal del aprendizaje reforzado es aprender la política π que maximice el valor acumulado de las recompensas obtenidas durante varias interacciones.

Por un lado, cuando la interacción agente-ambiente tiene estados terminales bien definidos, es decir, existe un estado final en que la tarea se da por finalizada, la interacción se caracteriza como "episódica". Debido a la existencia de un estado final para la tarea, la interacción agente-ambiente es acotada temporalmente por un tiempo $T < \infty$, por lo que el retorno puede ser definido como la suma de las recompensas obtenidas desde un instante de tiempo t, definido también por la Ecuación (2.5).

$$G_t \doteq r_t + r_{t+1} + r_{t+2} + ... + r_{t+T} \tag{2.5}$$

Por otro lado, cuando la interacción agente-ambiente no tiene estados terminales bien definidos, es decir, el episodio terminal ocurre en un tiempo $T = \infty$, la Ecuación (2.5) podría divergir. Para solucionar esto, se agrega un factor de descuento $\gamma \in [0, 1]$ a las recompensas obtenidas en cada instante de tiempo para definir un retorno descontado. Este retorno descontado se define como la suma de las recompensas descontadas a partir de cierto instante de tiempo t, definido también por la Ecuación (2.6).

$$G_t \doteq \sum_{k=0}^{\infty} \gamma^k r_{t+k} = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + ... \tag{2.6}$$

El factor de descuento γ permite controlar el horizonte sobre el cuál se evalúa el retorno en cada iteración. Cuando se utiliza un factor de descuento γ tal que $\gamma = 1$ se tiene el mismo caso que la Ecuación (2.5) y se considera que se le da una importancia equivalente a todas las recompensas obtenidas durante la interacción. Cuando se utiliza un factor de descuento γ tal que $0 < \gamma < 1$, según el valor utilizado de γ, se puede controlar la importancia de las recompensas obtenidas. Es decir, si el valor de γ es cercano a 0, se fomentará el aprendizaje de acciones que generen mayores recompensas en el corto plazo, mientras que si el valor de γ es cercano a 1, se fomentará el aprendizaje de acciones que generen mayores recompensas en el largo plazo.

La interacción completa entre el agente y el ambiente se define como trayectoria τ, esto corresponde a todos los pares de estados percibidos por el agente y todas las acciones ejecutadas por el agente. Debido a que la interacción está modelada como un MDP, el cual tiene una función de transición de estados que entrega la probabilidad de pasar de un estado a otro al ejecutar cierta acción, la probabilidad de que el agente recorra una trayectoria que inicia en un estado s_1 (determinado por una función de probabilidades p_f) y que termina en un instante de tiempo T queda definida por la Ecuación (2.7).

$$p_\pi(s_1, a_1, s_2, a_2, ..., s_T, a_T) = p_f(s_1) \prod_{t=1}^{T} \pi(a_t|s_t) p(s_{t+1}|s_t, a_t) \tag{2.7}$$

Finalmente, y como descrito anteriormente, la Ecuación (2.8) define el objetivo principal del aprendizaje reforzado, aprender una política π que maximice el retorno esperado obtenido a lo largo de las trayectorias que recorre en su interacción con el ambiente, determinadas por $p_\pi(\tau)$.

$$J_{RL}(\pi) = \mathbb{E}_{\tau \sim p_\pi(\tau)} \left[\sum_{t=1}^{T} \gamma^{t-1} r(s_t, a_t) \right] \tag{2.8}$$

2.2.1. Funciones de valor

Existen dos funciones de valor que permiten evaluar cuán beneficiosa son las trayectorias mediante el retorno que recibe el agente. La primera corresponde a la función de valor $V^{\pi}(s)$ y la segunda corresponde a la función de valor del par estado-acción $Q^{\pi}(s,a)$.

La función de valor $V^{\pi}(s)$ entrega el valor esperado del retorno al recorrer trayectorias determinadas por la función $p_{\pi}(\tau)$, siguiendo la política π y partiendo desde un estado s. Esto permite evaluar cuán bueno es para un agente estar en el estado s ya que va a tener que actuar según la política π. La Ecuación (2.9) define la función de valor $V^{\pi}(s)$.

$$V^{\pi}(s) \doteq \mathbb{E}_{\tau_t \sim p_{\pi}(\tau_t)} \left[\sum_{k=0}^{T} \gamma^k r(s_{t+k}, a_{t+k}) \middle| s_t = s \right] \tag{2.9}$$

La función de valor del par estado-acción $Q^{\pi}(s,a)$ entrega el valor esperado del retorno al recorrer trayectorias determinadas por la función $p_{\pi}(\tau)$, siguiendo la política π, partiendo desde un estado s y ejecutando como primera acción a. Esto permite evaluar cuán bueno es para un agente estar en el estado s y ejecutar la acción a. La Ecuación (2.10) define la función del par estado-acción $Q^{\pi}(s,a)$.

$$Q^{\pi}(s,a) \doteq \mathbb{E}_{\tau_t \sim p_{\pi}(\tau_t)} \left[\sum_{k=0}^{T} \gamma^k r(s_{t+k}, a_{t+k}) \middle| s_t = s, a_t = a \right] \tag{2.10}$$

En el caso de la función del valor par estado-acción, todas las acciones salvo la primera son determinadas por la política π. De esta forma, una vez realizada la primera acción, el resto de valor se determina con la función de valor presentada en la Ecuación (2.9), por lo que se puede expresar la función $Q^{\pi}(s,a)$ en función de $V^{\pi}(s)$, como se muestra en la Ecuación (2.11).

$$Q^{\pi}(s,a) = r(s,a) + \mathbb{E}_{s' \sim p(s'|s,a)} V^{\pi}(s') \tag{2.11}$$

De forma similar, si se considera que la función de valor $V^{\pi}(s)$ es equivalente a evaluar la función de valor del par estado-acción para todas las acciones que ejecutaría la política, se puede escribir la función $V^{\pi}(s)$ en función de $Q^{\pi}(s,a)$, como se muestra en la Ecuación (2.12).

$$V\pi(s) = \mathbb{E}_{a \sim \pi(a|s)} Q^{\pi}(s,a) \tag{2.12}$$

Las Ecuaciones (2.11) y (2.12) permiten escribir ambas funciones de valor mediante relaciones de recurrencia, generando así las ecuaciones (2.13) y (2.14). Estas nuevas ecuaciones son denominadas ecuaciones de Bellman.

$$V^{\pi}(s) = \mathbb{E}_{a \sim \pi(a|s)} \left[r(s,a) + \gamma \mathbb{E}_{s' \sim p(s'|s,a)} V^{\pi}(s') \right] \tag{2.13}$$

$$Q^{\pi}(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim p(s'|s,a)} \mathbb{E}_{a' \sim \pi(a'|s')} Q^{\pi}(s',a') \tag{2.14}$$

Para saber si es que una política π es mejor que una política π' se evalúa qué política entregue un mejor retorno esperado, es decir, si $V^{\pi}(s) \geq V^{\pi'}(s)$ entonces se tiene que la política π es mejor

que la política π'. Una política óptima π^* a un problema de aprendizaje reforzado es aquella que maximiza la Ecuación (2.8) presentada anteriormente, la cual también maximiza las funciones de valor $V^\pi(s)$ y $Q^\pi(s,a)$. Las funciones de valor óptimas están definidas en las Ecuaciones (2.15) y (2.16).

$$V^*(s) = \max_{a \in A} \left[r(s,a) + \gamma \mathbb{E}_{s' \sim p(s'|s,a)} V^*(s') \right] \tag{2.15}$$

$$Q^*(s,a) = r(s,a) + \gamma \max_{a' \in A} \mathbb{E}_{s' \sim p(s'|s,a)} Q^*(s',a') \tag{2.16}$$

2.2.2. Algoritmos de aprendizaje reforzado

Existen múltiples algoritmos de aprendizaje reforzado y cada uno con múltiples configuraciones, por lo que su selección es parte importante al momento de resolver el problema. En cierto casos, hay algoritmos que son incompatibles con ciertas tareas debido a los espacios de estados o acciones.

Si bien no existe una taxonomía definida para clasificar los distintos algoritmos de aprendizaje reforzado, existen ciertas características generales que permiten clasificar los distintos algoritmos. La característica principal para clasificar algoritmos de aprendizaje reforzado corresponde a si es que son *Model-Based* o *Model-Free*.

Un algoritmo que tiene acceso al modelo del ambiente o que lo puede aprender se denomina *Model-Based*. El modelo del ambiente es una función que predice el estado siguiente del agente y podría predecir la recompensa recibida al cambiar de estado. Cuando el agente tiene acceso al modelo del ambiente, mediante la predicción de estados futuros y recompensas puede estimar cuál va a ser la mejor acción a realizar y así llegar a una política óptima para el ambiente. De esta forma, el agente al momento de realizar una acción, con el modelo del ambiente puede determinar cuál va a ser la acción óptima antes de ejecutarla. El principal problema es que no siempre se tiene acceso a un modelo perfecto del ambiente, por lo este debe ser aprendido a partir de múltiples interacciones agente-ambiente. El proceso de aprender el ambiente requiere de una gran cantidad de interacciones para obtener un modelo cercano al real y así llegar a una política capaz de desempeñarse correctamente en el ambiente real. Además, durante el proceso de entrenamiento, se puede introducir un sesgo al modelo del ambiente que luego imposibilita al agente de funcionar en el ambiente real debido a que aprendió a resolver la tarea en un ambiente distinto. La familia de algoritmos *Model-Based* no tiene categorías fáciles a definir que permitan clasificar los algoritmos, por lo que se entregan algunos ejemplos de algoritmos [29, 30, 31, 32, 33, 34].

Un algoritmo que no usa o aprende un modelo del ambiente se denomina *Model-Free*. Debido a que no tienen conocimiento sobre el ambiente, generalmente los algoritmos *Model-Free* requieren una mayor cantidad de interacciones para llegar a una política capaz de resolver el problema. La familia de algoritmos *Model-Free* tiene tres subcategorías. La primera subcategoría corresponde a algoritmos que buscan los parámetros θ óptimos para la política π a través de la optimización de $J_{RL}(\pi_\theta)$. En estos casos, se debe asumir que la política es diferenciable con respecto a sus parámetros para poder actualizarlos mediante el gradiente ascendente determinado por $\nabla J_{RL}(\pi_\theta)$. Los algoritmos presentados en [35, 36] son ejemplos de esta familia de algoritmos. La segunda subcategoría de algoritmos corresponde a algoritmos que buscan parámetros θ para las funciones de valor $V^\pi_\theta(s)$ y $Q^\pi_\theta(s,a)$, también llamadas *V-function* y *Q-function* respectivamente, utilizando las ecuaciones (2.15) y (2.16). Al obtener la función de valor óptima, el agente ejecuta las acciones que maximicen el retorno de la función de valor. Los algoritmos presentados en [37, 38] son ejemplos de

esta familia de algoritmos. Existen algoritmos que utilizan ambas estrategias, entrenar una política y las funciones de valor, los cuales son denominados *Actor-Crítico*, como por ejemplo [39, 40].

Adicionalmente, los algoritmos de aprendizaje reforzado pueden ser distinguidos como algoritmos *on-policy* u algoritmos *off-policy*. Los algoritmos *on-policy*, como por ejemplo el algoritmo *policy iteration*, buscan mejorar la política actual la cual también es utilizada para determinar las acciones a ejecutar, como por ejemplo el algoritmo *policy iteration*. Mientras que los algoritmos *off-policy*, como por ejemplo el algoritmo *q-learning*, utilizan una política para estimar las funciones de valor y otra política para seleccionar qué acciones ejecutar, las cuales se van a actualizando a distintas frecuencias.

Finalmente, la forma de entrenar los algoritmos de aprendizaje reforzado también puede ser clasificada en tres categorías: *online* RL, *offline* RL y una combinación de ambas formas. La categoría *online* RL corresponde a entrenar los agentes interactuando directamente con el ambiente, como se muestra en la Figura 2.1. Mientras que el *offline* RL corresponde a entrenar los agentes utilizando una base de datos de distintas interacciones con el ambiente obtenidas previamente. La última categoría corresponde a una mezcla de las dos formas de entrenamiento, en donde se utiliza inicialmente una base de datos de interacciones para entrenar al agente y luego se refina la política aprendida interactuando con el ambiente. Tanto el *online* RL y *offline* RL han sido exitosamente utilizados en distintos ámbitos (e.g. [39, 40, 41, 42, 43]).

2.2.3. Aprendizaje reforzado profundo

El aprendizaje reforzado profundo (*Deep Reinforcement Learning*, DRL) es la combinación del aprendizaje profundo (*Deep Learning*, DL) y el aprendizaje reforzado. La introducción de elementos del mundo del aprendizaje profundo ha permitido al aprendizaje reforzado progresar significativamente en el último tiempo logrando así resolver tareas complejas, tanto en simulación como en el mundo real, tales como tareas de navegación, manipulación u otras tareas relacionadas a áreas externas a la robótica como el procesamiento de lenguaje (e.g. [44, 45, 46]). En la siguiente sección, se introduce el DRL junto con algunas de sus características y el algoritmo DDPG.

El aprendizaje reforzado clásico (sin uno de apoximadores funcionales o tabular) asocia una acción óptima a cada estado para resolver el problema, mediante la exploración de estados y el análisis del retorno para cada uno de ellos. No obstante, el desempeño de estos algoritmos disminuye con el crecimiento del espacio de acciones y estados, llegando a ser inviable para ciertos casos cuando la cardinalidad es muy alta. Esto ocurre debido a que se vuelve imposible la exploración de todos los estados y la cantidad de información (pares estado-acción óptima) a almacenar es demasiada.

Para superar este problema, los algoritmos de DRL utilizan redes neuronales como aproximadores funcionales de distintos componentes del algoritmo y así poder resolver problemas con alta cardinalidad de espacios de estados o acciones. Distintas redes neuronales pueden ser utilizadas, siendo las más comunes las redes neuronales *feed-forward* y variaciones de esta tales como las redes neuronales convolucionales y las redes neuronales recurrentes. Estas redes permiten mapear una entrada x a una salida y mediante la función f tal que $y = f(x, \theta)$, en donde θ son los parámetros de la red. Las redes pueden ser utilizadas para aproximar distintas partes del algoritmo DRL, como el modelo del ambiente, la política o las funciones de valor entre otras.

La selección de la red neuronal para entrenar un agente DRL depende directamente de las ca-

racterísticas del problema a solucionar. Las redes neuronales *feed-forward* son la formulación más básica de las redes neuronales como aproximadores funcionales y pueden ser utilizadas en cualquier problema de DRL. Las redes neuronales convolucionales son útiles para casos en que las entradas contienen imágenes, debido a que se aprovechan de información intrínseca de las imágenes como la relación entre píxeles contiguos. Las redes neuronales recurrentes son útiles para casos en que las entradas tienen alguna relación temporal entre datos, como las series de tiempo.

Deep Deterministic Policy Gradient (DDPG)

Deep Deterministic Policy Gradient es un algoritmo propuesto por Lillicrap et al. [39]. Es un algoritmo del tipo actor-crítico que busca aprender tanto una *Q-function* como una política. Su diseño está fuertemente vinculado al algoritmo de Q-Learning propuesto en [47]. A continuación se detallan algunas de las características claves del algoritmo DDPG y su funcionamiento para resolver tareas de aprendizaje reforzado.

El algoritmo DDPG aprende simultáneamente una política óptima y una *Q-function*. La principal motivación de combinar ambos aprendizajes es que si se conoce la *Q-function* óptima $Q^*(s, a)$, entonces en cualquier estado se puede encontrar la acción óptima $a^*(s)$ que maximiza la función de valor resolviendo $a^*(s) = \arg\max_{a \in A} Q^*(s, a)$. El aproximador de la *Q-function* $Q_\theta(s, a)$ es el crítico del algoritmo mientras que el aproximador de la política $\mu_\phi(s)$ (determinista) es el actor del algoritmo.

El algoritmo DDPG solo puede ser utilizado para ambientes con espacios de acciones continuos debido al método utilizado para calcular $\max_{a \in A} Q_\theta(s, a)$. Cuando el espacio de acciones es continuo, se asume que la función $Q^*(s, a)$ es diferenciable con respecto a la acción a. Luego, los parámetros de $\mu_\phi(s)$ pueden ser aprendidos mediante métodos de búsqueda basados en gradientes y se utiliza $\mu_\phi(s)$ para aproximar $\max_{a \in A} Q_\theta(s, a)$ como $\max_{a \in A} Q_\theta(s, \mu_\phi(s))$. Se utiliza $\mu_\phi(s)$ para calcular la función de valor puesto que tiene un menor costo computacional a ejecutar una optimización de la función en cada paso de tiempo.

La actualización de los parámetros de $Q_\theta(s, a)$ utiliza la Ecuación de Bellman óptima (2.16) para definir la función de pérdida *Mean-Squared Bellman Error* (MSBE) definida por la Ecuación (2.17) que busca que se satisfaga la ecuación de Bellman (2.14).

$$L(\theta) = \mathbb{E}_{(s,a) \sim p_\pi(s,a)} \left[\Big(Q_\theta(s, a) - \big(r + \gamma Q_\theta(s', a') \big) \Big)^2 \right] \tag{2.17}$$

Hay dos componentes importantes para el cálculo de (2.17) en el caso de DDPG, el *replay buffer* y las *target networks*. El *replay buffer* es un arreglo circular finito D el cual almacena las experiencias del agente que luego son utilizadas para entrenar las redes. Es un arreglo circular puesto que al alcanzar el número máximo de experiencias, las más antiguas se reemplazan por nuevas experiencias. Debido a esto, el tamaño del *replay buffer* es importante: si es muy grande, almacena experiencias obsoletas que ralentizan el aprendizaje de una política óptima; si es muy pequeño, existe la posibilidad de un sobreajuste a las nuevas experiencias lo que destruye el aprendizaje (e.g. *catastrophic forgetting* [48]).

La Ecuación (2.17) permite calcular el error con respecto al objetivo $y = r + \gamma Q_\theta(s', a')$. El problema principal de este objetivo es que también depende de los parámetros que se buscan

aprender, lo que genera una inestabilidad durante el entrenamiento. Para resolver esto, se crea una copia $Q_{\theta_{targ}}(s,a)$ de $Q_\theta(s,a)$ denominada *target network*. Ambos conjuntos de parámetros θ_{targ} y θ son actualizados aplicando un promedio móvil exponencial para copiar los parámetros de una red a la otra. Además de la *target network* de la función de valor, se crea una copia $\mu_{\phi_{targ}}$ de μ_ϕ que genera una *target network* para la política. El promedio móvil exponencial aplicado está definido por la Ecuación (2.18), en donde el parámetro $\rho \in [0,1]$ es el factor de suavizado, ψ y ψ_{targ} representan los parámetros de una función y su copia objetivo respectivamente.

$$\psi_{targ} \leftarrow \rho\psi_{targ} + (1-\rho)\psi \tag{2.18}$$

Este algoritmo es considerado *off-policy* debido a que el *replay buffer* contiene experiencias antiguas que son obtenidas con una política anterior. Estas experiencias antiguas se pueden utilizar debido a que por definición, la *Q-function* óptima satisface la ecuación de Bellman óptima para todas las transiciones posibles. Además, la exploración de distintos estados se promueve mediante el uso de ruido de exploración $\mathcal{N}$ (es un proceso aleatorio) el cual se agrega a la política.

Específicamente para DDPG, al aplicar ambas estrategias de *replay buffers* y *target networks*, la función de pérdida utilizada para encontrar los parámetros óptimos de la *Q-function* se define por la Ecuación (2.19), en donde la tupla (s,a,r,s') corresponde a experiencias extraídas desde el *replay buffer*.

$$L(\theta) = \mathbb{E}_{(s,a,r,s')\sim D}\left[\left(Q_\theta(s,a) - \left(r + \gamma Q_{\theta_{targ}}\left(s', \mu_{\phi_{targ}}(s')\right)\right)\right)^2\right] \tag{2.19}$$

La actualización de los parámetros de $\mu_\phi(s)$ se basa en maximizar el retorno esperado para la función de valor $Q_\theta(s,a)$. Se utiliza el gradiente del retorno esperado con respecto a los parámetros de la política para encontrar los parámetros óptimos. Esto queda definido por la Ecuación (2.20).

$$\begin{aligned}\nabla_\phi J(\phi) &\approx \mathbb{E}_{(s,a,r,s')\sim D}[\nabla_\phi Q_\theta(s, \mu_\phi(s))] \\ &= \mathbb{E}_{(s,a,r,s')\sim D}[\nabla_\phi Q_\theta(s,a)|_{a=\mu_\phi(s)} \nabla_\phi \mu_\phi(s)]\end{aligned} \tag{2.20}$$

El resumen del algoritmo de DDPG se muestra a continuación en el Algoritmo 1.

En este capítulo se introdujeron los procesos de decisión de Markov junto con el aprendizaje reforzado y su rama de aprendizaje reforzado profundo. En este trabajo, se utiliza el algoritmo DDPG para entrenar un agente de forma *online*.

Algoritmo 1: Deep Deterministic Policy Gradient

Inicializar $Q_\theta(s,a)$ y $\mu_\phi(s)$ con pesos θ y ϕ
Inicializar $Q_{\theta_{targ}}(s,a)$ y $\mu_{\phi_{targ}}(s)$ con pesos θ_{targ} y ϕ_{targ}
Copiar parámetros de redes principales a redes objetivos $\theta_{targ} \leftarrow \theta$ y $\phi_{targ} \leftarrow \phi$
Inicializar *replay buffer* D
while *episodio*$= 1, M$ **do**
    Inicializar proceso aleatorio $\mathcal{N}$ para ruido de exploración
    Obtener primer estado s_1
    while *t*$= 1, T$ **do**
        Elegir $a_t = \mu_\phi(s_t) + \mathcal{N}_t$
        Ejecutar acción a_t, observar r_t y s_{t+1}
        Guardar experiencia (s_t, a_t, r_t, s_{t+1}) en D
        Muestrear un *minibatch* de N transiciones $(s_j, a_j, r_j, s_{j+1}) \in D$
        Calcular $y_j = \begin{cases} r_j & \text{si } s_{j+1} \text{ es estado terminal} \\ r_j + \gamma Q_{\theta_{targ}}(s_{j+1}, \mu_{\phi_{targ}}(s_{j+1})) & \text{si } s_{j+1} \text{ no es estado terminal} \end{cases}$
        Actualizar $Q_\theta(s,a)$ minimizando el costo $L(\theta) = \frac{1}{N}\sum_{j=1}^{N}(Q_\theta(s,a) - y_j)^2$
        Actualizar $\mu_\phi(s)$ maximizando $J(\phi) = \frac{1}{N}\sum_{j=1}^{N} Q_\theta(s_j, \mu_\phi(s_j))$
        Actualizar *target networks*:
            $\theta_{targ} \leftarrow \rho\theta + (1-\rho)\theta_{targ}$
            $\phi_{targ} \leftarrow \rho\phi + (1-\rho)\phi_{targ}$
    end
end

Capítulo 3

Estado del arte

La automatización de máquinas LHD ha sido abordada en la literatura mediante múltiples métodos de control, tanto clásicos como con incorporación de inteligencia artificial y redes neuronales. En la siguiente sección, se consideran también aquellos estudios que trabajan con cargadores frontales (también referidos como *wheel loaders* o WL) o cargadores *skid steer* (también referidos como *skeed steer loaders* o SSL) debido a que la forma de excavación es similar a la de un LHD. En primer lugar, se analiza la literatura referente a técnicas clásicas de control, sin la incorporación de inteligencia artificial. Luego, se analiza la literatura referente a metodologías de carguío que utilicen inteligencia artificial y finalmente se analiza la literatura sobre metodologías de simulación de la interacción entre máquina y material.

Como se indica en [49], la operación de maquinaria diseñada para carguío de material tiene dos condiciones principales sobre su ciclo de funcionamiento: deben operar de forma segura para todos los trabajadores involucrados y deben tener un rendimiento adecuado con el fin de que su funcionamiento sea un aporte. La condición de seguridad es más importante que aquella de rendimiento. Para la condición de seguridad hay dos factores principales. El primer factor es el resbalamiento de las ruedas durante el carguío, debido a que genera la pérdida de control de la máquina y además produce un desgaste acelerado del neumático, lo que aumenta los costos de mantención. El segundo factor es la detección de colisiones y evasión de obstáculos, para evitar que la máquina colisione contra muros u otra maquinaría trabajando en el mismo entorno. Para medir el rendimiento del proceso de excavación se tienen varias métricas tales como el factor de llenado del balde, la eficiencia de combustible y la duración del ciclo de operación. Por último, [49] indica que el modelamiento de la interacción entre el balde y el material son considerablemente complejas y que un modelo fidedigno aún no es alcanzado.

3.1. Automatización de la excavación de material

3.1.1. Metodologías clásicas de automatización

Una de las metodologías básicas para el carguío autónomo es el control de posición del balde dentro de la pila para seguir trayectorias diseñadas para cargar material. Estas trayectorias son diseñadas en base a conocimiento previo sobre las condiciones del punto de extracción y sobre es-

trategias adecuadas para excavar material desde dicho punto de extracción. Estas metodologías en general tienen como dificultad que se vuelve imposible asegurar el seguimiento preciso de la trayectoria durante el carguío, especialmente en casos de roca fragmentada debido a la alta variabilidad de la granulometría del material a cargar. Recientes trabajos en esta área (e.g. [50, 51, 52, 53, 54]) buscan optimizar las trayectorias de carguío mediante el uso de simuladores o datos reales, con el fin de mejorar aspectos como el consumo energético y la cantidad de material extraído.

Para superar el problema del seguimiento de trayectoria, una de las propuestas en la literatura es ajustar la trayectoria en tiempo real basado en la resistencia generada por la pila. Para el caso de los LHDs o cargadores frontales, el uso de controladores de admitancia es la elección más popular. Un controlador de admitancia se describe como un controlador que observa fuerza y entrega movimiento. En [55] se propone que un controlador de admitancia lograría superar la deficiencias que presenta el control de trayectorias puro. Además, entregan un ejemplo de cómo diseñar un controlador de admitancia, utilizando datos extraídos de carguíos realizados por conductores profesionales. En [56] se propone un controlador de admitancia, el cual es probado en un cargador frontal con capacidad de 1 tonelada y un LHD con capacidad de 14 toneladas. El controlador descrito tiene como entrada al sistema la fuerza ejercida por la pila de material sobre el balde y entrega comandos de velocidad para los distintos actuadores. En dicho trabajo, el proceso de excavación se divide en tres etapas: (i) *Entrada*, que corresponde a enterrar el balde dentro de la pila; (ii) *Excavación*, que corresponde a llenar el balde; (iii) *Retirada*, que corresponde a retroceder y retirar el balde de la pila. Se describe además un cuarto estado que corresponde a cuando la máquina está atorada y no puede seguir con el carguío. El controlador de admitancia es utilizado en el estado de excavación, el cual controla la velocidad del actuador de volteo observando las fuerzas que percibe el actuador de levante. En las dos máquinas en las que fue probado el controlador, se obtuvo un mejor desempeño comparado con un operador experto. El uso del controlador basado en admitancia redujo el tiempo de ejecución y aumentó la cantidad de material cargado, no obstante, aumentó el esfuerzo que hacía la máquina durante el carguío. Dicho trabajo también logra evidenciar que un controlador diseñado para excavar con LHDs puede ser usado para excavar con cargadores frontales y viceversa, con sus respectivas modificaciones para que funcione en la otra máquina. El trabajo de [57] es una extensión del trabajo propuesto por [56] en donde se prueban distintas combinaciones de parámetros para el controlador de admitancia y propone un algoritmo para encontrar automáticamente los parámetros adecuados para excavar. De forma similar, en [58] se propone otro método para encontrar parámetros adecuados para el controlador. Sin embargo, en todos los casos anteriores, se requieren múltiples ciclos de carguío para encontrar los parámetros óptimos. En [59] se propone encontrar los parámetros óptimos mediante el uso de simuladores para así evitar la necesidad de utilizar equipamiento en el mundo real y acelerar el proceso.

En [60], se proponen modelos de regresión lineal para predecir el desplazamiento de los cilindros hidráulicos de levante y volteo del balde durante un carguío. Para el entrenamiento del modelo, se usaron datos extraídos de múltiples carguíos realizados por un conductor profesional utilizando un cargador frontal modelo Volvo L110G, en dos puntos de extracción con distintos tipos de material, el primero compuesto principalmente de grava gruesa y el segundo de rocas fragmentadas. Los modelos obtenidos en dicho trabajo logran captar la dinámica del ciclo de excavación pero los autores indican que tienen varias falencias. Por un lado, las soluciones basadas solamente en datos extraídos de conductores expertos tienen el problema de ser sesgadas, debido a que la calidad del controlador depende de la cantidad de condiciones diferentes de donde se extrajeron los datos. Por otro lado, estas soluciones no incluyen variables que ajusten el control hacia una rutina de exca-

vación que tenga en cuenta requerimientos tales como la cantidad de material cargado, el tiempo de ejecución o la energía gastada durante la maniobra. Consecuentemente, en [60] se propone que una solución basada en RL podría ser adecuada para este tipo de problemas. Se propone que una recompensa adecuada para el agente tenga un término de valor positivo alto relacionado a la cantidad de material cargado y términos pequeños de valor negativo relacionados al resbalamiento, estancamiento de la máquina, gasto de energía y duración de la maniobra. Estos son aspectos que se tienen en consideración para el diseño de la recompensa del agente propuesto en esta tesis.

En [61] se propone una solución para el ciclo de carguío completo, con todas las etapas incluyendo la navegación y posicionamiento. Es una extensión del trabajo propuesto en [62]. El algoritmo de excavación está basado en las técnicas usadas por conductores profesionales. En el trabajo de [62], el algoritmo de excavación se comporta como un controlador de tracción y fue diseñado para puntos de extracción en minería del tipo *sublevel stoping*, los cuales son similares entre distintos carguíos. Tiene dos estados entre los que cambia durante el ciclo: un primer estado en el que acelera de forma fija y levanta el balde siguiendo una función rampa, y un segundo estado en el que revisa si hay resbalamiento de ruedas para decidir si accionar el levante y así aumentar la tracción. Luego, en [61] se extiende esta máquina de estados agregando otros estados para que el algoritmo pueda funcionar frnte a una mayor cantidad de pendientes de pila y específicamente para puntos de extracción de una mina del tipo *room and pillar*. Este tipo de algoritmos para cargar requieren de un proceso de ajuste de parámetros y de cooperación con conductores expertos para ajustar la solución a cada ambiente donde es desplegado, por lo que todo proceso de migración a otro ambiente implica repetir este proceso de ajuste. No obstante, debido a que este método no utiliza ningún tipo de simulación no puede tener errores asociados al *reality-gap*. Por último, como los parámetros son ajustados en el despliegue del sistema autónomo, el sistema de carguío ejecuta de forma eficaz el carguío en el punto de extracción en el que se realiza la excavación. Cabe destacar que las soluciones presentadas en [61, 62] resuelven la totalidad del problema de carguío autónomo, incluyendo las etapas de transporte y descarga, mientras que este trabajo solo se aborda al etapa de extracción de material.

Otra metodología que cabe destacar es la del control predictivo, la cual utiliza un modelo del sistema para optimizar la siguiente acción a tomar. Una ventaja que tiene el modelo predictivo por sobre el aprendizaje reforzado es que este puede ser ajustado al momento de desplegar la solución en el mundo real, puesto que los parámetros pueden ser ajustados en el momento. Sin embargo, al tener que ser ajustado, los parámetros deben ser configurados de forma manual. Esta línea de estudio es cercana al aprendizaje reforzado, específicamente el caso del *Adaptive Model Predictive Control* (AMPC) siendo casi es mismo concepto que el *Model Based Reinforcement Learning* (MBRL), en donde un modelo del sistema y sus parámetros son aprendidos junto con la política o parámetros del controlador. Con respecto al control de los LHD, el control predictivo a sido utilizado principalmente para el control de la dirección de la máquina [63, 64, 65, 66]. Este tipo de control podría ser una opción futura para crear un controlador del LHD para efectuar carguíos.

3.1.2. Metodologías de automatización basadas en aprendizaje de máquinas

Por un lado, se tienen múltiples trabajos en los cuales se entrenan distintos controladores a partir de demostraciones de expertos y así evitar tener que simular la interacción entre la maquinaria y el material. En el conjunto de desarrollos presentados en [67, 68, 69], se entrenan redes neuronales siguiendo la metodología de aprendizaje a partir de demostraciones para aprender a cargar mate-

rial utilizando un cargador frontal liviano. En [67] se propone una red neuronal de una sola capa oculta con 5 neuronas y una base de datos construida por conductores no profesionales que contiene información sobre los actuadores, las acciones y la presión hidráulica de la máquina. La red neuronal propuesta aprende a excavar material y supera a una excavación realizada por un método heurístico pero no a uno realizado por un operador. En [68] se propone utilizar un regresor *Random Forest* (RF) como controlador y agregar imágenes como información adicional para el controlador, mediante la incorporación de redes convolucionales para preprocesar las imágenes. Con la introducción de nueva información proveniente de la red neuronal convolucional y el cambio a un regresor RF, el nuevo controlador es más robusto frente a cambios en el ambiente y supera con gran margen al controlador propuesto en [67]. Luego, en [69] se propone un controlador basado en el controlador propuesto en [67] con la adición de módulos de atención para filtrar la entrada de la red y con la adición de información extra como la presión hidráulica medida en los pistones del brazo. El nuevo controlador logra mejores resultados que los controladores propuestos en [67] y [68], y el ciclo de carguío es similar a uno realizado por un humano. De forma similar a los trabajos anteriores, en [70] se proponen controladores basados en redes neuronales para controlar un cargador frontal diseñado para cargar 7 toneladas. Específicamente, los controladores propuesto utilizan *Time Delayed Neural Networks* (TDNN) entrenadas con demostraciones de operadores expertos. En dicho trabajo, se entrenaron dos controladores distintos, un modelo de regresión que entrega directamente comandos para los actuadores hidráulicos y un modelo clasificatorio que entrega niveles predefinidos de acción para los actuadores hidráulicos. El mejor controlador obtenido fue el modelo clasificatorio, el cual alcanza un valor similar en la cantidad de material cargado pero con un tiempo de ejecución mayor comparado con un operador experto.

Por otro lado en la literatura, distintos agentes RL también han sido ocupados para solucionar la excavación autónoma. En [71] se propone aplicar RL para ajustar el comportamiento del controlador entrenado con *imitation learning* (IL) en [70]. Se utilizó el algoritmo *Deterministic Policy Gradient* (DPG) para entrenar el agente pero con cambios debido a los costos de entrenar con maquinaria real: no se utilizó ruido de exploración y el agente fue entrenado *on-policy*. Mediante el uso de recompensas enfocadas en la productividad y la fuerza percibida por la máquina durante el carguío, se entrena una política que logra aumentar la cantidad de material cargado en una cantidad menor de muestras. La metodología utilizada es costosa puesto que al entrenar el modelo en el mundo real, se necesita acceso a la maquinaria utilizada y a un ambiente donde desarrollar los experimentos.

Otro enfoque que utiliza RL, y que también es utilizado en este trabajo, es el de simular el ambiente, tanto la máquina como el punto de extracción, y entrenar desde cero un controlador capaz de realizar la maniobra de excavación. En [24] se propone una metodología para entrenar dos agentes RL con el fin de realizar la maniobra de carguío y sacar la mayor cantidad de material desde un punto de extracción. Ambos agentes fueron entrenados con el algoritmo de *Soft Actor Critic* (SAC). Un agente decide una vez por carguío cuál es la mejor ubicación de donde sacar material y el segundo agente ejecuta la excavación en la ubicación escogida. El agente de excavación tiene recompensas diseñadas para disminuir la cantidad de resbalamiento de las ruedas y la energía utilizada, y promover la extracción de material en la posición seleccionada. La inclusión de un agente de selección de posición permite evitar casos en los que el agente de excavación no puede realizar el carguío. Dicho trabajo utiliza información visual mediante cámaras montadas en distintas partes de la máquina además de la información intrínseca de la máquina (e.g. presión hidráulica, velocidad de las ruedas). Esta combinación de agentes solo fue evaluada en simulación por lo que no se

puede concluir si la información visual de cámaras mejora el desempeño. Durante los carguíos en el mundo real, las cámaras montadas en la máquina tienden a vibrar y el ambiente tiende a llenarse de polvo empeorando la visibilidad de las cámaras. Además, existe una diferencia entre las imágenes obtenidas en el mundo real y las imágenes entregadas por el simulador, por lo que es necesario probar estos agentes en la realidad y así comprobar su correcto funcionamiento. En [72] se propone un agente RL para ejecutar carguío con un SSL. El agente es entrenado únicamente en simulación utilizando el algoritmo DDPG y luego probado en el mundo real. La máquina simulada es idéntica a la máquina real con el fin de evaluar el *reality-gap* del método de simulación utilizado. Con respecto a las observaciones, además de la información intrínseca de la máquina, se agrega como observación el peso del material en el balde, calculado a partir de la fuerza ejercida sobre el balde por el material. Para entrenar el agente, en [72] los autores proponen dos recompensas: una función de recompensa que incentiva al agente a enterrar el balde en la pila de material y luego a retirarlo, y otra función de recompensa que incentiva a cargar material. El agente entrenado logra aprender a excavar material y es capaz de excavar material en ambientes nunca antes vistos similares al de entrenamiento.

Los trabajos presentados en [24] y en [72] son aquellos más cercanos con el desarrollo realizado en esta tesis, debido a que se entrena un agente RL sin utilizar experiencias previas de conductores expertos. Para la definición de la función de recompensa de este trabajo, se toman en cuenta varios aspectos de los distintos trabajos presentados anteriormente.

La Tabla 3.1 clasifica las distintas metodologías revisadas en esta sección, separando por la metodología utilizada para diseñar del controlador y si es que el controlado es validado en el mundo real o únicamente en simulación.

Tabla 3.1: Clasificación de metodologías de carguío.

Método	**Validación en simulación**	**Validación en realidad**
Analítico	[73][50][56]	[52][53][58][59]
Aprendizaje supervisado	[57]	[49][51][54][55][60][62][67][70]
Aprendizaje reforzado	[21][24]	[68][69][72][71]

3.2. Simulación de puntos de extracción de material

Como se mencionó anteriormente, la interacción entre la máquina y el punto de extracción es difícil de simular debido a las múltiples interacciones involucradas, por lo que múltiples metodologías que simulan esta interacción han sido estudiadas.

La simulación mediante *Discrete Element Modeling* (DEM) permite simular cada partícula de la pila y calcular cada interacción. Este método permite un alto grado de fidelidad pero requiere un alto costo computacional debido a que debe calcular las interacciones para todas las partículas. En [74, 53, 75] se utiliza DEM para la simulación de material y entrenar controladores de excavación.

Existen también los simuladores comerciales usados para el entrenamiento de futuros operadores de maquinaria [76], los cuales ofrecen simuladores capaces de no solo simular la interacción entre la máquina y la pila, sino también otras interacciones como el derrape de las ruedas. Para

la visualización del ambiente, simulan el ambiente completo de una mina. En [59, 24] se ocupa el módulo de AGX Dynamics para la simulación del punto de extracción, el cual se integra con motores de simulación como Unity.

Otro enfoque para la simulación de la interacción es calcular directamente la fuerza ejercida por la pila sobre el balde usando ecuaciones analíticas. Esta metodología tiene el beneficio de que es de bajo costo computacional pero pierde fidelidad. En [77, 78] se utilizan algunos de esto modelos para entrenar controladores de excavación. En este trabajo en específico se utiliza esta metodología con la implementación de la versión extendida de la ecuación fundamental de las mecánicas del movimiento de tierra (*Fundamental Equation of Earth-moving mechanics*, FEE), propuesta en [79].

3.2.1. Ecuación fundamental de las mecánicas del movimiento de tierra

La ecuación fundamental de las mecánicas del movimiento de la tierra, FEE por su nombre en inglés, es un modelo de la interacción entre herramienta y material que entrega las fuerzas resistivas percibidas por la herramienta al excavar. Este modelo fue propuesto por Reece [80] en 1964. La principal limitación de este modelo es que asume que el material es horizontal, debido a que fue diseñado principalmente para terrenos agrícolas. A continuación se describe una reformulación de la FEE propuesta en [79] que permite aplicar la FEE para calcular la fuerza resistiva del material en terrenos con distintas pendientes.

La FEE tiene varios supuestos que permiten reducir la interacción a una sola ecuación analítica. En primer lugar, se asume que las paredes de la herramienta utilizada, como por ejemplo el balde de un LHD, no permiten el movimiento lateral mientras la herramienta está en el material, lo que permite reducir el modelo a dos dimensiones. En segundo lugar, se asume que las fuerzas inerciales son despreciables debido a que las aceleraciones involucradas durante la excavación son lo suficientemente bajas. Por último, se asume que la superficie de falla puede ser aproximada por un plano.

La Figura 3.1 (adaptada de [79]) muestra las distintas fuerzas que interactúan durante la excavación de material, en donde F corresponde a la fuerza resistiva que percibe el balde, generado por la interacción con el material, como por ejemplo la presión de sobrecarga Q . La Tabla 3.2 detalla los distintos componentes presentados en la Figura 3.1.

La reformulación de la FEE detallada en [79] propone que la fuerza total resistiva se divide en tres componentes principales: F_s que corresponde a la fuerza de corte del material, F_g que corresponde a la fuerza de gravedad del material y F_r que corresponde a la fuerza necesaria para moldear el material que está dentro del balde. La fuerza F_s se define por la Ecuación (3.1) y corresponde a la resistencia del material a que se generen fallas.

$$F_s = d^2 w \gamma g N_w + cwdN_c + V_s \gamma g (N_q - 1) \tag{3.1}$$

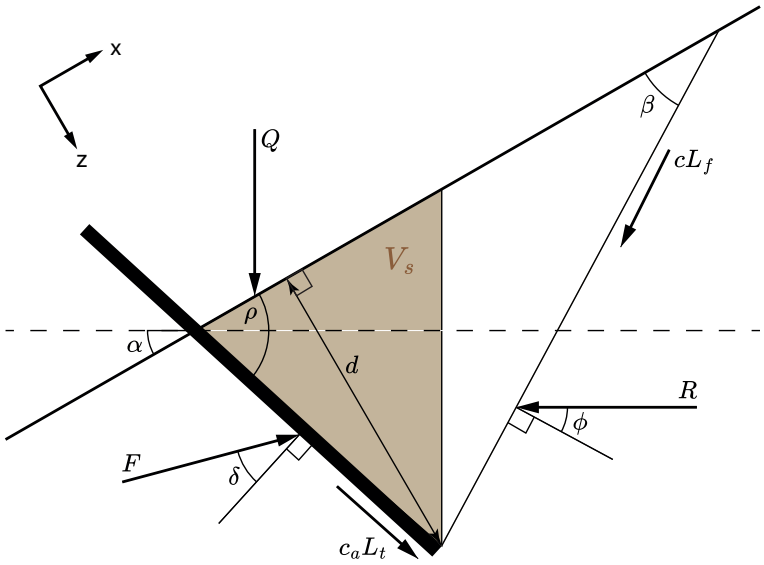

Figura 3.1: Diagrama de fuerzas involucradas en interacción entre la pila de material y el balde (adaptado de [79]).

Tabla 3.2: Componentes de la FEE.

Componente	Descripción
Q	Presión de sobrecarga
R	Fuerza que se resiste al movimiento de cuña de material
F	Fuerza resistiva total
L_t	Largo de la herramienta dentro del material
L_f	Largo de la superficie de falla
ϕ	Ángulo de fricción interna entre partículas de material
δ	Coeficiente de fricción entre material y metal
c	Cohesión del material
c_a	Adhesión del material a la herramienta
ρ	Ángulo entre el material y la herramienta
d	Profundidad a la que está enterrada la punta del balde
g	Fuerza de gravedad
w	Ancho del balde
γ	Densidad del material
β	Ángulo de la falla general
V_s	Volumen de material dentro del balde (zona café)

Los factores de la Ecuación (3.1) se definen por las ecuaciones (3.2), (3.3) y (3.4).

$$N_w = \frac{(\cot\beta - \tan\alpha)(\cos\alpha + \sin(\alpha)\cot(\beta + \phi))}{2[\cos(\rho + \delta) + \sin(\rho + \delta)\cot(\beta + \phi)]} \tag{3.2}$$

$$N_c = \frac{1 + \cot\beta\cot(\beta + \phi)}{\cos(\rho + \delta) + \sin(\rho + \delta)\cot(\beta + \phi)} \tag{3.3}$$

$$N_q = \frac{\cos\alpha + \sin\alpha\cot(\beta + \phi)}{\cos(\rho + \delta) + \sin(\rho + \delta)\cot(\beta + \phi)} \tag{3.4}$$

La fuerza de gravedad F_g que actúa sobre el balde se define por la Ecuación (3.5). Se asume que solo el material cargado dentro del balde provoca una fuerza de gravedad sobre el balde. Cabe notar que la fuerza gravitacional fue restada de la fuerza de corte, debido a que tiene que ser aplicada permanentemente.

$$F_g = V_s\gamma g \tag{3.5}$$

La fuerza de moldeo F_r se define por la Ecuación (3.6). Esta fuerza crece con la cantidad de material cargado en el balde, debido a que se debe comprimir y moldear para que quepa más material.

$$F_r = V_s\gamma g d \tag{3.6}$$

Finalmente, la FEE queda definida por la Ecuación (3.7).

$$F = F_s + F_g + F_r \tag{3.7}$$

Capítulo 4

Metodología

El objetivo principal de este trabajo es resolver el problema de excavación autónoma para un LHD usando DRL. En esta sección se describe el proceso de decisión de Markov implementado en el contexto de entrenar un agente RL para la tarea de excavación con un LHD. Este MDP es implementado en un ambiente simulado utilizando la plataforma Gazebo [81], en donde tanto el LHD como la pila de material son simulados. Toda la etapa de entrenamiento del agente se realiza en el ambiente de simulación, sin embargo, la evaluación del agente se realiza en el ambiente simulado y en el mundo real. La evaluación en el mundo real permite comparar el desempeño del agente RL con el desempeño de la máquina bajo teleoperación y el desempeño del algoritmo de control presentado en [62].

En esta sección se describe en primer lugar la simulación de la máquina LHD utilizada y el ambiente simulado en donde se ejecuta el entrenamiento. Luego, se describe el diseño completo del sistema utilizado para entrenar el agente RL.

4.1. Ambiente y simulación

4.1.1. Simulación de un LHD a escala

El LHD utilizado en este trabajo corresponde a una máquina a escala de un LHD real, con dimensiones reducidas. Este modelo de LHD es utilizado debido a que los experimentos en la vida real se realizan en la misma máquina. La máquina utilizada es un modelo a escala de un LHD real, en donde este LHD mide de largo 1.60 metros y tiene un balde diseñado para 30 kg de material, mientras que un LHD tiene por ejemplo 11.5 metros de largo y un balde diseñado para 14 toneladas de material.

La Figura 4.1 muestra una vista lateral del LHD utilizado. La máquina tiene tracción en las cuatro ruedas. El brazo de levante es la articulación que maneja la altura del balde, mientras que la barra de inclinación maneja la inclinación del balde. Esta configuración se denomina pantógrafo y causa que al levantar el balde sin accionar la barra de inclinación, la barra de inclinación se ajusta sola y mantiene la inclinación del balde. Para la tarea de excavación se utilizan los actuadores del brazo de la máquina y las ruedas para mover la máquina. El modelo también cuenta con la

articulación central que permite cambiar la dirección, pero esta no es utilizada durante la operación de carguío. El no uso de la articulación central es determinado a partir de conocimiento experto, en que los operadores profesionales de LHD evitan utilizar la articulación en el carguío puesto que pierden velocidad para impactar la pila de material y el entierre del balde es menor, o también se podrían generar otros problemas como que se levante el chasis trasero y golpée los muros. En lo que sigue de este documento, la articulación del brazo de levante será denominado como *boom* y la articulación de la barra de inclinación será denominado *bucket*. Los puntos rojos marcados en la Figura 4.1 indican dónde rotan las articulaciones del *boom* y el *bucket*.

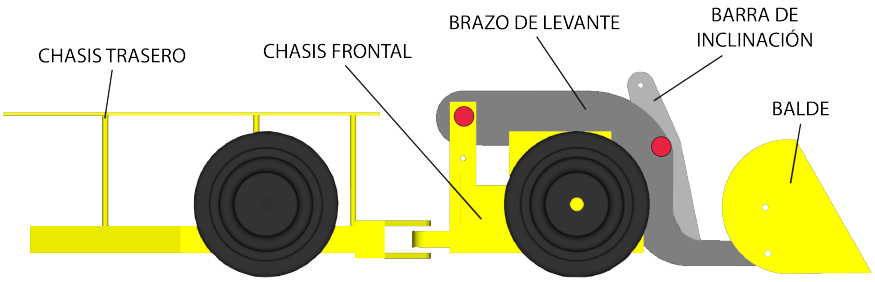

Figura 4.1: Diagrama con vista lateral del LHD a escala utilizado. Se muestran las partes más importantes del LHD, que corresponden al cuerpo y brazo. Los puntos rojos marcan dónde están actuadas las articulaciones.

4.1.2. Ambiente de simulación

Para simular la pila de material se desarrolla un *plugin* para el simulador de Gazebo que calcula la fuerza total ejercida por la pila de material y la aplica al LHD. Esta fuerza se calcula utilizando la FEE extendida presentada en la Sección 3.2.1. Utilizar una ecuación analítica para calcular las fuerzas ejercidas sobre el balde permite bajar el costo computacional de la simulación y consecuentemente acelerar el proceso de entrenamiento. Sin embargo, con este método se pierde fidelidad en las interacciones entre la herramienta y el material. Cabe resaltar que no se simula ningún cuerpo para el punto de extracción y el material, es decir que para el simulador no hay fuerzas de contacto que se calculen directamente. Este problema específico es abordado más adelante en el diseño del agente RL. A continuación se detalla la aplicación de la FEE para que sea compatible con la tarea a resolver.

Modelo 2D a modelo 3D

Como fue mencionado previamente, la FEE es una ecuación que entrega la fuerza que ejerce la pila de material en dos dimensiones. Para aplicar la FEE en tres dimensiones, la pila de material se divide en columnas de ancho w y cada columna es divida en *voxels*. Esto se muestra en la Figura 4.2, en donde la zona pintada de rojo corresponde a una columna y cada cubo corresponde a un *voxel*. Para cada columna se calcula la FEE correspondiente y la suma total de las fuerzas de cada columna corresponde a la fuerza total ejercida sobre el balde.

La división de la pila de material en secciones permite agregar variabilidad al material, puesto que cada sección puede ser configurada con distintos parámetros. Esto se utiliza para simular la presencia de rocas con distintos tamaños dentro de la pila, lo cual es usual cuando se trabaja con material tronado. El efecto de rocas con distintos tamaños en la pila de material es simulado mediante la variación del parámetro de la densidad de cada *voxel*. Un *voxel* con una densidad alta

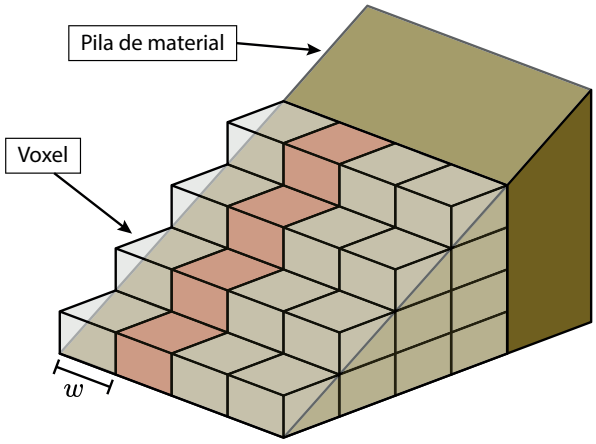

Figura 4.2: Diagrama modelo de pila con *voxels*.

permite simular la presencia de una roca de gran tamaño en la columna correspondiente, puesto que genera dificultad del paso de la punta del balde al aumentar la fuerza entregada por la Ecuación (3.7). Es decir, sea γ_v la densidad del *voxel* donde se encuentra la punta del balde, como las Ecuaciones (3.1), (3.6) y (3.5) son directamente proporcionales a la densidad de material γ, entonces con una densidad alta, las fuerzas calculadas aumentan. Al momento de iniciar la simulación y crear una pila de material, a cada *voxel* se le asigna una densidad dentro de un rango de valores, luego en cada episodio se asigna aleatoriamente un valor de densidad que esté dentro del rango a cada *voxel*.

Respuesta a velocidad de la máquina

La FEE solo varía según la posición del balde dentro de la pila y de los parámetros de la pila. Debido a esto, incluso si la máquina no se está moviendo pero tiene el balde dentro de la pila, se va a generar una fuerza sobre el balde. Se aplica un filtro a la fuerza total aplicada sobre el balde, el cual limita la fuerza máxima que se puede aplicar sobre la máquina según el comando de velocidad enviado a las ruedas. La calibración de este filtro, junto con la configuración de los parámetros de roce entre las ruedas y el suelo, se realizaron mediante ensayo y error para alcanzar un comportamiento lo más cercano a la realidad posible. Sea u_{wheels} el comando de velocidad de las ruedas y K_v un parámetro configurado experimentalmente, la Ecuación (4.1) define la fuerza que se ejerce sobre la máquina, donde F_{FEE} es la fuerza calculada con la Ecuación (3.7).

$$F_{applied} = \text{mín}(F_{\text{FEE}},\ u_{\text{wheels}} * K_v) \tag{4.1}$$

Volumen de material cargado

El volumen de material efectivamente cargado se calcula como una parte del volumen total de material, el cual corresponde al volumen delimitado por la trayectoria de la punta del balde dentro de la pila. En la Figura 4.2 se muestra la trayectoria de la punta del balde y el volumen total de material por sobre esta trayectoria.

Sea W_{full} la totalidad del material cargado y K_W un factor determinado experimentalmente, entonces la cantidad de material cargado efectivo W^T se define por la Ecuación (4.2). Mediante prueba y error, se determina que el valor del parámetro K_W no puede ser 1, es decir, no se puede

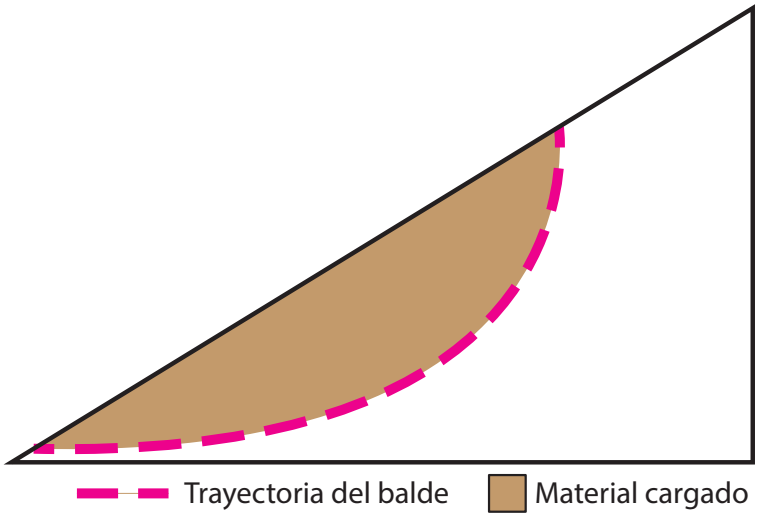

Figura 4.3: Diagrama del material cargado por el LHD. Una parte del volumen delimitado por la trayectoria recorrida es considerada como el material cargado efectivo.

considerar que todo el material es cargado efectivamente dentro del balde, puesto que esto genera valores sobredimensionados para el material cargado comparado con la realidad. Este sobredimensionamiento ocurre puesto que no se consideran efectos del mundo real como por ejemplo que la falla que se genera por cargar material se desplaza o que el material puede desbordarse hacia los lados, entre otros.

$$W^T = W_{\text{full}} * K_W \tag{4.2}$$

Deslizamiento del material

A medida que se retira material desde el punto de extracción, el material por debajo de la punta del balde desliza. La forma en que se desliza el material está determinada por el valor de ϕ del material y el material solo puede moverse dentro de la misma columna. Sea un pilar definido como todos los *voxels* con la misma profundidad en una columna, se compara la altura de dos pilares adyacentes para determinar si desliza el material. Si la pendiente entre los dos pilares α_p es mayor que el ángulo ϕ, entonces desliza el material del pilar más alto al más bajo hasta que la pendiente se estabilice (i.e. $\alpha_p < \phi$).

4.2. Modelamiento del problema

Para resolver el problema de excavación de material con un LHD, la interacción se modela como un POMDP definido por la tupla $(\mathcal{S}, \mathcal{A}, \mathcal{T}, r, \Omega, \mathcal{O})$. El conjunto de estados $\mathcal{S}$ y la función de transición de estados $\mathcal{T}$ quedan definidos por el ambiente. El conjunto de observaciones $\mathcal{O}$ y el conjuntos de acciones $\mathcal{A}$ son seleccionados considerando las limitaciones de la máquina en el mundo real y los sensores disponibles. La función de recompensa r es diseñada arbitrariamente para guiar el entrenamiento de la política. A continuación, se detalla la solución implementada, junto con los distintos conjuntos y funciones que componen el POMDP para esta tarea.

4.2.1. Descripción general de la solución

Para que un carguío sea exitoso, se debe encontrar la maniobra óptima que permita extraer lo máximo de material, en el menor tiempo posible y con el menor daño a la máquina. Además, la maniobra de excavación debe terminar en una posición que permita a la máquina retirarse del punto de extracción manteniendo todo el material dentro del balde.

En este trabajo, el diseño general de la solución consiste en que un agente de aprendizaje reforzado aprenda la política óptima (asociado al POMDP definido para esta tarea) que permita llevar la punta del balde hasta una zona objetivo con cierta pose objetivo cargando la mayor cantidad de material posible. La Figura 4.4 ilustra cómo se vería una trayectoria del balde dentro de la pila de material que llega hasta la zona final. El tamaño y la posición de la zona final es dinámica y se ajusta para cada pendiente, esto es descrito con mayor detalle en la Sección 4.2.4, específicamente en la recompensa por zonas.

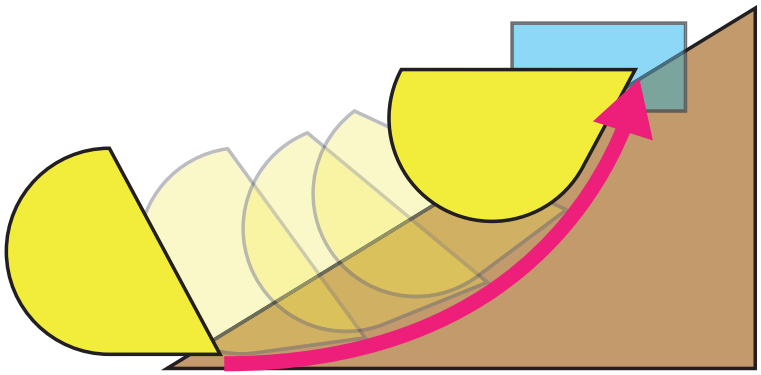

Figura 4.4: Diagrama general de la solución RL al problema de excavación autónoma. El balde del LHD es representado por la figura amarilla y la zona celeste corresponde a la zona objetivo final donde debe terminar la punta del balde.

El agente entrenado debe aprender a cargar material en las distintas condiciones de pila (e.g distintas pendientes y distinto material) y evitar situaciones perjudiciales para la máquina (e.g. resbalamiento de las ruedas). Finalmente, tanto las observaciones como las recompensas están diseñadas para guiar al agente hacia una política que cumpla todas estas condiciones.

4.2.2. Observaciones

La observación principal corresponde a la pose del balde con respecto al punto de extracción. La pose corresponde a la información de la posición (x_{sho}[1], y_{sho}, z_{sho}) y a la información del ángulo de *pitch* ($\theta^{\text{pitch}}_{\text{sho}}$). Para esta observación se implementa una zona de observación que limita los valores de (x_{sho}, y_{sho}, z_{sho}) a un rango $[-1, 1]$ para todos los ejes, con el valor de 0 en el centro de cada eje. El ancho de la zona de observación w_{obs} es determinado por el ancho del punto de extracción, la altura h_{obs} depende de la pose final objetivo del balde y el largo l_{obs} es arbitrario. Para el caso del ángulo $\theta^{\text{pitch}}_{\text{sho}}$, este también es transformado al rango $[-1, 1]$ utilizando como límites los valores máximos y mínimos que puede alcanzar.

Como la pila de material tiene distintas pendientes, la zona de observación se desplaza respectivamente siguiendo la punta de la pila de material pero no cambia sus dimensiones. De esta forma

[1]El subíndice *sho* indica que hace referencial al balde, significa *shovel*.

solo es importante determinar dónde empieza la pila de material para ubicar la zona de observaciones. La Figura 4.5 muestra dos ejemplos de la zona de observación con los valores mínimos y máximos de pendiente.

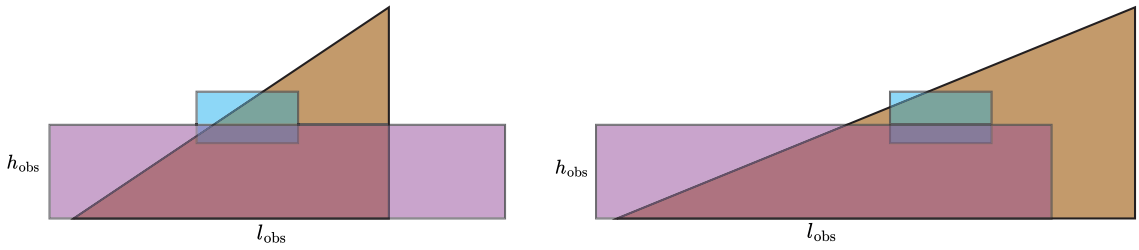

Figura 4.5: Ejemplos zona de observación.

La mayoría de las observaciones corresponde a información intrínseca del estado de la máquina y de sus actuadores. Con respecto al brazo, las observaciones son los estados de las articulaciones (ϕ_{boom}, ϕ_{bucket}) y sus velocidades ($\dot{\phi}_{\text{boom}}$, $\dot{\phi}_{\text{bucket}}$). Con respecto a la máquina, las observaciones son la velocidad del LHD (v_{mach}) y un indicador de la presencia de resbalamiento de las ruedas ($\mathbb{I}_{\text{drift}}$). Para verificar si es que hay resbalamiento de las ruedas, se compara la velocidad de la máquina y la velocidad angular de las ruedas: si es que la máquina no se está moviendo pero alguna rueda está girando significa que hay resbalamiento. Todas las observaciones ϕ_{boom}, ϕ_{bucket}, $\dot{\phi}_{\text{boom}}$, $\dot{\phi}_{\text{bucket}}$ y v_{mach} son transformadas a un rango de valores $[-1, 1]$ mientras que la observación $\mathbb{I}_{\text{drift}}$ solo puede alcanzar valores de 0 o 1.

Se agrega la observación de la pendiente de la pila de material (α), para entregarle información al agente sobre el ambiente. Además, se agrega como observación la distancia entre la punta del balde y el límite de profundidad de la zona final (d_{end}). La Figura 4.6 muestra cómo se determina la observación d_{end}, en donde la zona final corresponde a la zona de color celeste. Se calcula la distancia entre la punta del balde y el valor máximo de la zona final, con el fin de entregar información directa sobre el fin de la zona final. Ambas observaciones α y d_{end} tienen valores en el rango $[-1, 1]$.

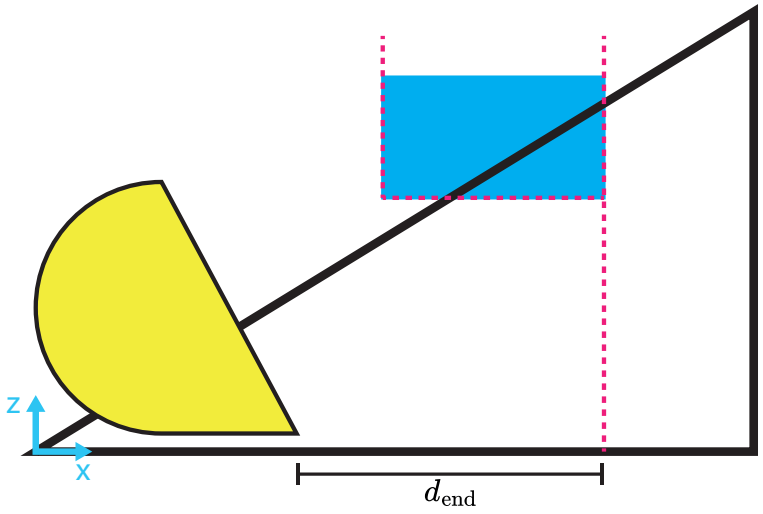

Figura 4.6: Diagrama de la observación del límite de profundidad d_{end}.

La Tabla 4.1 resume todas las observaciones junto con una descripción acotada de cada una.

Tabla 4.1: Observaciones utilizadas para entrenar el controlador RL.

Observación	Descripción
x_{sho}	Posición en eje X de la punta del balde
y_{sho}	Posición en eje Y de la punta del balde
z_{sho}	Posición en eje Z de la punta del balde
$\theta_{\text{sho}}^{\text{pitch}}$	Ángulo de pitch de la punta del balde
v_{mach}	Velocidad del LHD
$\mathbb{I}_{\text{drift}}$	Indicador de resbalamiento
ϕ_{boom}	Ángulo de articulación *boom*
$\dot{\phi}_{\text{boom}}$	Velocidad angular de articulación *boom*
ϕ_{bucket}	Ángulo de articulación *bucket*
$\dot{\phi}_{\text{bucket}}$	Velocidad angular de articulación *bucket*
d_{end}	Distancia entre punta del balde y límite de profundidad
α	Pendiente de la pila de material

4.2.3. Acciones

Durante el carguío, la máquina puede ejecutar tres acciones: u_{wheels} que corresponde al comando de velocidad de las ruedas, u_{boom} que corresponde al comando de velocidad de la articulación *boom* y u_{bucket} que corresponde al comando de velocidad de la articulación *bucket*. Como el LHD tiene tracción en las cuatro ruedas, todas las ruedas ejecutan la misma acción. Normalmente, para realizar un carguío, la máquina se posiciona de forma recta frente al punto de extracción para que pueda chocar con la mayor velocidad posible y que el momento de la máquina permita enterrar el balde más profundo dentro de la pila de material. Debido a esto, no se utiliza la articulación central durante el carguío y solo se utilizan las articulaciones del brazo y las ruedas.

Todas las acciones son limitadas a cierto rango de acción para evitar comportamientos que no aportan a lograr un carguío exitoso. En el caso de las ruedas, la máquina no puede ir hacia atrás, debido a que se busca que el carguío sea en un solo movimiento. En el caso de las articulaciones del brazo, no se permiten acciones que bajen el brazo, es decir, el *boom* no puede ser bajado y el balde no puede ser inclinado hacia adelante, esto para evitar situaciones peligrosas que generen daño a la máquina. La Tabla 4.2 resume todas las acciones que puede ejecutar el agente.

Tabla 4.2: Acciones que puede ejecutar el controlador RL.

Acción	Descripción
u_{wheels}	Velocidad de las ruedas
u_{boom}	Velocidad angular articulación *boom*
u_{bucket}	Velocidad angular articulación *bucket*

4.2.4. Función de recompensa

Una función de recompensa se define como *sparse* cuando el agente recibe dicha recompensa de forma esporádica, generalmente solo al final del episodio. Una función de recompensa se define como *densa* cuando en cada paso de tiempo el agente recibe dicha recompensa.

La función de recompensa en este trabajo están diseñada para guiar al agente hacia una política que pueda cargar la mayor cantidad de material posible y que disminuya el tiempo de ejecución y el daño a la máquina. Algunas componentes de la función de recompensa fueron añadidas para compensar la falta de físicas de contacto con el material debido a la simulación utilizada. Para las componentes descritas a continuación, cada una tiene un superíndice asociado el cual indica si la recompensa se entrega en cada paso de tiempo (superídice *t*, r_α^t) o si se entrega de forma esporádica (superíndice *T*, r_α^t). En este trabajo, la función de recompensa final queda definida por la Ecuación (4.3) y es considerada una función de recompensa *densa*. A continuación, se detallan todas las recompensas implementadas para entrenar la política, las cuales son resumidas en la Tabla 4.3.

$$\begin{aligned} R_{\text{total}} = {} & r^t_{\text{spline}} + r^t_{\text{drift}} + r^t_{\text{bottom}} + r^t_{\text{mid\_goal}} + r^t_{\text{stuck}} + r^t_{\text{inact}} + r^t_{\text{weight}} + r^t_{\text{dump}} \\ & + r^T_{\text{zone}} + r^T_{\text{success}} \end{aligned} \tag{4.3}$$

Tabla 4.3: Resumen de recompensas para agente RL.

Recompensa	**Valor**
Zona	$r^T_{\text{zone}} = \begin{cases} 0 & \text{si el balde está dentro de la zona final} \\ -R_{\text{zone}} & \text{si el balde está fuera de la zona final} \end{cases}$
Trayectoria	$r^t_{\text{spline}} = \begin{cases} 0 & \text{si } \alpha_{\text{lower}} \leq \alpha_{\text{sho}} \leq \alpha_{\text{upper}} \\ -R_{\text{spline}} & \text{otro} \end{cases}$
Drift	$r^t_{\text{drift}} = \begin{cases} 0 & \text{si } (\lvert u^t_{\text{boom}}\rvert > 0 \vee \lvert u^t_{\text{bucket}}\rvert > 0) \wedge \mathbb{I}_{\text{drift}} \\ -R_{\text{drift}} & \text{si } (u^t_{\text{boom}} = 0 \wedge u^t_{\text{bucket}} = 0) \wedge \mathbb{I}_{\text{drift}} \end{cases}$
Fondo	$r^t_{\text{bottom}} = \begin{cases} 0 & \text{si } z_{\text{bottom}} > f(x_{\text{bottom}}) \\ -R_{\text{bottom}} & \text{si } z_{\text{bottom}} \leq f(x_{\text{bottom}}) \end{cases}$
Mid Goal	$r^t_{\text{mid\_goal}} = \begin{cases} 0 & \text{si } x_{\text{sho}} \geq x_{\text{mid\_goal}} \\ -R_{\text{mid\_goal}} * \frac{d_{\text{mid\_goal}}}{d^{max}_{\text{mid\_goal}}} & \text{si } x_{\text{sho}} < x_{\text{mid\_goal}} \end{cases}$
Stuck	$r^t_{\text{stuck}} = -R_{\text{stuck}} * \mathbb{I}_{\text{stuck}}$
Inact	$r^t_{\text{inact}} = \begin{cases} -R_{\text{inact}} & \text{si } (x_{\text{sho}} \geq x_{\text{mid\_goal}}) \wedge \mathbb{I}_{\text{inact}} \\ 0 & \text{otro} \end{cases}$
Weight	$r^t_{\text{weight}} = M_{\text{weight}} * W^t$
Dump	$r^t_{\text{dump}} = \begin{cases} 0 & \text{si } \theta^{\text{pitch}}_{\text{sho}} < \theta^{\text{pitch}}_{\text{thresh}} \\ -R_{\text{dump}} & \text{si } \theta^{\text{pitch}}_{\text{sho}} \geq \theta^{\text{pitch}}_{\text{thresh}} \end{cases}$
Succcess	$r^T_{\text{success}} = \frac{\Delta_{\text{weight}} + \Delta_{\text{pitch}}}{2} * M_{\text{bonus}} * \text{máx}(W^T, W_{\text{target}})$

Recompensa por zonas de término (Zona)

El punto de extracción es dividido en tres zonas principales: (i) la zona final, (ii) la zona permitida, que corresponde a una zona de libre circulación y (iii) la zona restringida. Estas divisiones solo consideran el plano XZ. La Figura 4.7 muestra un diagrama con vista lateral al punto de extracción, en donde el borde grueso marca el perfil del material y cada zona tiene un color asignado. El diseño de esta división es detallado en la Sección 4.2.5. Esta recompensa, al terminar un episodio, indica

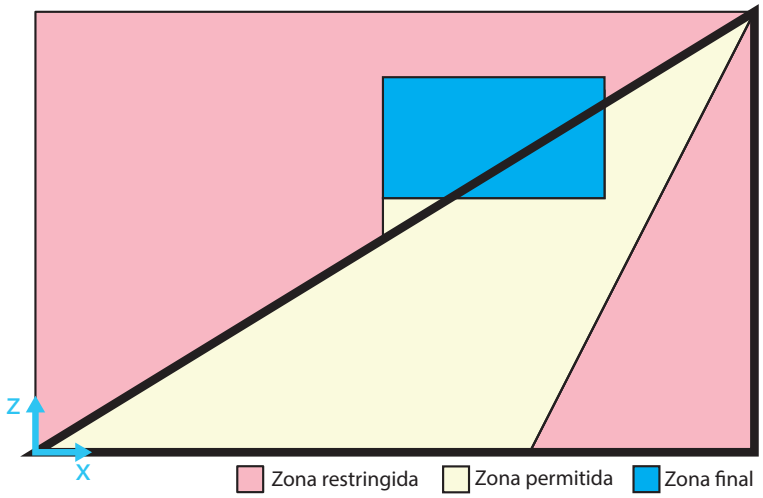

Figura 4.7: División en zonas del punto de extracción. Se penaliza al agente cuando la punta del balde llegue a una zona restringida o el episodio termina en la zona permitida.

al agente si la posición final de la punta del balde es favorable o no. Al finalizar el episodio, se penaliza al agente si la punta del balde está fuera de la zona final. Esta recompensa se define por la Ecuación (4.4).

$$r^{T}_{\text{zone}} = \begin{cases} 0 & \text{si el balde está dentro de la zona final} \\ -R_{\text{zone}} & \text{si el balde está fuera de la zona final} \end{cases} \tag{4.4}$$

Las dimensiones de la zona final quedan definidas por los parámetros $x_{\min}$, $x_{\max}$ y $z_{\min}$. El límite $z_{\min}$ se define de forma arbitraria como la altura esperada para la pose final del balde. Los límites $x_{\min}$ y $x_{\max}$ se determinan tomando en cuenta todo el material disponible en el punto de extracción hasta la profundidad respectiva. Para el caso del límite $x_{\min}$, sea V_{full} el volumen de material determinado por el área A_{full} y el ancho del balde, y sea $W^{\min}_{\text{target}}$ un peso arbitrario, luego, el límite $x_{\min}$ se calcula como la profundidad necesaria para que el volumen $V_{\text{available}}$ tenga un peso $W^{\min}_{\text{target}}$. El caso del límite $x_{\max}$ es similar pero para un peso objetivo mayor $W^{\max}_{\text{target}}$. Para calcular las distancias dinámicas $x_{\min}$ y $x_{\max}$ se utiliza la Ecuación (4.5), donde w es el ancho del balde y γ la densidad del material.

$$x_{\text{dynamic}} = \sqrt{2 \cdot W_{\text{target}} \cdot \tan(\alpha) \cdot w \cdot \gamma} \tag{4.5}$$

Esta formulación busca definir límites en base a la cantidad de material que potencialmente puede cargar el material hasta la distancia determinada, es decir, si el balde avanza a ras de suelo hasta la distancia $x_{\min}$ entonces va a cargar un cantidad de material $W^{\min}_{\text{target}}$. Además, con esta definición se obtienen límites dinámicos que se ajustan a la pendiente de la pila de material. Los valores de

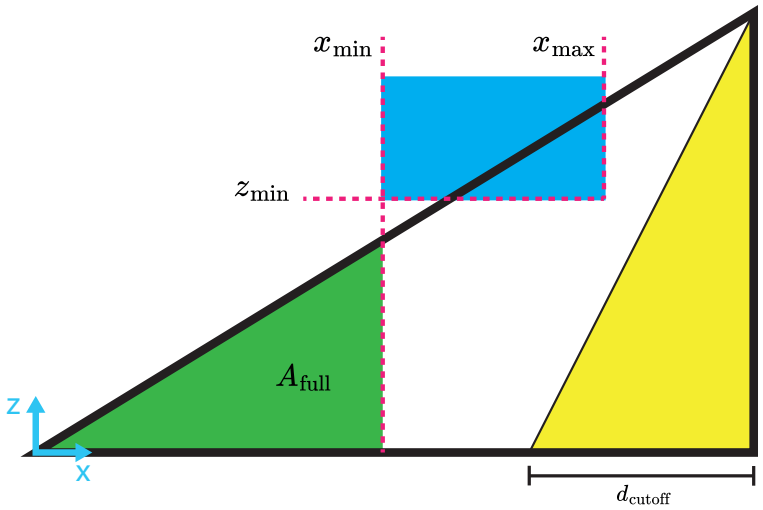

Figura 4.8: Definición de parámetros finales recompensa por zonas que definen las distintas zonas del punto de extracción.

$W^{\text{min}}_{\text{target}}$ y $W^{\text{max}}_{\text{target}}$ son determinados experimentalmente. La Figura 4.8 muestra la definición de los distintos límites de la zona final.

La zona restringida inferior corresponde a una zona que el balde no puede alcanzar físicamente debido a que es muy profundo y el balde no puede llegar físicamente. Dicha zona queda determinada por el parámetro d_{cutoff}, la cual se muestra en la Figura 4.8 como la zona amarilla. La zona restringida superior corresponde a todo el espacio fuera del punto de extracción, exceptuando el espacio por debajo de la zona final. La zona permitida corresponde a todo el resto de espacio disponible.

Recompensa por trayectorias (Trayectoria)

Esta recompensa está diseñada para guiar la trayectoria de la punta del balde. En cada instante de tiempo, se calcula un límite superior y un límite inferior los cuales encierran la trayectoria que sigue la punta del balde para limitar las posibles trayectorias que puede seguir. Si la trayectoria está fuera de los límites, se penaliza al agente. En la Figura 4.9, la flecha naranja corresponde a la trayectoria de la punta del balde y las flechas celestes corresponde a los límites.

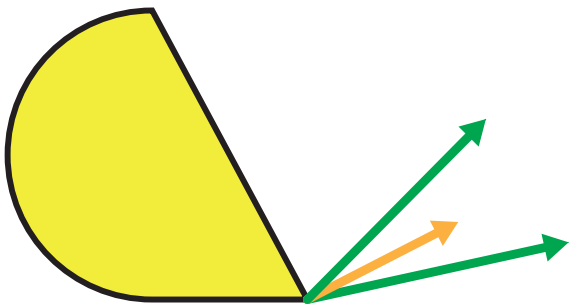

Figura 4.9: Diagrama de recompensa por trayectorias. Las líneas verdes corresponden a los límites que guían la trayectoria y La flecha naranja corresponde al movimiento de la punta del balde. Se penaliza al agente si la flecha naranja está fuera de los límites verdes.

El límite superior de la trayectoria es calculado por etapas. La primera etapa corresponde cuando

la posición en el eje X de la punta del balde es inferior al límite de profundidad mínimo de la zona final, es decir, $x_{\text{sho}} < x_{\text{min}}$, mientras que la segunda etapa corresponde cuando $x_{\text{sho}} \geq x_{\text{min}}$. Se define un punto medio que tiene como altura la mitad de la altura de la pila en la profundidad x_{min}, el cual se puede ver en la Figura 4.10. La Figura 4.10 muestra a la izquierda un ejemplo de la primera etapa y a la derecha un ejemplo de la segunda etapa. Durante la primera etapa, el límite superior corresponde al vector entre la punta del balde y el punto medio definido previamente, mientras que durante la segunda etapa, el límite superior corresponde al vector entre la punta del balde y la esquina inferior izquierda de la zona final, tal como se muestra en la Figura 4.10. Este formato de etapas está diseñado para que durante la primera etapa se guíe al agente a avanzar y no subir el balde, mientras que en la segunda etapa se abre el límite para que pueda subir el balde hasta la zona final.

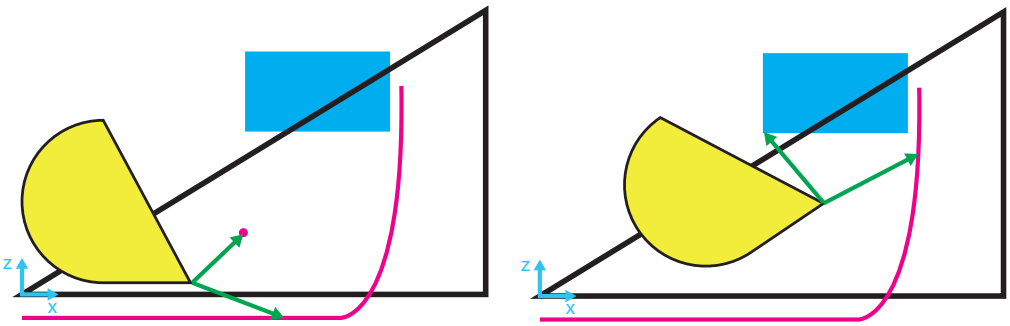

Figura 4.10: Ejemplos de trayectorias guías calculadas para la recompensa por trayectorias. En la figura de la izquierda se tiene un ejemplo de la primera etapa y en la derecha un ejemplo de la segunda etapa.

Para el límite inferior de la trayectoria, se define una curva auxiliar definida como "riel", el cual tiene una forma exponencial y se representa por la curva rosada inferior mostrada en la Figura 4.10. Se definen tres parámetros auxiliares: x_{offset} e y_{offset} que determinan la ubicación del riel, y x_{Δ} que define cuán adelante se calcula el punto en el riel. Luego, el límite inferior corresponde al vector entre la punta del balde y un punto calculado sobre el riel con coordenadas $(y_{\text{riel}}, x_{\text{sho}} + x_{\Delta})$, el cual es calculado utilizando la Ecuación (4.6). Con esta definición del límite inferior, se obtiene un comportamiento en que el límite inferior se dispara y apunta hacia arriba cuando la punta del balde se acerque a x_{max}, lo que guía al agente a que debe subir el balde y terminar el episodio.

$$y_{\text{riel}} = (x_{\text{sho}} + x_{\Delta} - x_{\text{max}} + x_{\text{offset}})^{17} - y_{\text{offset}} \tag{4.6}$$

Finalmente, la recompensa por trayectorias se define por la Ecuación (4.7), en donde α_{sho} es el ángulo de la trayectoria del balde, α_{lower} es el ángulo del límite inferior y α_{upper} es el ángulo del límite superior. Puesto que las trayectorias límites son calculadas en cada instante de tiempo, actualizan también los ángulos límites de la trayectoria del balde. El diseño de esta recompensa busca entregar la menor cantidad de información a priori sobre qué trayectorias debería seguir el balde para introducir la menor cantidad de conocimiento experto.

$$r^{t}_{\text{spline}} = \begin{cases} 0 & \text{si } \alpha_{\text{lower}} \leq \alpha_{\text{sho}} \leq \alpha_{\text{upper}} \\ -R_{\text{spline}} & \text{otro} \end{cases} \tag{4.7}$$

Recompensa por resbalamiento (Drift)

Esta recompensa está diseñada para incentivar políticas que actúen contra el resbalamiento de las ruedas. En cada instante de tiempo, si hay resbalamiento de las ruedas se penaliza al agente si este no ejecuta ninguna acción paliativa contra el drift. El diseño de esta recompensa introduce conocimiento experto al agente, ya que se le indica al agente que debe levantar el *boom* o el *bucket* cuando hay resbalamiento de las ruedas. Esta recompensa se define por la Ecuación (4.8), en donde $\mathbb{I}_{\text{drift}}$ es la observación de resbalamiento.

$$r^t_{\text{drift}} = \begin{cases} 0 & \text{si } (|u^t_{\text{boom}}| > 0 \vee |u^t_{\text{bucket}}| > 0) \wedge \mathbb{I}_{\text{drift}} \\ -R_{\text{drift}} & \text{si } (u^t_{\text{boom}} = 0 \wedge u^t_{\text{bucket}} = 0) \wedge \mathbb{I}_{\text{drift}} \end{cases} \tag{4.8}$$

Recompensa por enterrar el fondo del balde (Fondo)

El fondo del balde corresponde a la parte del balde que está cercana a donde se conecta la articulación y se muestra en la Figura 4.11 como el punto rosado denominado "B". Esta recompensa está diseñada para penalizar al agente por enterrar el fondo del balde. La Figura 4.11 ejemplifica un caso en donde el fondo del balde está enterrado, mostrado por el punto rosado que está dentro del material. Esto es posible debido al método de simulación de material utilizado ya que no se tienen físicas de contacto y toda la fuerza resistiva se determina exclusivamente según la pose de la punta del balde. Por lo tanto, según la trayectoria realizada por la punta del balde, el fondo del balde se puede enterrar en el material. En el mundo real, enterrar el fondo del balde corresponde a presionar con el balde el material, lo que puede llevar a levantar el eje frontal y dañar la máquina. Sea B el punto que representa el fondo del balde con coordenadas $(x_{\text{bottom}}, y_{\text{bottom}}, z_{\text{bottom}})$ y sea $f(x)$ la función que entrega la altura del material en cierta profundidad x, entonces la recompensa por enterrar el fondo del balde se define por la Ecuación (4.9).

$$r^t_{\text{bottom}} = \begin{cases} 0 & \text{si } z_{\text{bottom}} > f(x_{\text{bottom}}) \\ -R_{\text{bottom}} & \text{si } z_{\text{bottom}} \leq f(x_{\text{bottom}}) \end{cases} \tag{4.9}$$

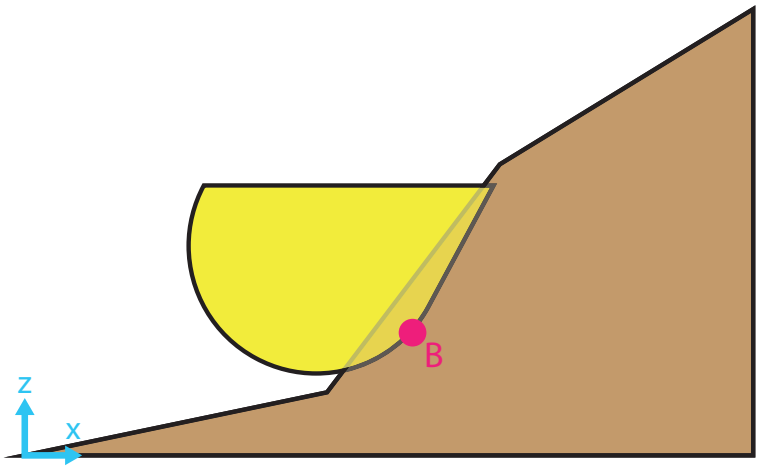

Figura 4.11: Diagrama de recompensa por enterrar el fondo del balde el cual corresponde al punto rosado B. Que este puntó esté dentro de la pila de material significa que el balde está comprimiendo material, lo que puede generar complicaciones como levantar el tren delantero.

Recompensa por enterrar el balde (Mid Goal)

Esta recompensa está diseñada para incentivar al agente a enterrar el balde durante el inicio de la excavación. Al inicio de la excavación, es más fácil que el balde salga del punto de extracción debido a que hay menos material. Por lo tanto, se agrega esta recompensa para incentivar que el agente avance cierta cantidad antes de empezar a levantar el balde. Sea $x_{\text{mid\_goal}}$ una profundidad arbitraria dentro del punto de extracción y $d_{\text{mid\_goal}}$ la distancia entre la punta del balde y $x_{\text{mid\_goal}}$, sea $d^{max}_{\text{mid\_goal}}$ la distancia máxima posible para $d_{\text{mid\_goal}}$, luego, esta recompensa se define por la Ecuación (4.10).

$$r^t_{\text{mid\_goal}} = \begin{cases} 0 & \text{si } x_{\text{sho}} \geq x_{\text{mid\_goal}} \\ -R_{\text{mid\_goal}} \cdot \frac{d_{\text{mid\_goal}}}{d^{max}_{\text{mid\_goal}}} & \text{si } x_{\text{sho}} < x_{\text{mid\_goal}} \end{cases} \tag{4.10}$$

La recompensa por incentivo a enterrar es aplicada solamente cuando $x_{\text{sho}} < x_{\text{mid\_goal}}$ puesto que cuando $x_{\text{sho}} \geq x_{\text{mid\_goal}}$ la recompensa incentivaría a retroceder, lo cual además es imposible debido a las acciones disponibles. La Figura 4.12 muestra un diagrama de la recompensa por incentivo a enterrar y cómo están definidos los parámetros.

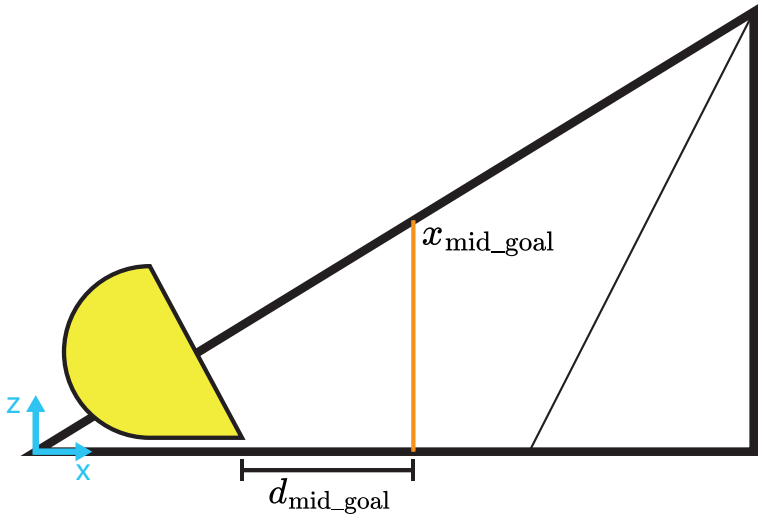

Figura 4.12: Diagrama de recompensa por incentivo a enterrar el balde dentro de la pila de material. Esta recompensa es anulada una vez que la punta del balde supera el valor de $x_{\text{mid\_goal}}$.

Recompensa por atascamiento (Stuck)

Esta recompensa está diseñada para penalizar al agente por atascarse. Se considera que la máquina está atascada cuando la posición de la punta de la pala no se ha movido cierta distancia después de una cantidad arbitraria de pasos de tiempo. Sea $\mathbb{I}_{\text{stuck}}$ un indicador que adquiere el valor 1 cuando la máquina está atascada y 0 cuando la máquina no está atascada, entonces esta recompensa se define por la Ecuación (4.11). La máquina se considera atascada cuando está ejecutando acciones pero la posición de la punta del balde no varía sobre cierto umbral. Este indicador se define en la Ecuación (4.12), en donde Δ_{sho} es la variación de la posición de la punta del balde y Δ_{thresh} el umbral.

$$r^t_{\text{stuck}} = -R_{\text{stuck}} * \mathbb{I}_{\text{stuck}} \tag{4.11}$$

$$\mathbb{I}_{\text{stuck}} = (|u_{\text{boom}}| > 0 \vee |u_{\text{bucket}} > 0) \wedge \Delta_{\text{sho}} < \Delta_{\text{thresh}} \tag{4.12}$$

Recompensa por inactividad (Inact)

Esta recompensa está diseñada para penalizar al agente si este no ejecuta acciones con magnitud mayor a cierto umbral arbitrario. Esta recompensa solo se puede activar luego de que la puntas del balde haya superado la profundidad objetivo definida para la recompensa por enterrar el balde $x_{\text{mid\_goal}}$. Se requiere esta condición ya que forzar ambas acciones del brazo al inicio la excavación incentiva al agente a levantar el brazo y el balde, y en consecuencia el balde se retira anticipadamente de la pila de material. Sea $\mathbb{I}_{\text{inact}}$ el indicador que adquiere el valor de 1 cuando la máquina está ejecutando alguna acción mayor a cierto umbral y 0 cuando no, el cual es definido por la Ecuación (4.14), luego, esta recompensa se define por la Ecuación (4.13).

$$r^t_{\text{inact}} = \begin{cases} -R_{\text{inact}} & \text{si } (x_{\text{sho}} \geq x_{\text{mid\_goal}}) \wedge \mathbb{I}_{\text{inact}} \\ 0 & \text{otro} \end{cases} \tag{4.13}$$

$$\mathbb{I}_{\text{inact}} = |u_{\text{boom}}| < u^{\text{thresh}}_{\text{boom}} \vee |u_{\text{bucket}}| < u^{\text{thresh}}_{\text{bucket}} \tag{4.14}$$

Recompensa por material cargado (Weight)

Esta recompensa está diseñada para incentivar al agente a cargar más material. Se agrega un factor multiplicativo a la recompensa para controlar el rango de valores que puede alcanzar. Sea $M_{\text{weight}} \in \mathbb{R}$ un factor multiplicativo y W^t la cantidad de material dentro del balde en paso de tiempo t, entonces esta recompensa se define por la Ecuación (4.15).

$$r^t_{\text{weight}} = M_{\text{weight}} * W^t \tag{4.15}$$

Recompensa por botar material (Dump)

Botar material corresponde a que material cargado caiga del balde de vuelta a la pila de material. La recompensa por botar material es del tipo *densa*. Esta recompensa está diseñada para penalizar al agente por botar material. En la vida real, debido a la configuración del brazo, al levantar el *boom* sin activar el *bucket*, el balde comienza a inclinarse hacia adelante provocando que material caiga del balde. Además, con el balde inclinado hacia adelante, es más fácil que el LHD se quede atascado en la pila de material. Sea $\theta^{\text{pitch}}_{\text{thresh}}$ el límite para el ángulo $\theta^{\text{pitch}}_{\text{sho}}$, entonces esta recompensa se define por la Ecuación (4.16).

$$r^t_{\text{dump}} = \begin{cases} 0 & \text{si } \theta^{\text{pitch}}_{\text{sho}} < \theta^{\text{pitch}}_{\text{thresh}} \\ -R_{\text{dump}} & \text{si } \theta^{\text{pitch}}_{\text{sho}} \geq \theta^{\text{pitch}}_{\text{thresh}} \end{cases} \tag{4.16}$$

Recompensa por éxito (Success)

Esta recompensa está diseñada a incentivar el terminar un episodio con una cantidad de material determinada y con el balde en una pose final adecuada. En primer lugar, tanto para la cantidad de material cargado al final de la excavación W^T como para el ángulo de la punta del balde $\theta^{\text{pitch}}_{\text{sho}}$ se calcula un factor multiplicativo, $\Delta_{\text{weight}} \in [0, 1]$ y $\Delta_{\text{pitch}} \in [0, 1]$ respectivamente, que representan la distancia de los valores $(W^T, \theta^{\text{pitch}}_{\text{sho}})$ a sus respectivos objetivos $(W_{\text{target}}, \theta_{\text{target}})$. Estos factores se calculan mediante la función definida por la Ecuación (4.17), en donde $\rho(x, x_{\text{target}})$ corresponde a la distancia euclidiana entre dos puntos definida por la Ecuación (4.18) y ρ_{thresh} corresponde a un valor mínimo que puede alcanzar la distancia euclidiana. Esta formulación es utilizada en otros

trabajos de aprendizaje reforzado, y permite obtener una recompensa que incentiva al agente a llegar al objetivo rápidamente [82].

$$\Delta(x) = 1 - \delta(x, x_{\text{target}}) / \max_{x} \delta(x, x_{\text{target}}) \tag{4.17}$$

$$\delta(x, x_{\text{target}}) = -\rho(x, x_{\text{target}}) - ln(\rho(x, x_{\text{target}})) - K_{\text{thresh}} \tag{4.18}$$

En donde la variable K_{thresh} utilizada en la Ecuación (4.18) se define por la Ecuación (4.19).

$$K_{\text{thresh}} = -\rho_{\text{thresh}} - ln(\rho_{\text{thresh}}) \tag{4.19}$$

La función $\Delta(x)$ está diseñada para que alcance un valor de 0 cuando la distancia entre el valor y el objetivo es máxima, y alcance un valor de 1 cuando la distancia está por debajo de un umbral. Luego, la recompensa por éxito se define por la Ecuación (4.20).

$$r^T_{\text{bonus}} = \frac{\Delta_{\text{weight}} + \Delta_{\text{pitch}}}{2} * M_{\text{bonus}} * \max(W^T, W_{\text{target}}) \tag{4.20}$$

Esta función de recompensa está diseñada para alcanzar su valor máximo cuando tanto el ángulo final del balde como la cantidad de material cargado alcanzan su objetivo. Esta es la única recompensa que indica al agente el ángulo final objetivo del balde, el cual está determinado para que el balde no deje caer material. Mediante el término $\max(W^T, W_{\text{target}})$ se limita el valor de esta recompensa, para incentivar que cargue solo la cantidad objetivo de material. El parámetro M_{bonus} es un factor multiplicativo de la recompensa, para controlar cuán grande es el bonus.

4.2.5. Condiciones de término de episodio

La zona final es el principal gatillante de la finalización de un episodio, si la punta del balde está en cualquier parte de la zona final, se da por terminado el episodio. Debido a que se busca que el agente aprenda a realizar el carguío en una sola maniobra, no se permite retirar el balde del punto de extracción antes de haber alcanzado la zona final, salvo por la zona que está justo por debajo de la zona final. Si el agente retira anticipadamente el balde del punto de extracción, se finaliza la excavación. Se busca también que el balde no alcance profundidades mayores al máximo de la zona final, por lo que si el balde alcanza una profundidad mayor que el máximo de la zona final, se finaliza la excavación. Además, si el balde alcanza la zona inferior restringida, se finaliza la excavación.

La Figura 4.13 muestra en rojo las zonas en el punto de extracción que gatillan la finalización del episodio. Adicionalmente, se agrega como condición de término un límite de interacciones con el ambiente (i.e. pasos de tiempo) que la excavación no puede superar.

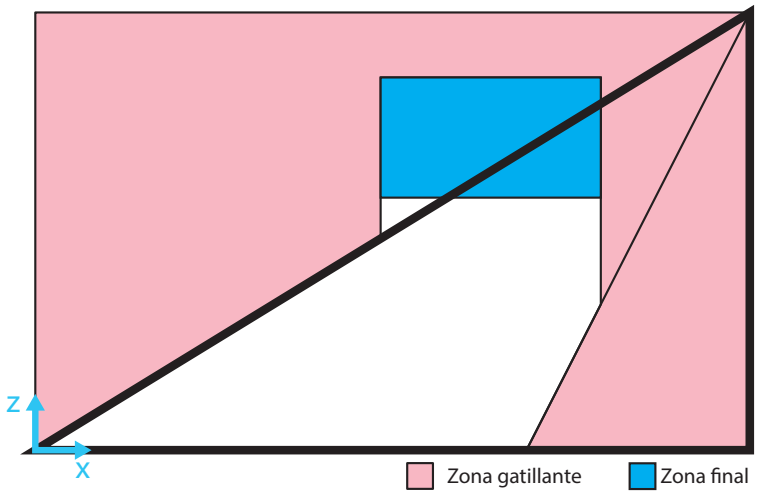

Figura 4.13: Zonas de término de episodio.

Capítulo 5

Resultados

La política aprendida debe ser evaluada tanto en simulación como en el mundo real. Evaluar la política aprendida en el mundo real es importante para verificar si es que el *reality-gap* ha sido superado exitosamente y si el agente RL ha sido capaz de aprender a excavar material. En la siguiente sección, primero se describe el proceso de entrenamiento del agente, luego se evalúa el desempeño del agente en simulación y finalmente se valida la política en el mundo real.

5.1. Evaluación en simulación

5.1.1. Algoritmo

Para el entrenamiento de la política, el algoritmo de DDPG presentado en la Sección 2.2.3 es utilizado. El algoritmo DDPG es utilizado puesto que en [72] y [24], en la vida real solo fue probado un algoritmo entrenado con DDPG. Tanto para el actor como para el crítico del algoritmo DDPG, se utiliza una estructura de red neuronal con una capa de entrada, una capa oculta y una capa de salida. La Figura 5.1 muestra un diagrama de las estructuras de ambas redes utilizadas en DDPG.

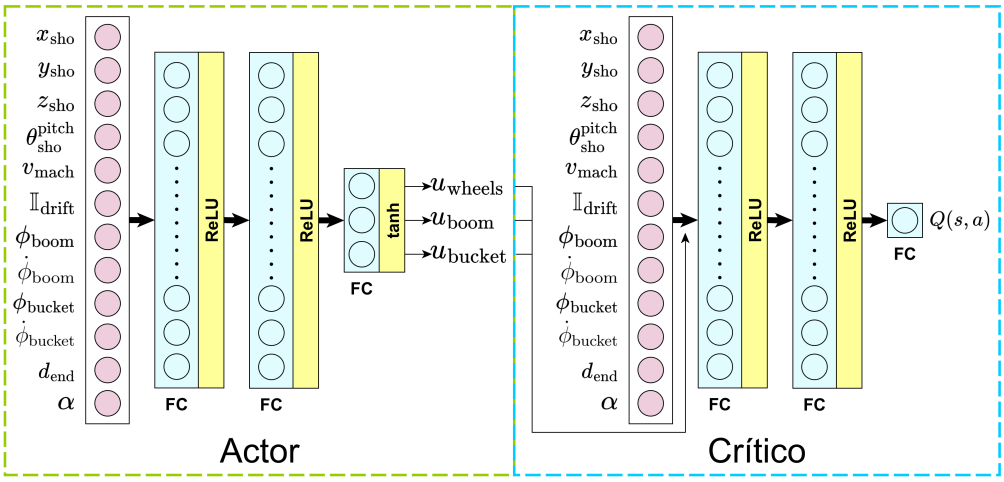

Figura 5.1: Estructura de redes neuronales de actor y crítico.

Todas las capas de las redes neuronales son *Fully Conected* (FC). En el caso del actor, para la capa de entrada y la capa oculta se utiliza la función ReLU como función de activación. Luego, para la capa de salida, se utiliza la función tanh para acotar las salidas al intervalo $[-1, 1]$, las cuales pueden ser escaladas fácilmente a un intervalo $[u_\lambda^{\min}, u_\lambda^{\max}]$, $\lambda \in \{\text{wheels}, \text{boom}, \text{bucket}\}$ para cada acción respectivamente. En el caso del crítico, la red neuronal tiene como objetivo estimar la función de valor $Q(s, a)$, la cual entrega valores en un rango generalmente arbitrario, por lo que se utiliza una función de activación lineal.

5.1.2. Parametrización del entrenamiento

Todo el entrenamiento de la política es llevado a cabo en un ambiente simulado con un punto de extracción similar a un punto de extracción en la realidad. Las fuerzas ejercidas sobre la máquina, causadas por la interacción con el material en el punto de extracción durante la excavación, son calculadas utilizando la FEE modificada descrita anteriormente en la Sección 3.2.1. Se utilizan los parámetros de la grava para calcular las fuerzas [83, 84]. Para la densidad, se asigna un rango de valores para los distintos *voxels* mencionados en la Sección 4.1.2, sin embargo, para calcular el peso del material cargado en el balde se utiliza un único valor de densidad, el cual corresponde al valor obtenido de [83, 84]. Con respecto a la dimensión de los *voxels*, estos tienen una dimensión aproximada de 5 cm por lado, lo que entrega una cantidad de 2100 *voxels* para representar la pila de material. La Tabla 5.1 contiene los parámetros utilizados para calcular las fuerzas resistivas que genera el punto de extracción.

Tabla 5.1: Parámetros de la FEE modificada.

Parámetro	**Valor**
Pendiente de material α [deg]	$[15, 30]$
Densidad de material γ $[kg/m^3]$	$[1700, 2800]$
Densidad de material para peso γ_{w} $[kg/m^3]$	2141
Cohesión c $[N/m^2]$	0
Ángulo de fricción entre material y metal δ [deg]	21.77
Ángulo de fricción interna ϕ [deg]	36.5
Ángulo de reposo [deg]	45

Se utiliza el modelo de LHD descrito en la Sección 4.1.1 y el modelo de pila de material descrito en la Sección 4.1.2 para construir el ambiente completo de entrenamiento de la política en ROS [85] y Gazebo [81]. Un episodio de entrenamiento tiene una duración máxima de 200 pasos de tiempo y tiene como condición de término llevar la punta del balde a alguna de las zonas prohibidas o la zona final que aparecen en la Figura 4.13. En cada episodio se genera una nueva pila de material con distintos parámetros y una pendiente determinada, la cual puede tener valores entre 15 y 30 grados. Luego, se ubica la máquina a cierta distancia del inicio de la pila con tal que pueda alcanzar una velocidad de ataque dentro de un rango predeterminado. La velocidad de ataque es distinta cada episodio para variar las condiciones iniciales del episodio. La Tabla 5.2 resume todos los parámetros de las velocidades de los actuadores y la velocidad de ataque.

La etapa de ataque no es considerada en la política de excavación puesto que la estrategia utilizada en todos los distintos procesos de excavación con LHD es la misma: la máquina embiste a

Tabla 5.2: Parámetros de velocidades de los actuadores.

Parámetro	**Valor**
Velocidad de ruedas $[\mathrm{rad/s}]$	$[0, 2.701]$
Velocidad de actuador *bucket* $[\mathrm{rad/s}]$	$[0, 0.2]$
Velocidad de actuador *boom* $[\mathrm{rad/s}]$	$[-0.191, 0]$
Velocidad de ruedas en ataque $[\mathrm{rad/s}]$	$[2.4, 2.7]$

velocidad máxima contra la pila de material. Un episodio de excavación comienza cuando la punta del balde está a una profundidad de 10 centímetros. Todo el ciclo de control funciona a una frecuencia de 10 Hz. La Figura 5.2 muestra un ejemplo de un inicio de episodio en el ambiente de simulación. El balde se coloca en posición de ataque, que corresponde a que la punta del balde y el fondo del balde están lo más abajo posible sin tocar el suelo.

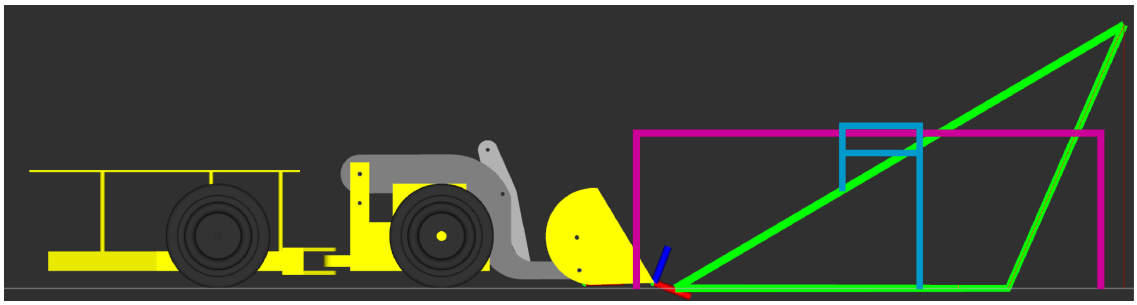

Figura 5.2: Inicio de un episodio de entrenamiento en simulación. El triángulo verde corresponde al perfil de la pila de material, la zona morada corresponde a la zona de observación y la zona celeste corresponde a la zona final. Se agrega un marcador de ejes en la punta del balde.

En la Figura 5.2, la zona marcada de color rosado corresponde a la zona de observación, la zona marcada de celeste corresponde a la zona final y su límite de profundidad, y la zona marcada por verde corresponde a la pila de material. La zona de la pila tiene en la parte inferior un sacado que representa la zona inferior restringida. La Figura 5.2 muestra también la pose de ataque utilizada para todos los carguíos. La Figura 5.3 muestra una visión alternativa del ambiente de entrenamiento.

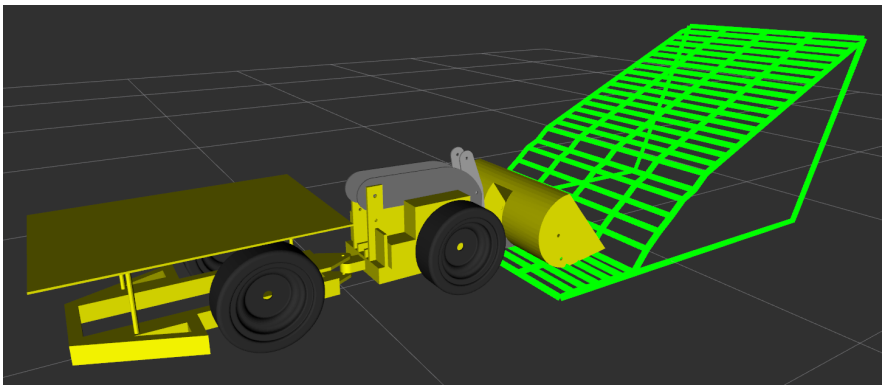

Figura 5.3: Vista alternativa de ambiente de simulación. El perfil de la pila de material cambia cuando se carga material en el balde.

Para el algoritmo de DDPG, la Tabla 5.3 resume los parámetros utilizados para el entrenamiento de la política. Todos los parámetros de DDPG no reportados en la Tabla 5.3 son tomados directamente de [39]. La librería de PyTorch [86] fue utilizada para implementar las redes neuronales del actor y del crítico.

Tabla 5.3: Parámetros de algoritmo DDPG.

Parámetro	**Valor**
Tasa de aprendizaje del actor	0.0001
Tasa de aprendizaje del crítico	0.001
Factor de suavizado λ	0.001
Tasa de descuento γ	0.99
Tamaño máximo del *Experience Replay Buffer*	200000
Tasa del *mini batch*	256
Neuronas en capas ocultas	256

El entrenamiento tiene como máximo 300000 pasos de tiempo y el factor de ruido está diseñado para decaer de forma lineal desde 1 hasta 0.05 después de 250000 pasos de tiempo. Todo el procesamiento computacional es realizado en un computador equipado con un procesador AMD Ryzen 3600 y una unidad de procesamiento gráfico Nvidia GeForce RTX 2060 Super.

Con respecto a la configuración de las recompensas, la Tabla 5.4 muestra los valores de los parámetros de las recompensas utilizadas para entrenar la política. El peso objetivo de material cargado W_{target} es de 30 kg.

Tabla 5.4: Resumen recompensas de parámetros de recompensas.

Recompensa	**Parámetro**	**Valor**
Zona	R_{zone}	2000
Trayectoria	R_{spline}	60
Drift	R_{drift}	5
Fondo	R_{bottom}	5
Mid Goal	$R_{\text{mid\_goal}}$	25
Stuck	R_{stuck}	5
Inact	R_{inact}	50
Weight	M_{weight}	1
Dump	R_{dump}	20
Succcess	M_{bonus}	40

5.1.3. Resultados en simulación

Dos políticas distintas son entrenadas para resolver la tarea de excavación autónoma. La primera corresponde a una política con un espacio de acciones continuo, en donde los actuadores del brazo pueden ejecutar todo el rango de acciones descrito anteriormente. La segunda corresponde a una política con un espacio de acciones discreto, en donde los actuadores del brazo solo pueden ejecutar

los valores extremos del rango de acciones descrito anteriormente. De aquí en adelante, la política con espacio de acciones continua es referida como política RLContinua y la política con espacio de acciones discreta es referida como política RLDiscreta.

El LHD utilizado para las pruebas en la realidad (detallado en la Sección 5.2), tiene actuadores ternarios en las articulaciones del brazo, es decir, funcionan siempre a la velocidad máxima en cualquier dirección y tienen la opción de no moverse. La política RLDiscreta es agregada para evaluar si una política entrenada con la dinámica de la máquina objetivo (RLDiscreta) tiene un mejor rendimiento que una política entrenada con todo el rango de acciones (RLContinua) y luego ajustada mediante un filtro de acciones.

Un filtro de acciones decide si la acción se debe ejecutar definiendo un umbral que debe superar la acción, es decir, es una *deadzone* para las acciones entregadas por la política. Para la simulación, la política RLContinua es entrenada sin ningún filtro mientras que la política RLDiscreta es entrenada con un filtro tal que $|u_{\text{boom,bucket}}| > 0.5$ para que las acciones se ejecuten y estas son ejecutadas por el valor máximo posible en la dirección entregada por la política.

Se evalúa el desempeño de las políticas (RLContinua y RLDiscreta) cada 10000 pasos de tiempo mediante la ejecución de 120 carguíos. Durante la evaluación, el ruido de exploración es deshabilitado. Tres métricas son calculadas para evaluar el desempeño de las políticas en simulación.

- **Retorno promedio**: Promedio de los retornos no descontados obtenidos en los episodios evaluados.
- **Material promedio**: Promedio del material total cargado en los episodios evaluados.
- **Success Rate** (SR): Nivel de éxito en los episodios evaluados.

Para considerar que un carguío es exitoso, se debe considerar tanto el material cargado como la pose final del balde. El balde debe estar en un ángulo que permita mantener el material cargado para la movilización posterior al carguío. Luego, se consideran distintos niveles de dificultad para evaluar el SR, los cuales son definidos según la cantidad de material cargado. Sea $\mathbb{I}_{\text{pitch}}$ un indicador que obtiene el valor de 1 lógico cuando el ángulo de la punta del balde supera el umbral necesario para considerar la pose final como válida para navegación, entonces la métrica SR que entrega el nivel de éxito de la excavación se define por la Ecuación (5.1). Se busca que el agente entrenado alcance un nivel SR5 (i.e. SR=5) en todos sus carguíos.

$$SR = \begin{cases} 5 & \text{si } W^T \geq 28\ kg \wedge \mathbb{I}_{\text{pitch}} \\ 4 & \text{si } 28 > W^T \geq 26\ kg \wedge \mathbb{I}_{\text{pitch}} \\ 3 & \text{si } 26 > W^T \geq 24\ kg \wedge \mathbb{I}_{\text{pitch}} \\ 2 & \text{si } 24 > W^T \geq 22\ kg \wedge \mathbb{I}_{\text{pitch}} \\ 1 & \text{si } 22 > W^T \geq 20\ kg \wedge \mathbb{I}_{\text{pitch}} \\ 0 & \text{otro} \end{cases} \tag{5.1}$$

Para evaluar la estabilidad del entrenamiento de las políticas, tanto para el caso *continuo* como para el caso *discreto* se entrenaron cinco modelos. Se utilizan seis semillas distintas para el generador de números aleatorios, de las cuales cinco son utilizadas para entrenar las dos políticas RLContinua y RLDiscreta, y la última semilla es utilizada para la evaluación de los modelos.

La Figura 5.4 muestra la evolución de las distintas métricas al evaluar las políticas cada 10000 pasos de tiempo y la Tabla 5.5 muestra los resultados de las métricas para las políticas después de 300000 pasos de entrenamiento. Los resultados obtenidos muestran que la política entrenada RL-Discreta obtiene mejores resultados en todas las métricas que la política RLContinua, sin embargo, ambas alcanzan valores que indican carguíos exitosos.

Tabla 5.5: Resultados de evaluación en simulación para políticas RLContinua y RLDiscreta.

Política	**Retorno esperado**	**Material cargado [kg]**	**Nivel SR**
RLContinua	1579±173	29.2±0.9	4.5±0.25
RLDiscreta	2384±178	32.9±0.4	4.95±0.12

Tanto los resultados mostrados en la Figura 5.4 y la Tabla 5.5 muestran que en ambos casos las políticas son capaces de ejecutar carguíos exitosos. La Figura 5.4 muestra que ambas políticas tienen una evolución similar durante el entrenamiento, alcanzando un nivel estable alrededor de los 100000 pasos de entrenamiento. Salvo por unas excepciones, la política RLDiscreta alcanza mejores resultados que la política RLContinua a lo largo de todo el proceso de entrenamiento en las tres métricas evaluadas. La Tabla 5.5 muestra que la política RLDiscreta alcanza mejores resultados finales en las tres métricas evaluadas, con una diferencia del 50.9 % para el retorno esperado, del 12.7 % para la cantidad de material cargado y del 24.4 % para el nivel de SR alcanzado.

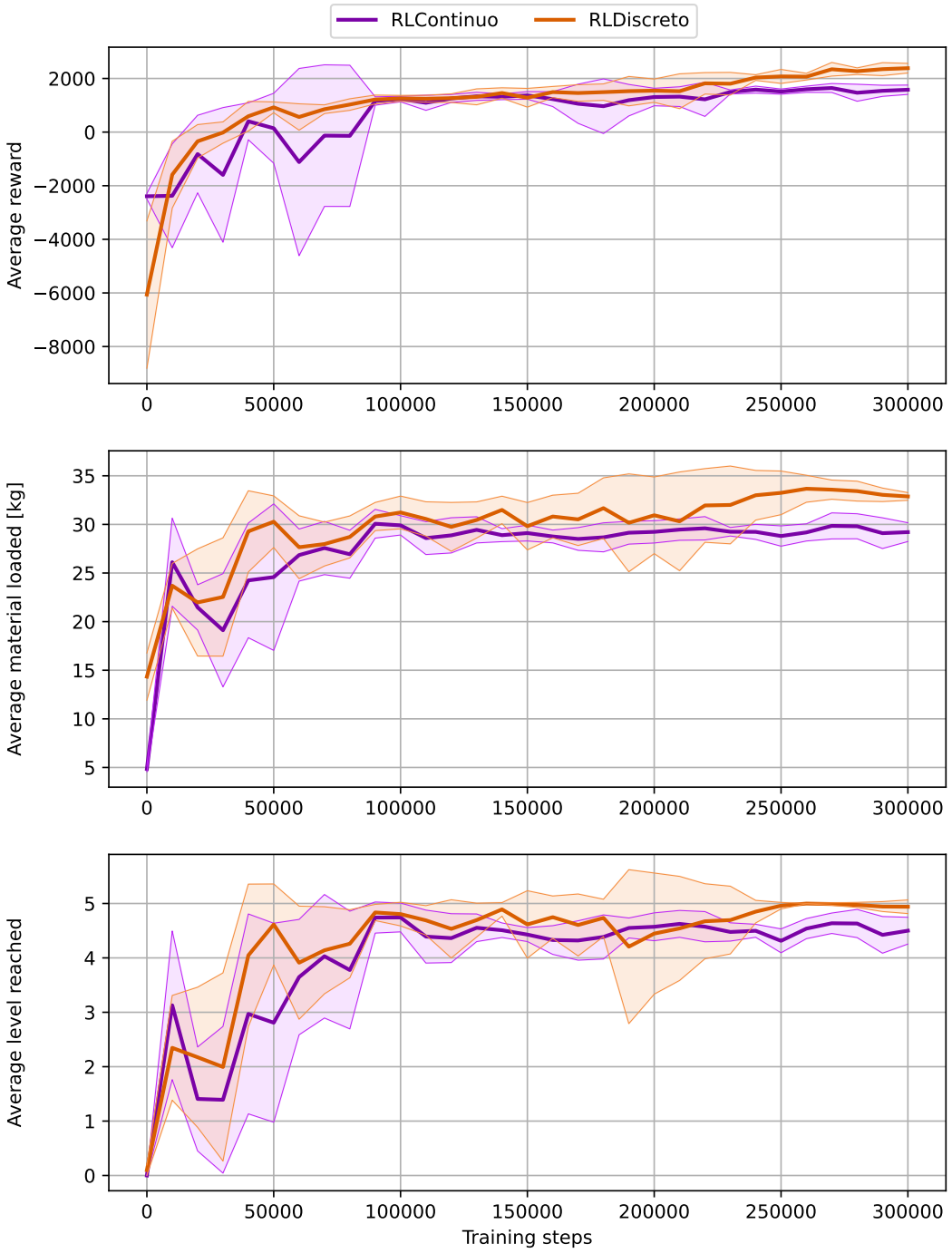

Figura 5.4: Evolución retorno promedio, material cargado promedio y nivel promedio de SR evaluado en simulación, para las cinco políticas entrenadas en los casos continuo y discreto. Las área sombreadas representan la desviación estándar de los resultados obtenidos.

5.1.4. Sensibilidad al tamaño del voxel al entrenar políticas

Con el fin de evaluar el desempeño de la política al cambiar las dimensiones del *voxel*, se entrenan nuevos modelos con dos configuraciones distintas: (i) *voxels* de tamaño aproximado de 9 cm por lado y una densidad de 945 celdas (referido como configuración MEDIA) y (ii) *voxels* de tamaño aproximado de 13 cm por lado y una densidad de 188 celdas (referido como configuración GRANDE). El caso base, que corresponde a la política RLContinua descrita en la Sección 5.1.3 (5 cm por lado y una densidad aproximada de 2100 celdas), es referido como configuración BASE. Los nuevos modelos son entrenados con un espacio de acciones continuo y con las mismas condiciones descritas en la Sección 5.1.2 y la Sección 5.1.3, salvo por el tamaño de los *voxels*. Para cada configuración se entrenan tres políticas distintas, se efectúan 120 carguíos con pendientes aleatorias y se evalúan las mismas métricas presentadas en la Sección 5.1.3. La Tabla 5.6 muestra los resultados obtenidos para las distintas configuraciones.

Tabla 5.6: Resultados de evaluación en simulación para políticas entrenadas con distintos tamaños de voxel.

Configuración	**Retorno esperado**	**Material cargado [kg]**	**Nivel SR**
BASE	1669±168	29.30±1.3	4.48±0.34
MEDIA	1574±171	29.89±1.4	4.53±0.31
GRANDE	1730±121	30.71±0.1	4.73±0.21

En primer lugar, se puede observar de la Tabla 5.6 que para las tres configuraciones el retorno esperado promedio es similar, con un aumento del 3.6 % entre la configuración BASE y la configuración GRANDE, y una disminución del 5.7 % entre la configuración BASE y la configuración MEDIA. Luego, tanto para la métrica del material cargado como para el nivel de SR, se tiene una tendencia a mejorar le desempeño al aumentar el tamaño del *voxel*, con un aumento del 4.8 % en la cantidad promedio de material cargado y del 5.6 % en el nivel promedio de SR alcanzado, ambas comparando los resultados entre la configuración BASE y la configuración GRANDE. Como se detalla en la Sección 5.1.2, la configuración BASE es utilizada para entrenar las políticas en este trabajo. Los resultados que se muestran en la Tabla 5.6 indican que cambios en el tamaño del *voxel* utilizado no generarían cambios significativos en el desempeño de la política entrenada.

5.2. Validación en el mundo real

La validación en el mundo real permite comprobar que las políticas entrenadas realizan carguíos exitosos y que la solución implementada (i.e. el diseño de la función de recompensa y el ambiente de simulación) permiten superar el *reality-gap*. A continuación, se describe el arreglo experimental utilizado para validar las políticas aprendidas y los resultados de distintas excavaciones realizadas.

5.2.1. Arreglo experimental

Se utiliza un LHD con las mismas características al presentado en la Sección 4.1.1, es decir, con las mismas dimensiones y articulaciones, para facilitar el traspaso de la política al mundo real. Con respecto a las articulaciones, todas estas son accionadas mediante pistones hidráulicos, los cuales tienen un bajo retardo entre la recepción del comando y su ejecución. El bajo retardo se debe principalmente al tamaño de la máquina, la cual tiene baja inercia por ser pequeña, y al tamaño de los pistones, los cuales tienen el caudal suficiente para moverse rápidamente. Con respecto a las ruedas, las ruedas traseras comparten un motor eléctrico y están conectadas mediante un diferencial, mientras que las ruedas delanteras cada una tiene un motor eléctrico. La Figura 5.5 muestra el LHD que se utiliza para realizar las pruebas en el mundo real.

Figura 5.5: Máquina LHD a escala utilizada para experimentos en mundo real.

La máquina cuenta con encoders en todos los motores eléctricos y en los puntos de rotación de las articulaciones para obtener la posición y velocidad de rotación de las ruedas y articulaciones. Todos los actuadores tienen las mismas velocidades de rotación que en la simulación, definidas en la Tabla 4.2. La máquina cuenta con un Arduino Due montado en la parte trasera para enviar la información de los sensores y ejecutar los comandos de acción recibidos desde un computador externo. En un computador externo, la información de los sensores es transformada al formato de observación para la política entrenada, la cual entrega posteriormente la acción a ejecutar. El sistema de control funciona a una frecuencia de 10 Hz.

La Figura 5.6 muestra el punto de extracción donde se realizan los distintos experimentos. El diseño de la infraestructura permite devolver el material cargado al punto de extracción desde arriba mediante el uso de la plataforma móvil. Los muros de la infraestructura contienen el material por ambos lados para simular un punto de extracción en túnel. Se utiliza un sensor LiDAR montado verticalmente para medir la pendiente del material.

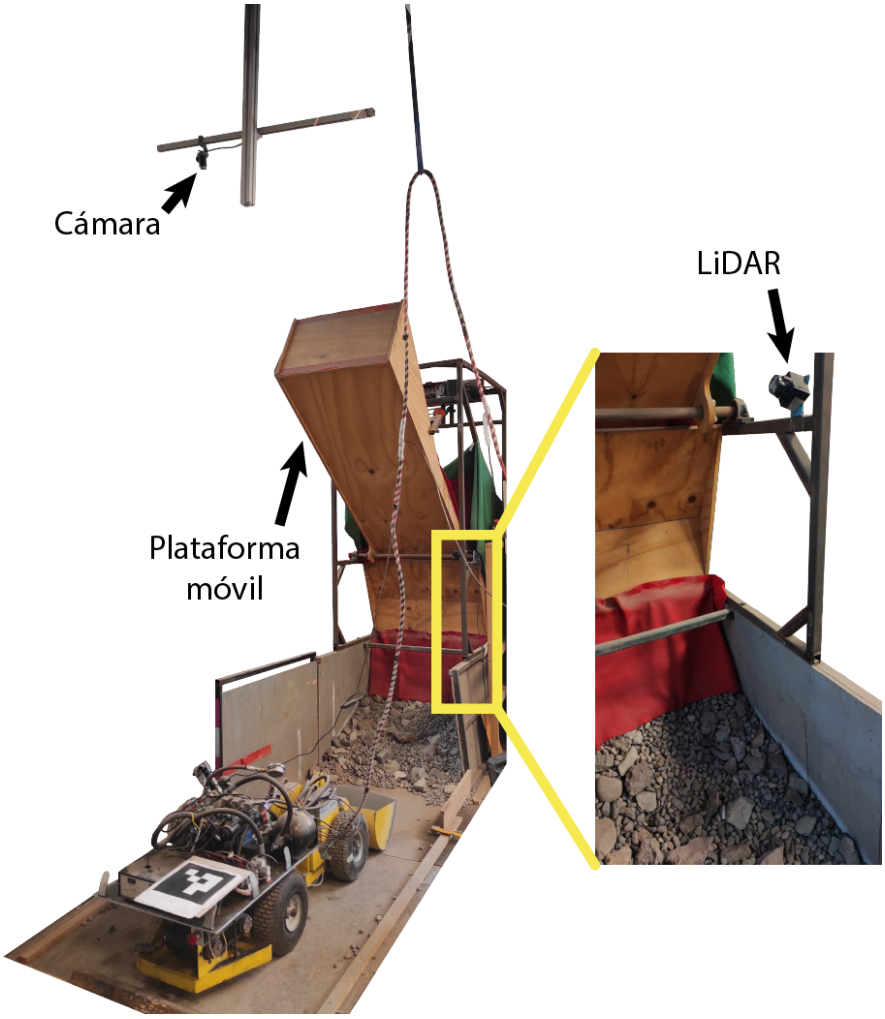

Figura 5.6: Punto de extracción para experimento reales.

Para la localización de la máquina, se utiliza un marcador ArUco [87] ubicado en la parte trasera del LHD. Mediante una cámara montada sobre la infraestructura, se mide la pose del marcador ArUco en todo momento. Con la pose del marcador y la posición de las articulaciones se calcula la pose de la punta del balde. Consecuentemente, esto permite calcular la observación d_{end}, que corresponde a la distancia entre la punta del balde y el límite de profundidad. La ubicación de la cámara se muestra en la Figura 5.6. El cambio de posición del ArUco se utiliza para medir la observación de la velocidad de la máquina v_{mach}.

Para la observación del resbalamiento, se compara la corriente utilizada por los motores eléctricos de las ruedas durante el carguío. Si alguna de las ruedas está resbalando, la corriente utilizada es menor debido a que la carga es menor. Luego, para el indicador de resbalamiento $\mathbb{I}_{\text{drift}}$, se le da el valor de 1 lógico cuando alguna de las ruedas esté resbalando.

Se utiliza material homogéneo y no homogéneo, es decir con rocas, para realizar los distintos carguíos. El material homogéneo consiste en ripio con partículas de tamaño entre 2 cm y 5 cm. El material no homogéneo consiste en el material homogéneo con la adición de rocas de tamaño entre 8 cm y 15 cm. Ambos materiales se muestran en la Figura 5.7.

Figura 5.7: Materiales utilizados para experimentos.

Para la validación realizada en el mundo real, se utilizan las mejores políticas aprendidas al entrenar en simulación. La Tabla 5.7 muestra los resultados de las métricas evaluadas en 120 episodios distintos en simulación, para ambas políticas seleccionadas para la validación en el mundo real.

Tabla 5.7: Resultados de evaluación en simulación para agentes finales seleccionados para validación.

Política	**Retorno esperado**	**Material cargado [kg]**	**Nivel SR**
RLContinua	1860±301	30.5±3.5	4.68±0.53
RLDiscreta	2272±249	32.5±1.8	4.99±0.09

En los experimentos descritos a continuación, el posicionamiento inicial del LHD para todos los experimentos es realizado manualmente. La articulación central es ajustada para que la máquina quede recta frente a la pila y el brazo es puesto en posición de ataque. La punta del balde es posicionada a 50 ± 10 cm del inicio de la pila, distancia que permite que la máquina alcance su velocidad máxima antes de enterrar el balde. Para todos los experimentos, la velocidad de ataque corresponde a la velocidad máxima que puede alcanzar la máquina, la cual es de 2.7 km/h aproximadamente. Para evaluar el desempeño con distintos valores de pendientes, los experimentos son realizados con la pila de material configurada con una pendiente de 15 grados, 20 grados, 25 grados y 30 grados. Con respecto al filtro de acciones, para los carguíos realizados con la política RLDiscreta se aplica el mismo filtro mencionado en la Sección 5.1.3, es decir, las acciones solo son ejecutadas si $|u_{\text{boom,bucket}}| > 0.5$. No obstante, también se aplica un filtro para cargar con la política RLContinua, el cual corresponde a $|u_{\text{boom,bucket}}| > 0.2$. Este último filtro es aplicado con el fin de proteger las válvulas hidráulicas frente a rápidos accionamientos que podría enviar el controlador RL.

Ambas políticas RLContinua y RLDiscreta son comparadas con otros dos sistemas de carguío: el algoritmo desarrollado en [62] (de ahora en adelante designado como algoritmo Tampier) y la teleoperación de la máquina (de ahora en adelante designado como agente Teleop). Los carguíos teleoperados fueron realizados por tres teleoperadores distintos, los cuales gracias a experiencias pasadas tienen conocimiento sobre estrategias para cargar material con LHDs. Estos dos agentes extras se consideran como *baselines* para evaluar el desempeño del agente RLContinuo.

La maniobra de carguío que realiza el algoritmo Tampier se divide en dos etapas. Primero, el LHD ataca la pila a máxima velocidad para enterrar el balde dentro de la pila de material. Luego,

la segunda etapa es gatillada con un detector de colisiones, que en base a la presión hidráulica en los pistones del brazo, determina si es que el balde está enterrado lo suficiente. La segunda etapa consiste en alternar el levantar el brazo y acelerar mientras se revisa cuánto están resbalando las ruedas. Esta etapa tiene cuatro estados posibles: (i) avanzar a máxima velocidad, (ii) avanzar mientras levanta el *bucket*, (iii) levantar el *bucket* y (iv) avanzar mientras baja el *boom*. Para cambiar entre estados, el agente revisa si es que el resbalamiento de las ruedas supera cierto umbral. Si es que el resbalamiento es muy alto, para reducirlo el algoritmo pasa al estado (ii) o (iii) ya que levantar el brazo mejora el agarre de las ruedas. El estado (i) y (iv) están pensados para avanzar y cargar la mayor cantidad de material posible antes de comenzar a subir el brazo.

Para el caso de teleoperación, se busca simular la teleoperación remota del LHD, en donde los operadores tienen principalmente información visual de lo que está ocurriendo mediante cámaras montadas en el LHD. De esta forma, una cámara es montada en el chasis frontal de la máquina y entrega una visión como la que se observa en la Figura 5.8. Los teleoperadores controlan el LHD utilizando un control de Xbox.

Figura 5.8: Ejemplo de vista que tiene el operador al efectuar un carguío teleoperado.

Para todos los resultados que se muestran a continuación, se realizan 5 carguíos por cada configuración de pendiente para la política RLContinua, la política RLDiscreta y el algoritmo Tampier. En el caso del agente Teleop, cada teleoperador realiza tres carguíos por pendiente para alcanzar un total de nueve carguíos. La condición de término para los cuatro agentes está relacionada con el ángulo y la altura del balde. Con el fin de proteger los motores del LHD a escala, se agrega una condición de término extra por tiempo, en que las maniobras tienen un tiempo máximo de 10 segundos. Esto permite evitar situaciones en donde la máquina solo trata de avanzar a máxima potencia contra la pila durante mucho tiempo sin acciones que muevan el balde, lo que podría quemar los motores. Se considera que un carguío no es exitoso cuando la cantidad de material cargada es menor a los 20 kg.

En primer lugar, se presentan los resultados de los carguíos realizados en material homogéneo. Para el caso de material no homogéneo, se repiten los mismos experimentos que para el caso de material homogéneo, es decir, se realiza la misma cantidad de carguíos. Finalmente, se realizan también carguíos en material homogéneo con ambas políticas entrenadas, pero se les entrega información errónea sobre la pendiente de la pila de material, para así evaluar los efectos de errores en la observación de pendiente.

5.2.2. Excavación en material homogéneo

En primer lugar, la Tabla 5.8 presenta la tasa de éxito por pendiente para todos los controladores evaluados. Salvo por el algoritmo Tampier, todos los controladores lograron ejecutar la totalidad de los carguíos de forma exitosa. Los fallos en el carguío del algoritmo Tampier se deben a fallos en la detección de la colisión con la pila de material, por lo que no accionaba en ningún momento el balde y procedía a agotar el tiempo de carguío.

Tabla 5.8: Tasa de éxito por pendiente en material homogéneo.

Pendiente [deg]	RLContinuo	RLDiscreto	Tampier	Teleop
15	1	1	0.6	1
20	1	1	1	1
25	1	1	0.8	1
30	1	1	0.8	1

A continuación, se presentan los resultados finales de todos los carguíos realizados en la pila de material homogéneo, los cuales son resumidos en la Tabla 5.9. Estos resultados solo consideran aquellos carguíos que son considerados exitosos. Como se puede observar en la Tabla 5.9, la política RLDiscreta obtiene los mejores resultados en las cuatro categorías evaluadas. Con respecto al material cargado, todos los controladores alcanzan valores similares cercanos a los 25 kg de material cargado, lo que corresponde a un 83.3 % del material objetivo a cargar, el cual es de 30 kg. Con respecto a la duración del carguío, se destaca que el agente Teleop es aquel que más tiempo tomó para realizar los carguíos, demorando más de tres veces que los demás controladores. Esto se explica principalmente con que los operadores debían adaptarse al control de la máquina con el control de Xbox.k La Tabla 5.9 también muestra que la política RLContinua obtiene resultados mejores que el algoritmo Tampier y el agente Teleop.

Como fue mencionado anteriormente, el resbalamiento de las ruedas se debe evitar durante el carguío para minimizar el daño a las llantas y así disminuuir los costos de operación. La Tabla 5.9 muestra que, con respecto al eje delantero, la política RLDiscreta y el agente Teleop alcanzan los mejores resultados con menor variación que la política RLContinua y que el algoritmo Tampier. Esto indica que estos dos controladores tienen un mejor control sobre cuando levantar el balde para volver a tener tracción en el eje delantero. Con respecto al eje trasero, los cuatro controladores presentan valores similares, sin embargo, cabe destacar los que el algoritmo Tampier alcanza un procetaje de resbalamiento en el eje trasero de más de 90 % para las pendientes de 20, 25 y 30 grados. Esto se debe a la estrategia que tiene el algoritmo, que generalmente nunca dejar de acelerar hasta terminar la excavación y tratar de disminuir el resbalamiento al levantar el balde.

Tabla 5.9: Resultados de carguíos realizados en material homogéneo para todos los controladores.

	S[2]	RLContinuo	RLDiscreto	Tampier	Teleop
Material cargado [kg]	15	22.4 ± 2.36	23.28 ± 1.43	23.13 ± 2.24	25.21 ± 0.86
	20	24.18 ± 1.22	23.48 ± 0.95	23.87 ± 2.41	24.27 ± 0.73
	25	26.08 ± 0.39	25.56 ± 0.9	25.14 ± 0.88	24.04 ± 1.89
	30	27.24 ± 0.79	28.12 ± 0.91	25.92 ± 0.33	25.04 ± 1.33
	$\bar{X}$[3]	**24.97 ± 1.84**	**25.24 ± 1.84**	**24.52 ± 1.08**	**24.64 ± 0.50**
Tiempo [s]	15	4.06 ± 0.23	4.38 ± 0.5	2.58 ± 1.35	10.42 ± 7.03
	20	6.12 ± 2.76	3.4 ± 0.5	4.89 ± 1.51	13.05 ± 6.68
	25	3.78 ± 0.28	4.52 ± 0.19	5.74 ± 0.24	16.39 ± 8.03
	30	3.98 ± 0.2	4.72 ± 0.36	5.56 ± 0.1	15.91 ± 6.84
	$\bar{X}$	**4.48 ± 0.95**	**4.25 ± 0.51**	**4.69 ± 1.26**	**13.94 ± 2.40**
Drift delantero[%]	15	47.99 ± 11.32	26.45 ± 5.75	92.53 ± 9.13	28.64 ± 10.44
	20	70.59 ± 18.82	22.94 ± 11.38	61.67 ± 26.55	38.1 ± 18.08
	25	25.09 ± 7.93	17.84 ± 2.92	46.19 ± 3.19	36.17 ± 18.34
	30	42.62 ± 16.33	16.32 ± 3.84	36.44 ± 2.14	25.07 ± 13.47
	$\bar{X}$	**46.57 ± 16.25**	**20.89 ± 4.04**	**59.21 ± 21.24**	**32.00 ± 5.34**
Drift trasero [%]	15	35.12 ± 19.59	52.59 ± 4.55	31.75 ± 33.45	78.03 ± 16.75
	20	88.91 ± 8.4	60.61 ± 4.01	95.77 ± 4.09	75.36 ± 21.73
	25	74.87 ± 5.46	67.35 ± 4.5	96.94 ± 2.21	90.91 ± 9.76
	30	89.24 ± 6.6	88.5 ± 6.1	93.34 ± 1.64	73.56 ± 21.06
	$\bar{X}$	**72.03 ± 22.09**	**67.19 ± 13.41**	**79.45 ± 27.57**	**79.47 ± 6.80**

[2]La columna S (*Slope*) indica la pendiente de la pila de material.

[3]La fila $\bar{X}$ indica el promedio por categoría de todas las pendientes.

La Figura 5.9 muestra el detalle de los carguíos realizados en material homogéneo, mostrando cuanto cargó y demoró cada maniobra realizada. Se puede observar que en las cuatro configuraciones de pendiente, los resultados de las dos políticas entrenadas con aprendizaje reforzado y del agente Tampier son similares, es decir, se concentran cerca de los mismos valores. Luego, como se mencionó anteriormente, el agente Teleop es aquel que demora en este caso la mayor cantidad de tiempo en ejecutar la excavación, lo que se puede observar en la Figura 5.9. La Figura 5.9 también muestra la gran diferencia entre los distintos carguíos realizados. Para complementar, la Figura 5.10 muestra en detalle los resultados de los carguíos realizados por los distintos operadores, en donde se puede observar que hay diferencias en el desempeño entre los distintos operadores.

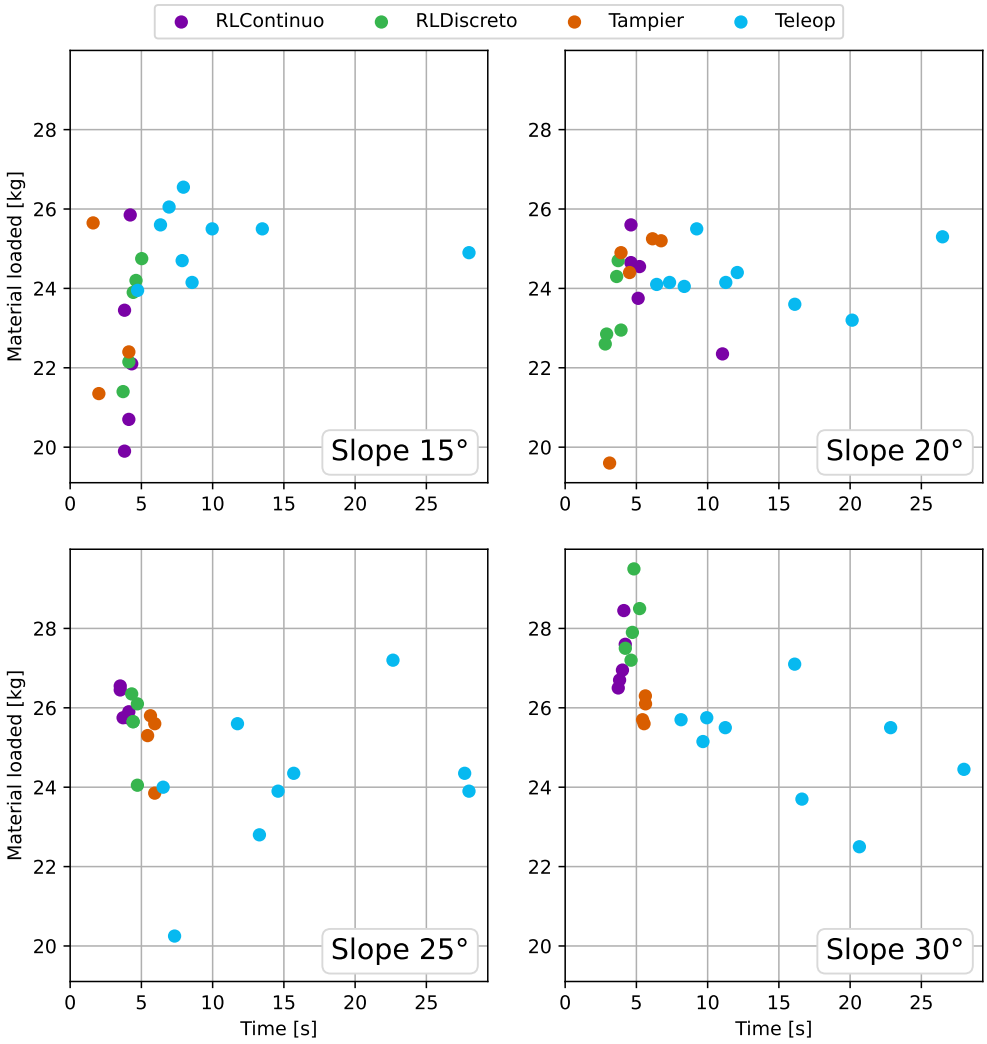

Figura 5.9: Visualización de resultados de carguíos realizados en material homogéneo para todos los controladores.

Para ejemplificar las diferencias de estrategia entre los diferentes controladores, en la Figura 5.11 se muestran las acciones ejecutadas durante un carguío. Debido a la configuración de la máquina, en el caso del *boom* y *bucket*, las acciones negativas son aquellas que levantan las articulaciones. En primer lugar, se tiene que el algoritmo Tampier es el único que durante el carguío ejecuta la acción de bajar el *boom*. Si bien las políticas RL no pueden ejecutar esta acción debido al espacio de acciones aplicado, el agente Teleop tampoco lo ejecuta en ninguno de los otros carguíos

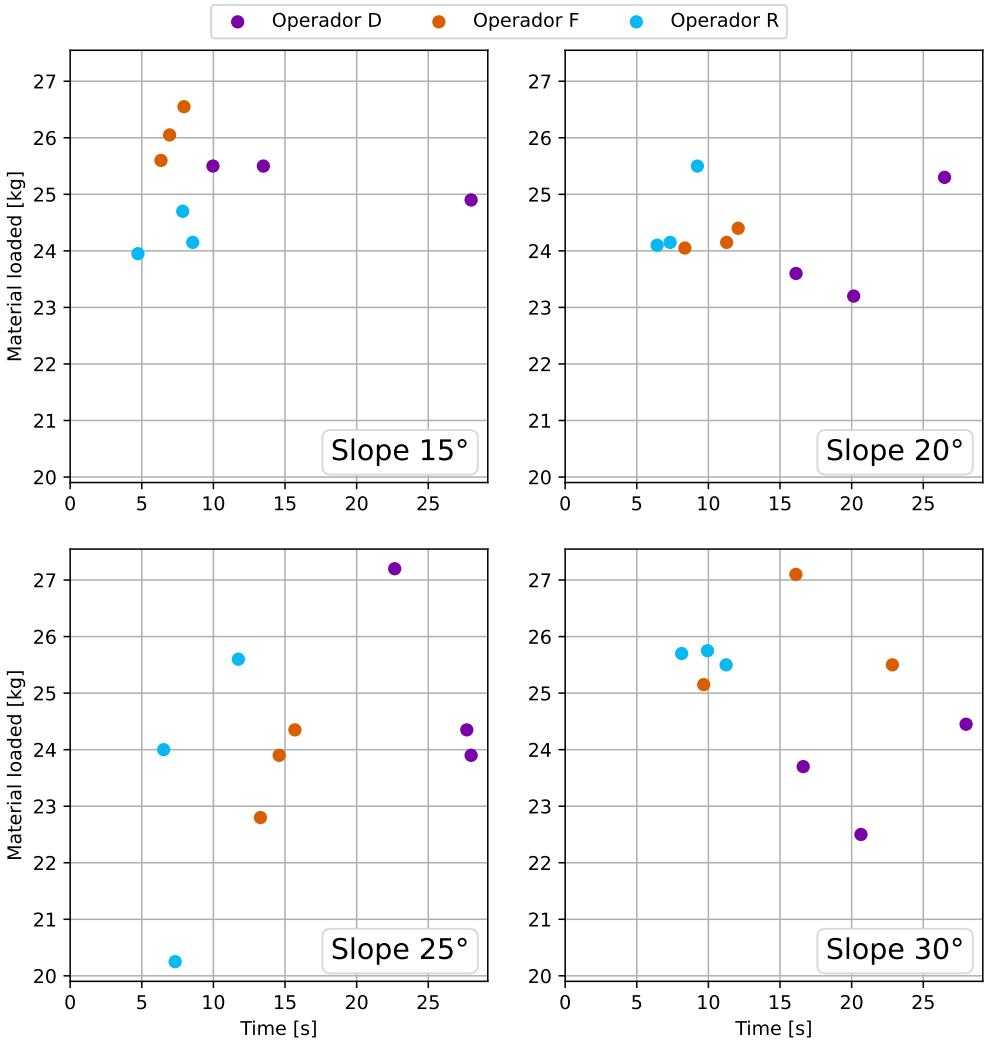

Figura 5.10: Visualización de resultados de carguíos realizados en material homogéneo para distintos teleoperadores.

realizados. Se puede observar en la Figura 5.11 una similitud entre las acciones ejecutadas por las políticas RL, en donde primero se utiliza el *bucket* y luego se activa el *boom* para terminar con la maniobra. Se puede observar además que ambas políticas RL y el agente Teleop frenan antes de terminar la excavación. Este comportamiento es deseado para evitar que la máquina suba las ruedas al material y estas se dañen.

La Figura 5.12 muestra un ejemplo de la trayectoria que sigue la punta del balde durante un carguío para los cuatro controladores evaluados para las distintas pendientes. En esta figura se puede ver que el algoritmo de Tampier funciona por condiciones, en donde se ejecutan subacciones por bloques. Esto se observa en que la trayectoria que sigue la punta del balde es de subir un poco y avanzar lo máximo que puede, y así sucesivamente hasta terminar el carguío. Los carguíos realizados por las políticas RL presentan un comportamiento más fluido que los carguíos realizados por el algoritmo Tampier y los operadores.

A continuación, la Figura 5.13, la Figura 5.14, la Figura 5.15 y la Figura 5.16 muestran todas las trayectorias de los carguíos realizados anteriormente. A partir de las siguientes figuras, se puede

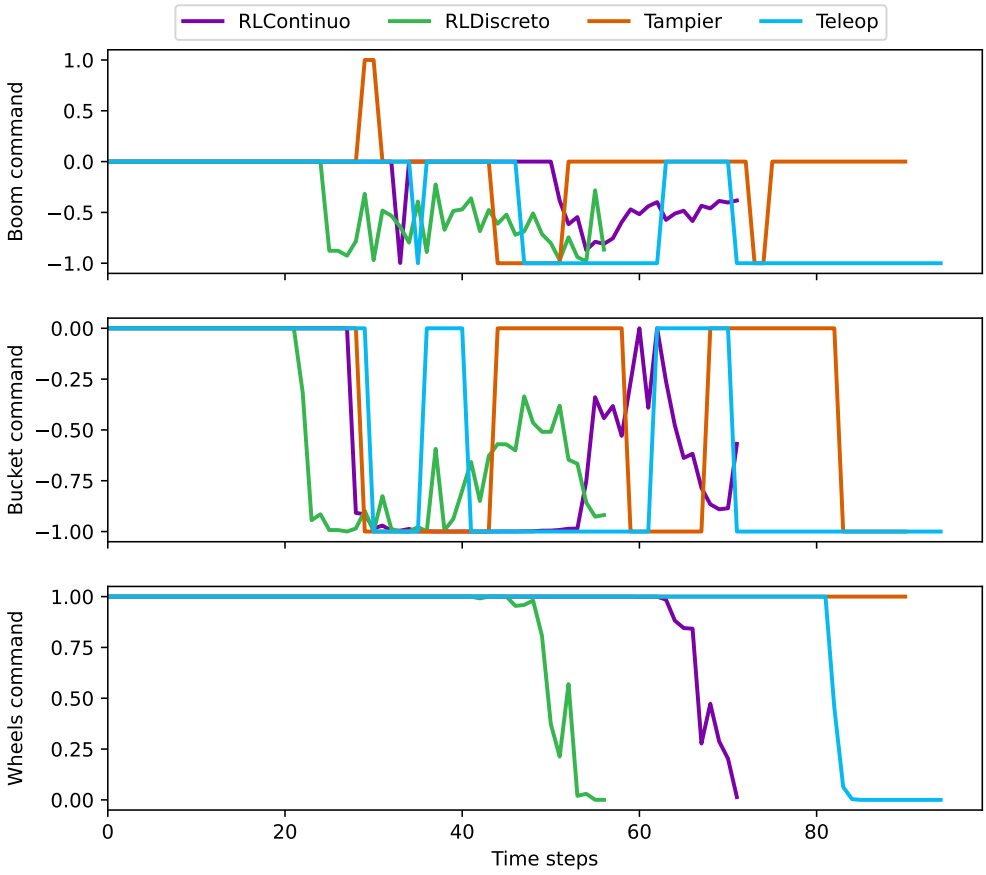

Figura 5.11: Ejemplo acciones de control durante un carguío en material homogéneo.

observar el comportamiento general que todos entierran el balde en la pila de material una cierta distancia antes de elevar el balde y concluir con el carguío. En el caso de las políticas RL, cuyos carguíos se muestran en la Figura 5.13 y en la Figura 5.14, se tienen carguíos fluidos en donde la máquina no se atasca significativamente en la pila de material y el balde termina en una posición que permite retirarse solamente retrocediendo. En el caso del algoritmo Tampier y el agente Teleop, cuyos carguíos se muestran en la Figura 5.15 y en la Figura 5.16, se observa que la estrategia de carguío es avanzar lo máximo posible y luego levantar un poco el balde para ganar tracción y seguir el carguío, lo que genera trayectorias con forma “escalonada”.

Los resultados presentados en esta sección muestran que ambas políticas RL aprendidas son capaces de ejecutar carguíos de forma exitosa y consistente en una pila de material homogéneo. Además, se tiene que la política RLDiscreta obtiene mejores resultados que la política RLContinua.

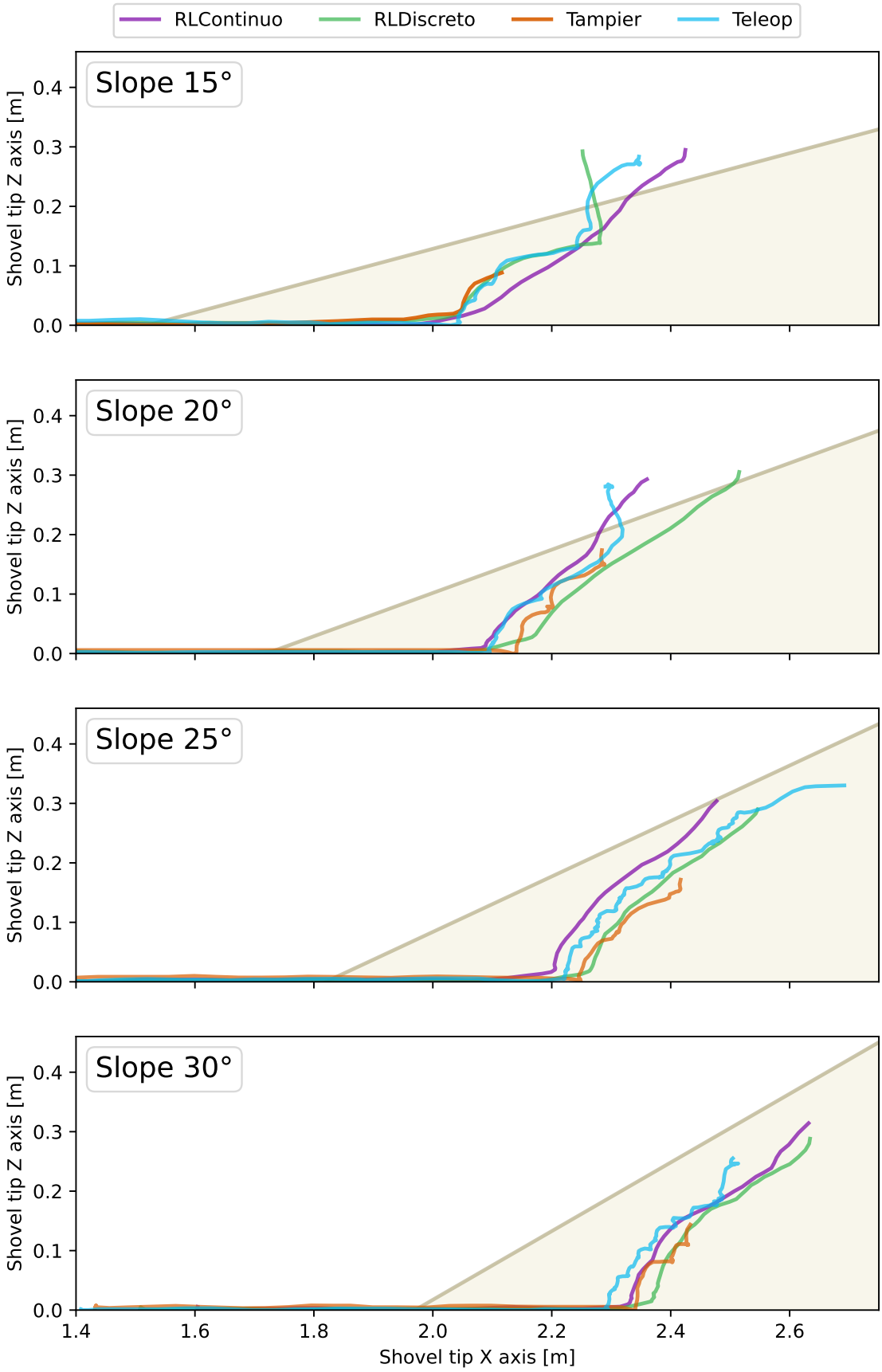

Figura 5.12: Trayectorias de mejores carguíos realizados en material homogéneo por cada controlador.

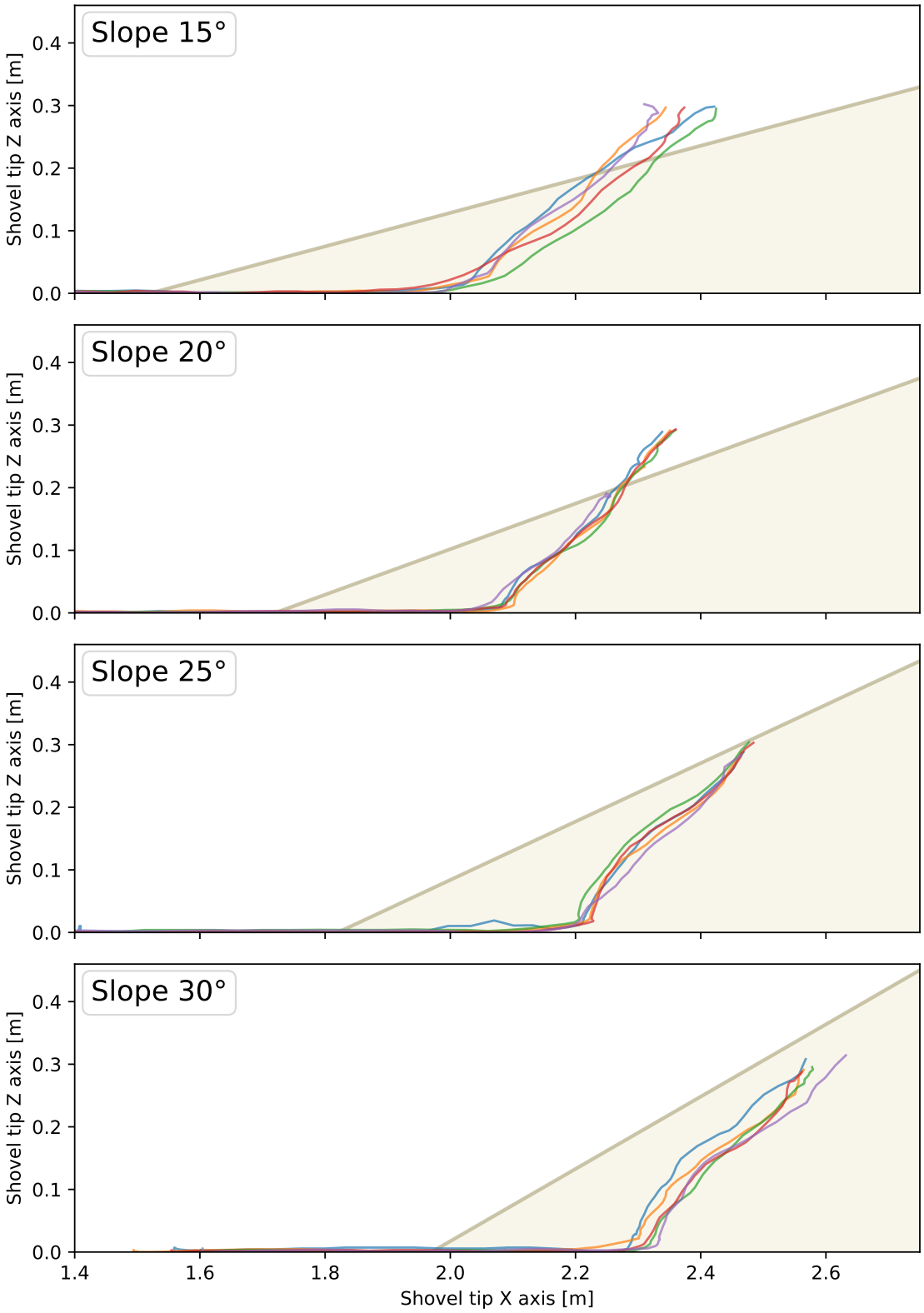

Figura 5.13: Trayectorias distintos carguíos realizados por política RLContinua en material homogéneo.

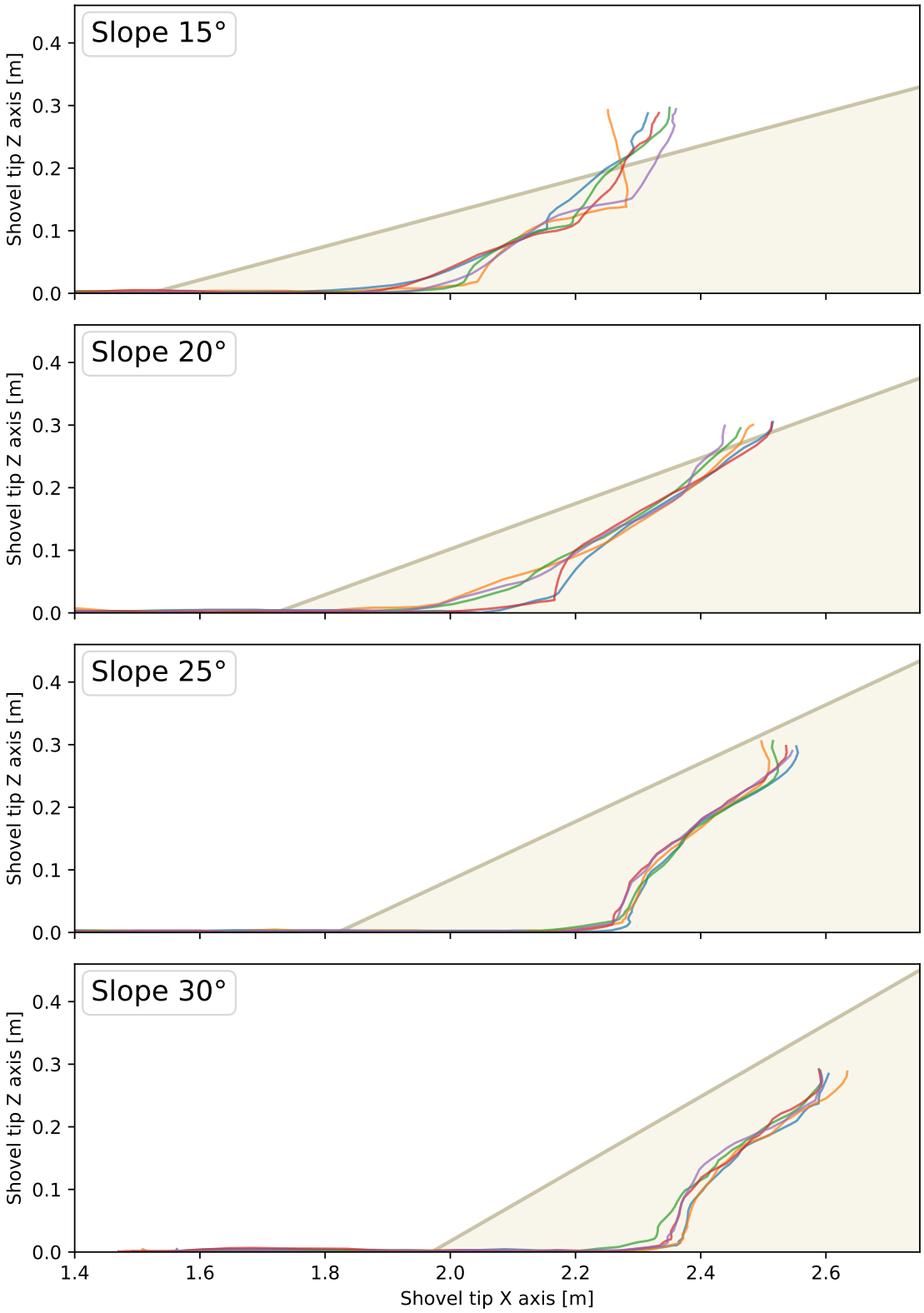

Figura 5.14: Trayectorias distintos carguíos realizados por política RLDiscreta en material homogéneo.

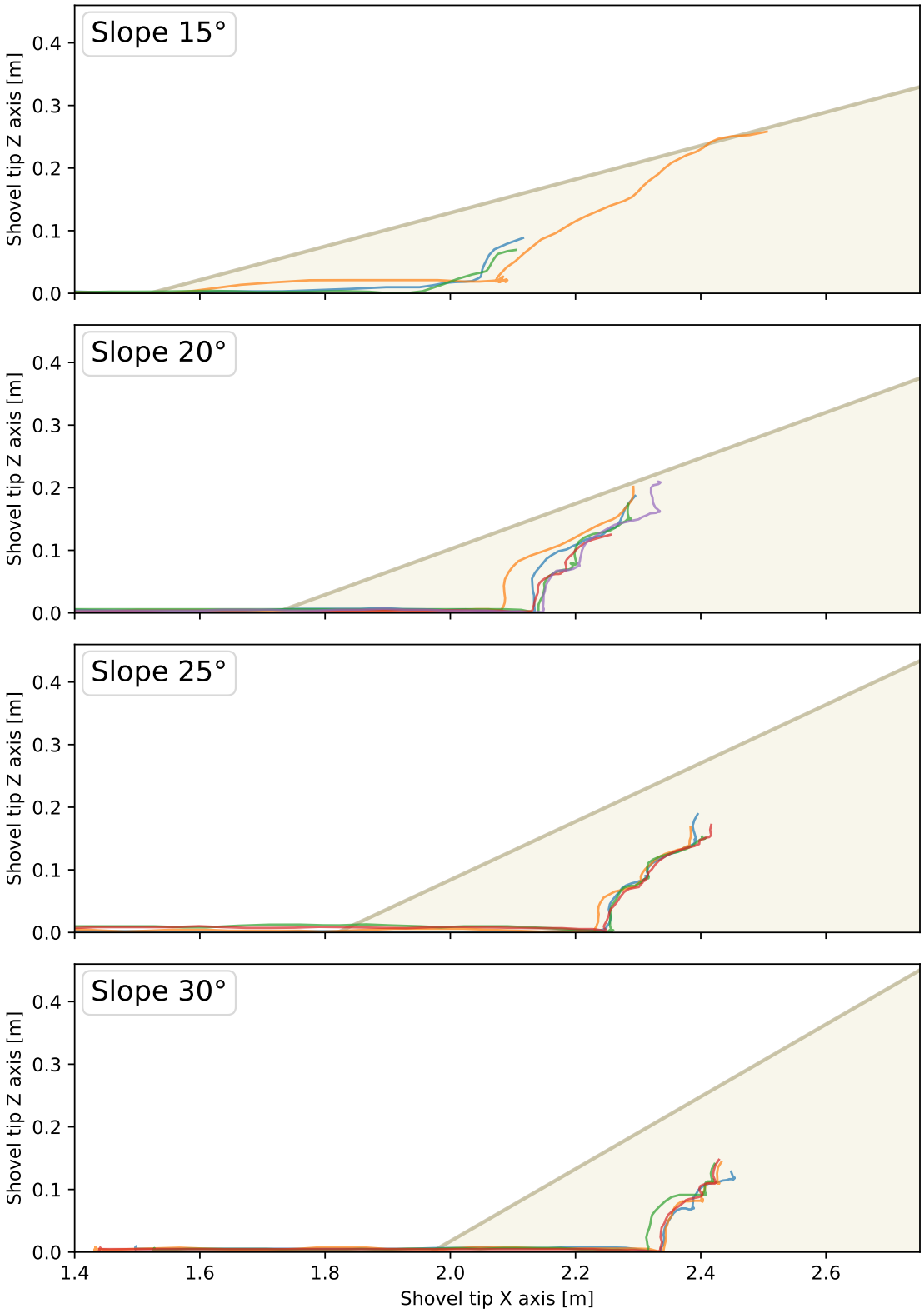

Figura 5.15: Trayectorias distintos carguíos realizados por algoritmo Tampier en material homogéneo.

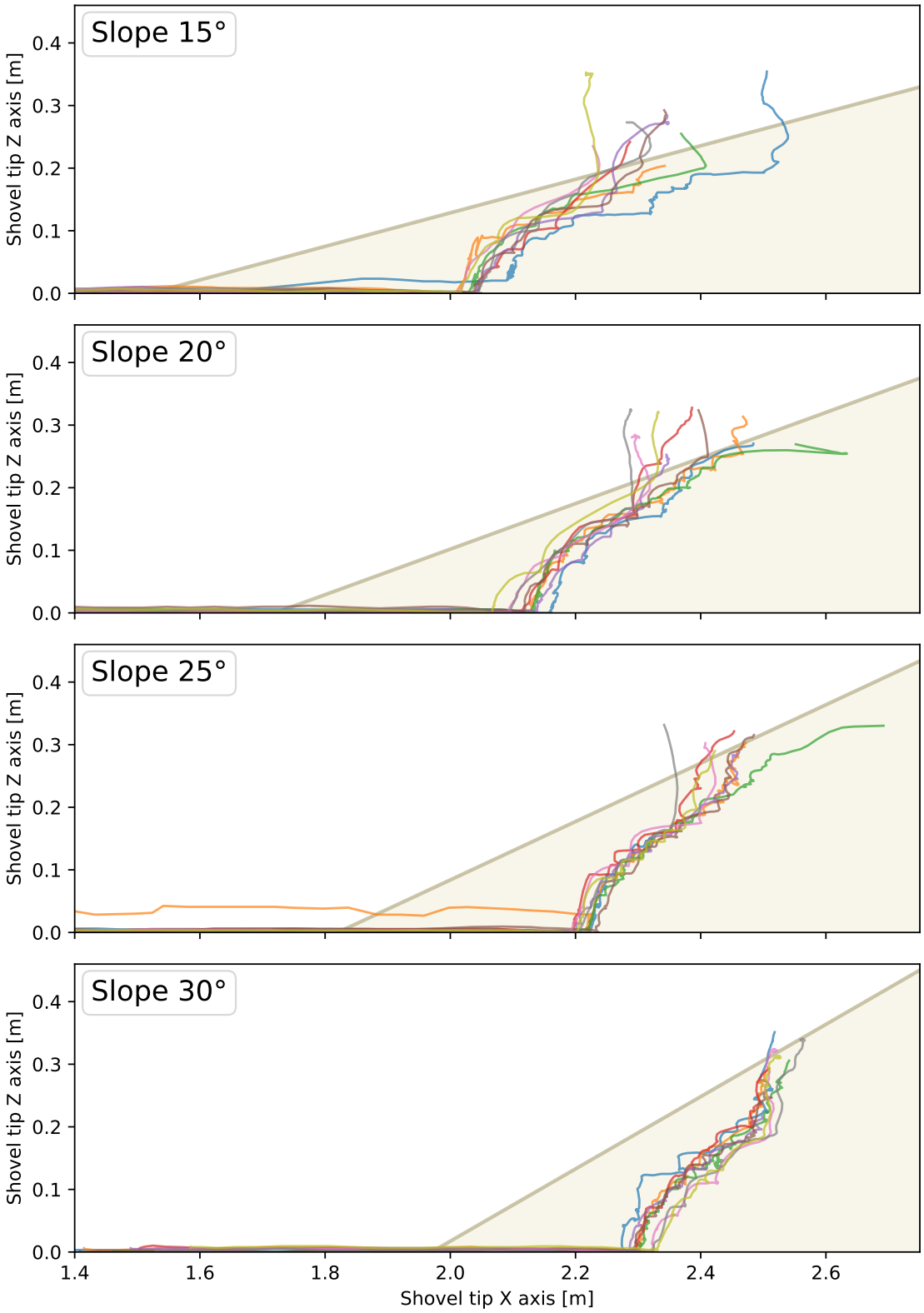

Figura 5.16: Trayectorias distintos carguíos realizados por agente Teleop en material homogéneo.

5.2.3. Excavación en material no homogéneo

De forma similar a la sección anterior, a continuación se presentan los resultados de los carguíos realizados en la pila de material no homogéneo. En primer lugar, la Tabla 5.10 presenta la tasa de éxito por pendiente para todos los controladores evaluados. En este caso, la política RLContinua falló en carguíos realizados en una pila de material con una pendiente de 25 y 30 grados, mientras que el algoritmo Tampier falló en carguíos realizados en una pila de material con una pendiente de 20, 25, y 30 grados. Los fallos de la política RLContinua fueron por agotar el límite de tiempo y cargar menos de 20 kg de material, puesto que la máquina se queda atascada sin levantar el balde.

Tabla 5.10: Tasa de éxito por pendiente en material no homogéneo.

Pendiente [deg]	RLContinuo	RLDiscreto	Tampier	Teleop
15	1	1	1	1
20	1	1	0.8	1
25	0.8	1	0.8	1
30	0.8	1	0.8	1

La Tabla 5.11 muestra el resumen de los carguíos realizados en la pila de material no homogéneo y considera solo los carguíos que fueron exitosos.

Tabla 5.11: Resultados de carguíos realizados en material no homogéneo para todos los controladores.

	S	RLContinuo	RLDiscreto	Tampier	Teleop
Material cargado [kg]	15	24.54 ± 0.63	25.69 ± 0.79	25.54 ± 1.43	25.36 ± 1.64
	20	23.99 ± 1.15	23.74 ± 0.77	21.34 ± 0.8	24.36 ± 1.15
	25	21.92 ± 1.84	22.53 ± 1.07	22.28 ± 0.97	22.22 ± 0.98
	30	25.61 ± 2.48	24.01 ± 2.77	23.22 ± 0.76	25.39 ± 1.38
	$\bar{X}$	$\mathbf{24.02 \pm 1.34}$	$\mathbf{23.99 \pm 1.13}$	$\mathbf{23.09 \pm 1.56}$	$\mathbf{24.33 \pm 1.29}$
Tiempo [s]	15	4.10 ± 0.5	5.21 ± 0.66	3.16 ± 0.31	7.23 ± 2.44
	20	4.18 ± 0.26	3.72 ± 0.34	3.72 ± 0.81	6.93 ± 2.41
	25	3.84 ± 0.25	3.27 ± 0.47	3.59 ± 0.15	8.88 ± 2.16
	30	5.30 ± 1.10	4.68 ± 0.64	3.97 ± 0.79	7.69 ± 1.18
	$\bar{X}$	$\mathbf{4.35 \pm 0.56}$	$\mathbf{4.22 \pm 0.77}$	$\mathbf{3.61 \pm 0.29}$	$\mathbf{7.68 \pm 0.74}$
Drift delantero[%]	15	39.29 ± 8.84	18.32 ± 7.69	73.11 ± 14.39	14.47 ± 9.87
	20	50.31 ± 17.18	23.30 ± 6.35	77.71 ± 13.74	13.27 ± 8.21
	25	59.55 ± 27.72	21.96 ± 11.65	75.80 ± 13.50	22.53 ± 18.17
	30	36.74 ± 14.01	12.77 ± 4.54	58.86 ± 15.33	18.20 ± 8.38
	$\bar{X}$	$\mathbf{46.47 \pm 9.11}$	$\mathbf{19.09 \pm 4.08}$	$\mathbf{71.37 \pm 7.41}$	$\mathbf{17.12 \pm 3.62}$
Drift trasero [%]	15	47.56 ± 14.41	53.79 ± 12.70	64.77 ± 39.30	52.68 ± 12.95
	20	74.07 ± 8.51	61.44 ± 15.39	92.10 ± 4.91	64.40 ± 11.64
	25	89.93 ± 4.18	61.96 ± 7.43	91.13 ± 3.55	65.64 ± 5.12
	30	94.13 ± 6.18	80.94 ± 4.81	91.39 ± 5.99	59.34 ± 16.13
	$\bar{X}$	$\mathbf{76.42 \pm 18.27}$	$\mathbf{64.53 \pm 10.01}$	$\mathbf{84.85 \pm 11.60}$	$\mathbf{60.52 \pm 5.10}$

En este caso, se puede observar en la Tabla 5.11 que la política RLContinua alcanza una mayor

cantidad de material cargado que la política RLDiscreta, pero es una diferencia del 0.1 %. Por otro lado, es el agente Teleop aquel que logra los mejores resultados, con una diferencia del 1.4 % por sobre ambas políticas RL. El algoritmo de Tampier es aquel que queda en último lugar en términos de material cargado. Sin embargo, con respecto al tiempo de carguío, se tiene que el algoritmo de Tampier es aquel que logra el mejor tiempo de carguío y el agente Teleop es aquel que toma más tiempo en cargar material. Con respecto al resbalamiento de las ruedas, se mantiene el mismo resultado que en el material homogéneo, en donde el agente Teleop y la política RLDiscreta son aquellos controladores que logran los porcentajes más bajos de resbalamiento. De forma general, estos resultados indican que ambas políticas aprendidas son capaces de realizar carguíos exitosos en pilas con material no homogéneo, con un desempeño similar o mejor que el algoritmo de Tampier y el agente Teleop. La Figura 5.17 muestra el detalle de los carguíos realizados en la pila de material no homogéneo, mostrando cuanto cargó y demoró cada maniobra realizada. En la Figura 5.17 se puede observar lo descrito anteriormente, en donde el agente Teleop es aquel que demora más tiempo en realizar la mayoría de sus carguíos.

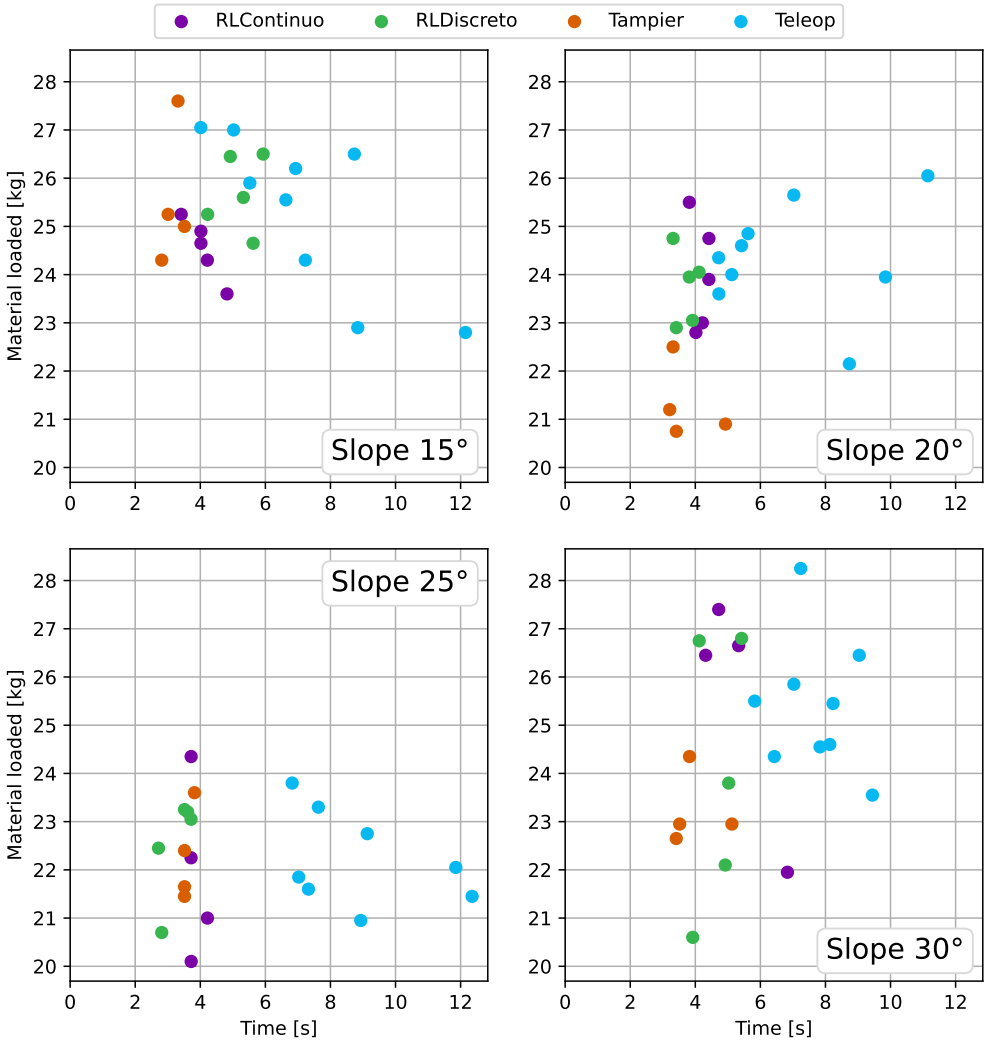

Figura 5.17: Visualización de resultados de carguíos realizados en material no homogéneo para todos los controladores.

Para complementar, se muestra en la Figura 5.18 el detalle de los carguíos realizados por ca-

da teleoperador, en donde se puede observar que una gran diferencia entre los tiempos de carguío,variando entre los 4 y 12 segundos, y la cantidad de material cargada, variando entre los 21 y 29 kilogramos.

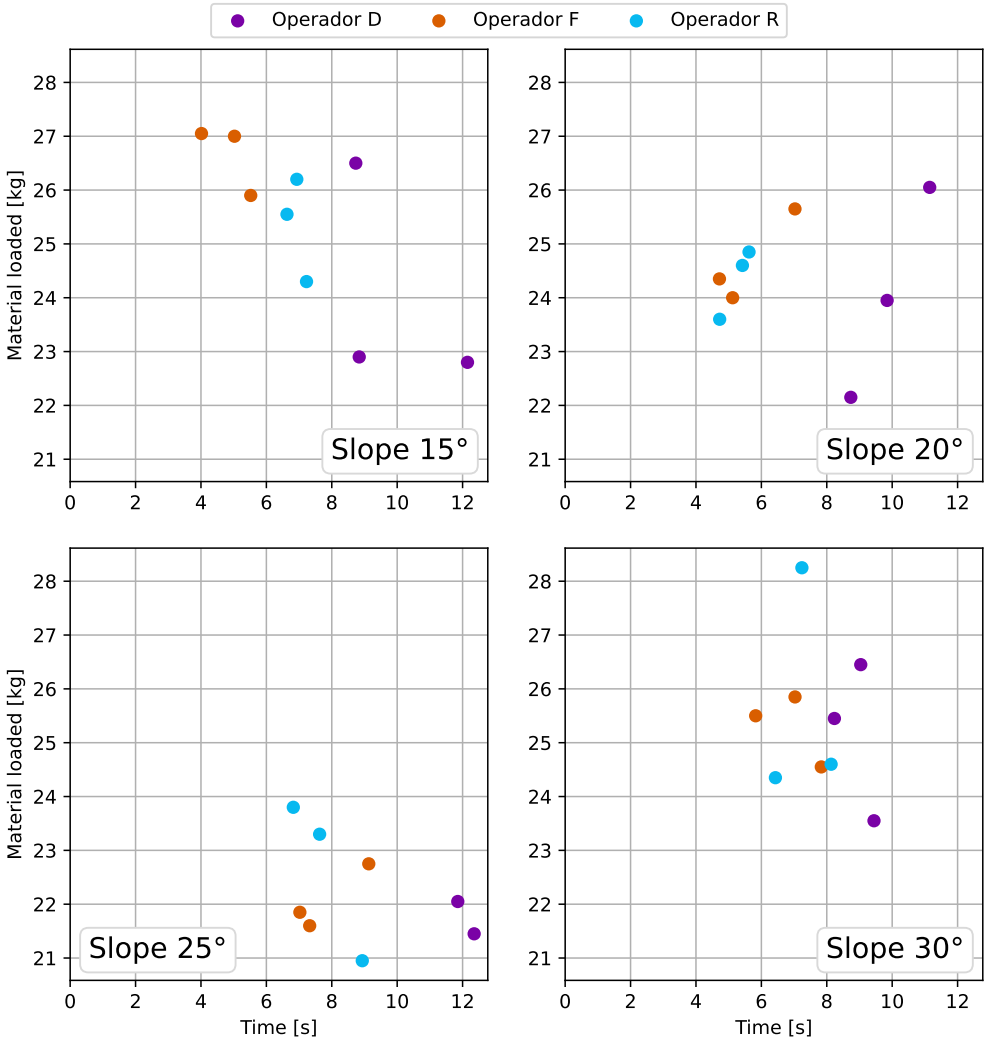

Figura 5.18: Visualización de resultados de carguíos realizados en material no homogéneo para distintos teleoperadores.

La Figura 5.19 muestras las acciones que ejecutaron los distintos controladores durante un carguío. En primera instancia, al comparar con las acciones ejecutadas en material homogéneo (Figura 5.11), las políticas RL ambas activan y desactivan con mayor frecuencia el actuador del *bucket*. En este caso, la política RLDiscreta frena la máquina al terminar el carguío, mientras que la política RLContinua mantiene la aceleración al máximo.

De los resultados obtenidos en la pila de material no homogéneo y comparando con los carguíos realizados en la pila de material homogéneo, el desempeño de los distintos controladores no fue afectado significativamente. La mayor baja en la cantidad de material cargado es del 4 % y ocurrió para la política RLContinua. Con respecto al tiempo, el algoritmo Tampier y el agente Teleop ambos mejoraron su desempeño, mientras que los tiempos de las políticas RL se mantuvieron similares. El agente Teleop logra mejorar sus tiempos principalmente debido a que los operadores realizaron

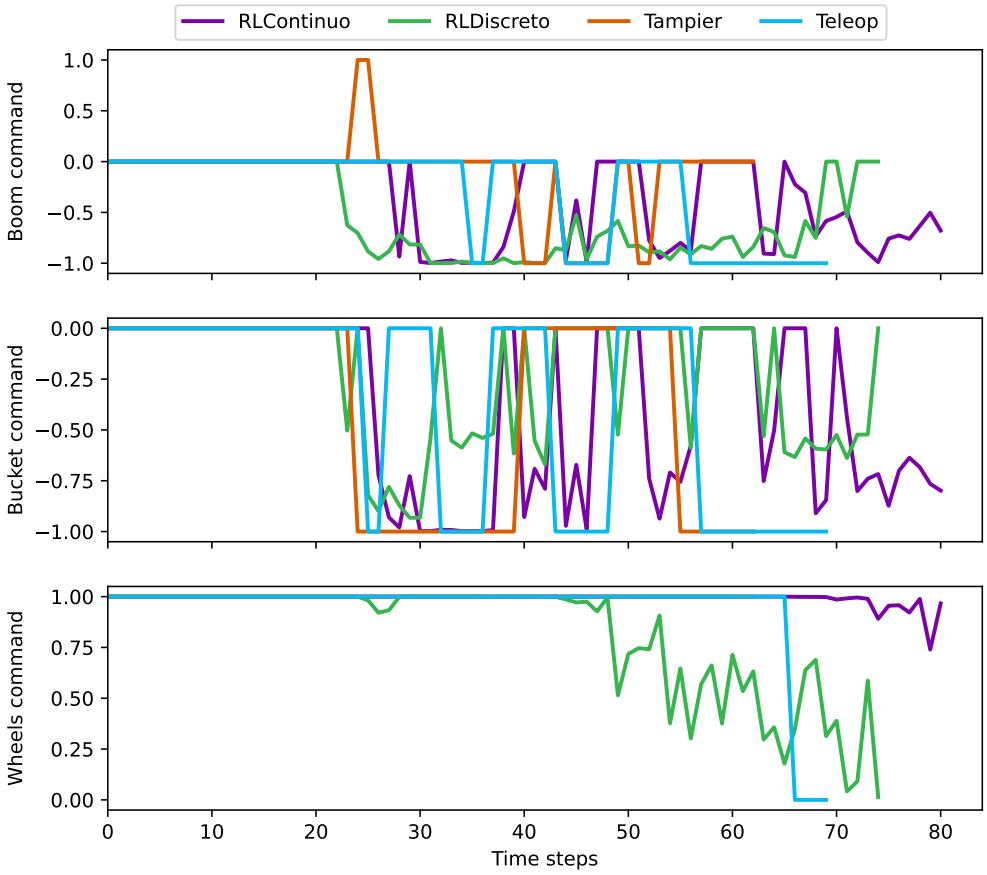

Figura 5.19: Ejemplo acciones de control durante un carguío en material no homogéneo.

los carguíos en el material no homogéneo después de los carguíos en el material homogéneo, por lo que tenían mayor experiencia con el manejo de la máquina.

A continuación, se presenta en primer lugar en la Figura 5.20 las trayectorias de los mejores carguíos por controladores para todas las pendientes. Similar a la Sección 5.2.2, las trayectorias que siguen los distintos operadores son semejantes entre ellas. Cuando la pendiente es de 30 grados, se puede observar justo un caso en que la política RLContinua actúa similar al algoritmo Tampier mientras que la política RLDiscreta actúa similar al agente Teleop.

En las Figuras 5.21, 5.22, 5.23 y 5.24, que corresponden a todas las trayectorias que siguieron los carguíos de los controladores en la pila de material heterogéneo, se puede observar que en todos los casos las trayectorias no varían entre ellas dentron de una misma pendiente. Se puede destacar que para las trayectorias de las políticas RLContinua y RLDiscreta, que estas trayectorias ya no presentan la misma fluidez que en el caso de la pila con material homogéneo. Esto se puede observar especialmente en la pila con pendiente de 30 grados, en donde estas tienen el comportamiento "escalonado" descrito en la Sección 5.2.2.

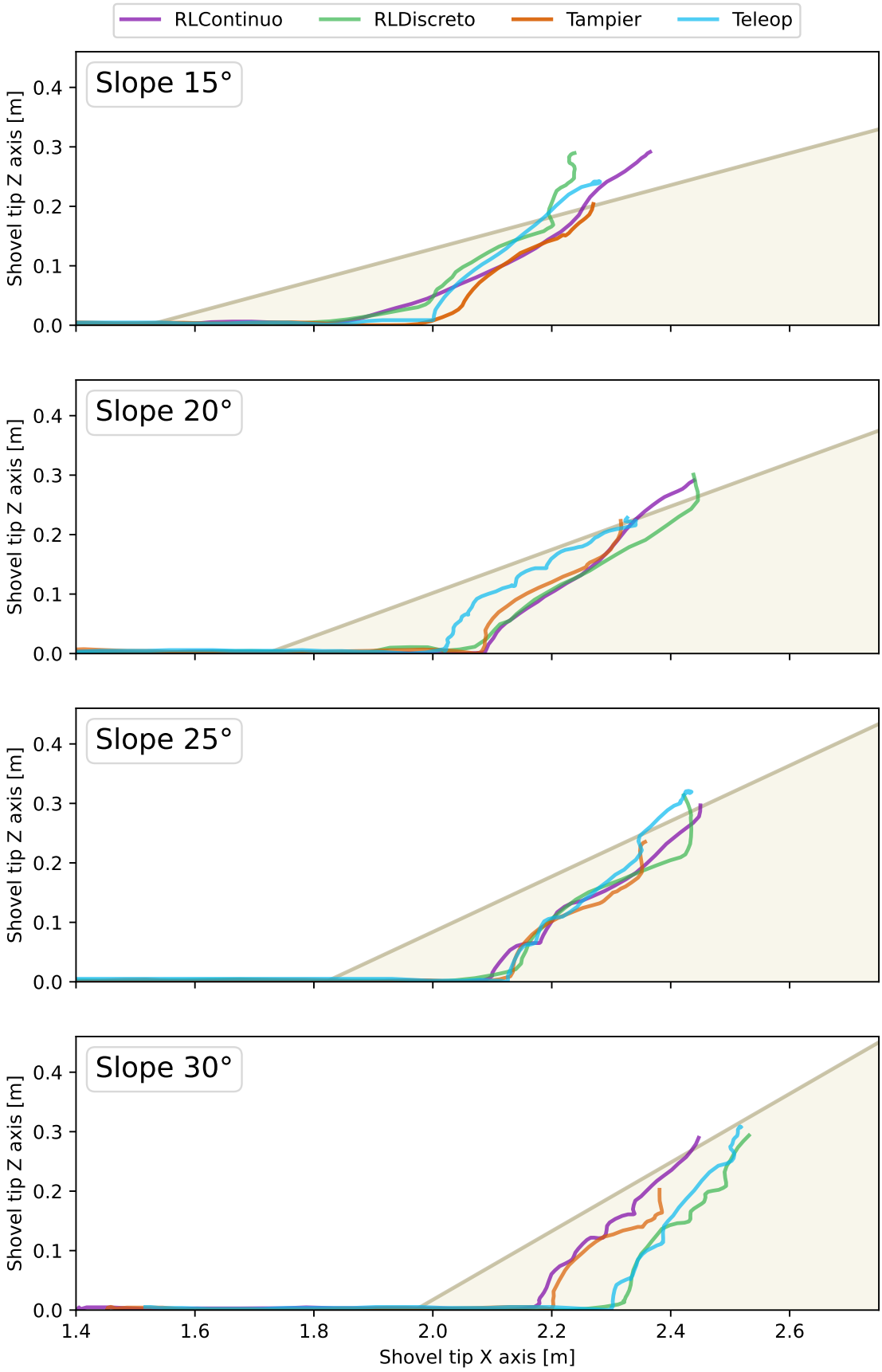

Figura 5.20: Trayectorias de mejores carguíos realizados en material no homogéneo por cada controlador.

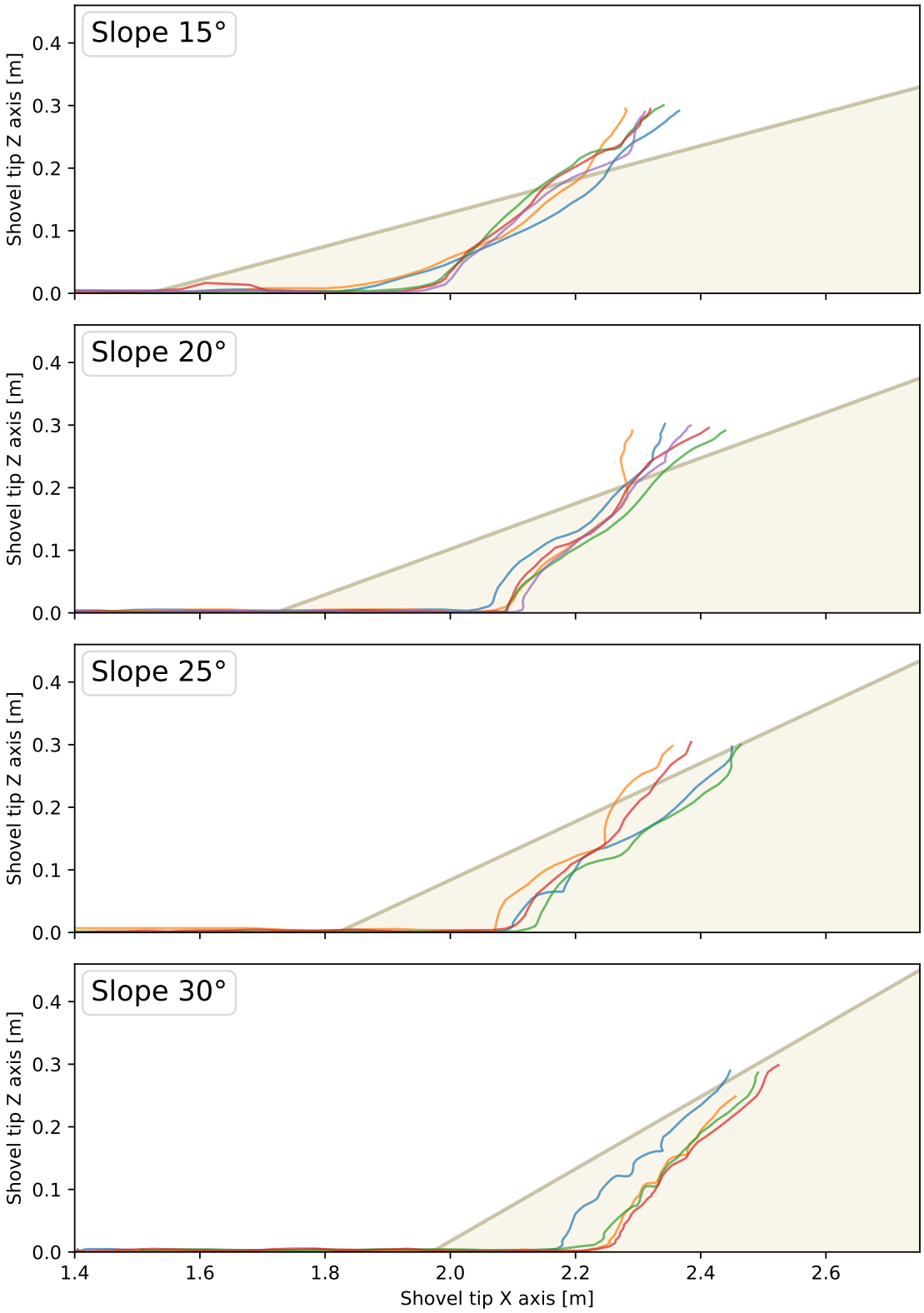

Figura 5.21: Trayectorias distintos carguíos realizados por política RLContinua en material no homogéneo.

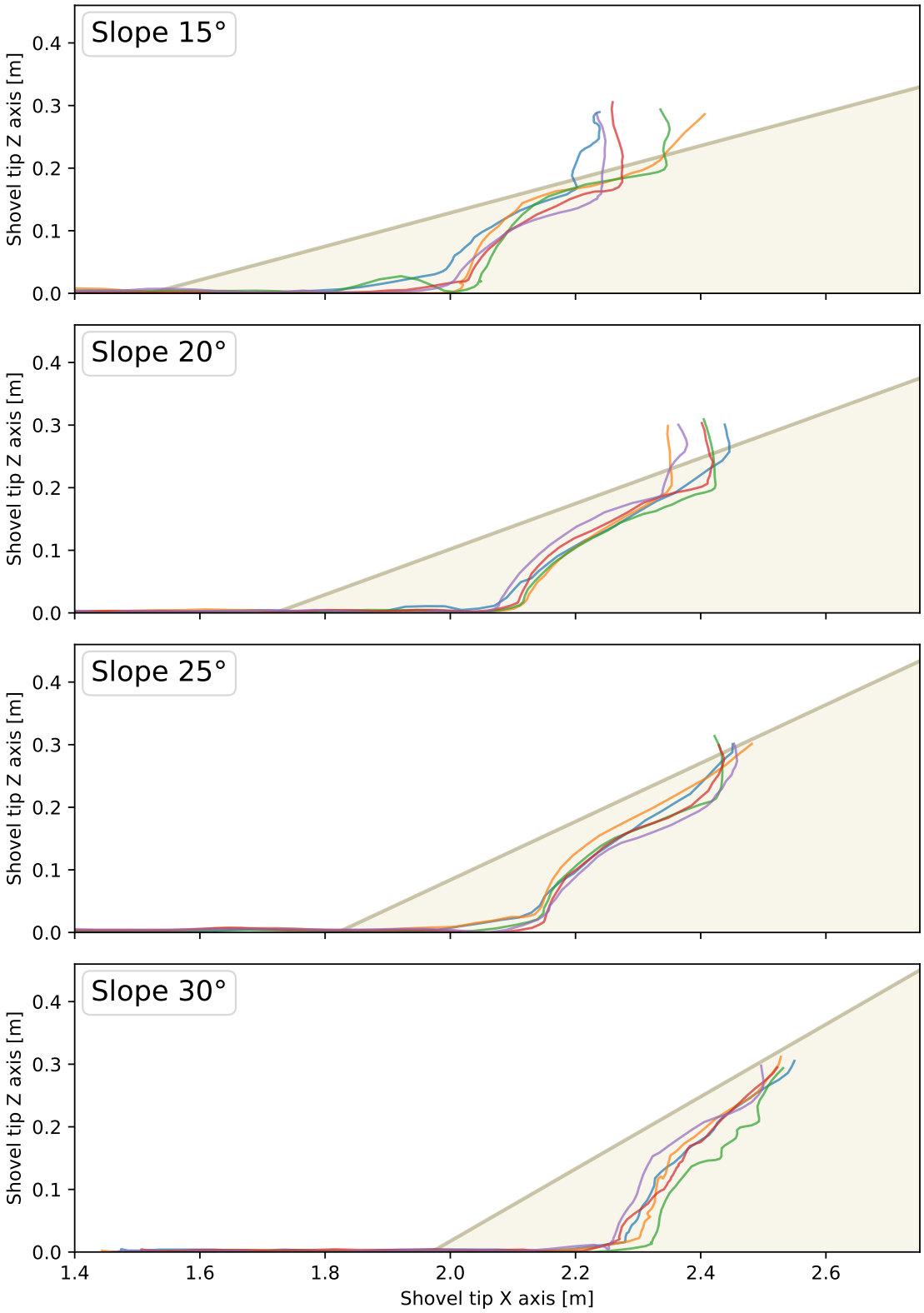

Figura 5.22: Trayectorias distintos carguíos realizados por política RLDiscreta en material no homogéneo.

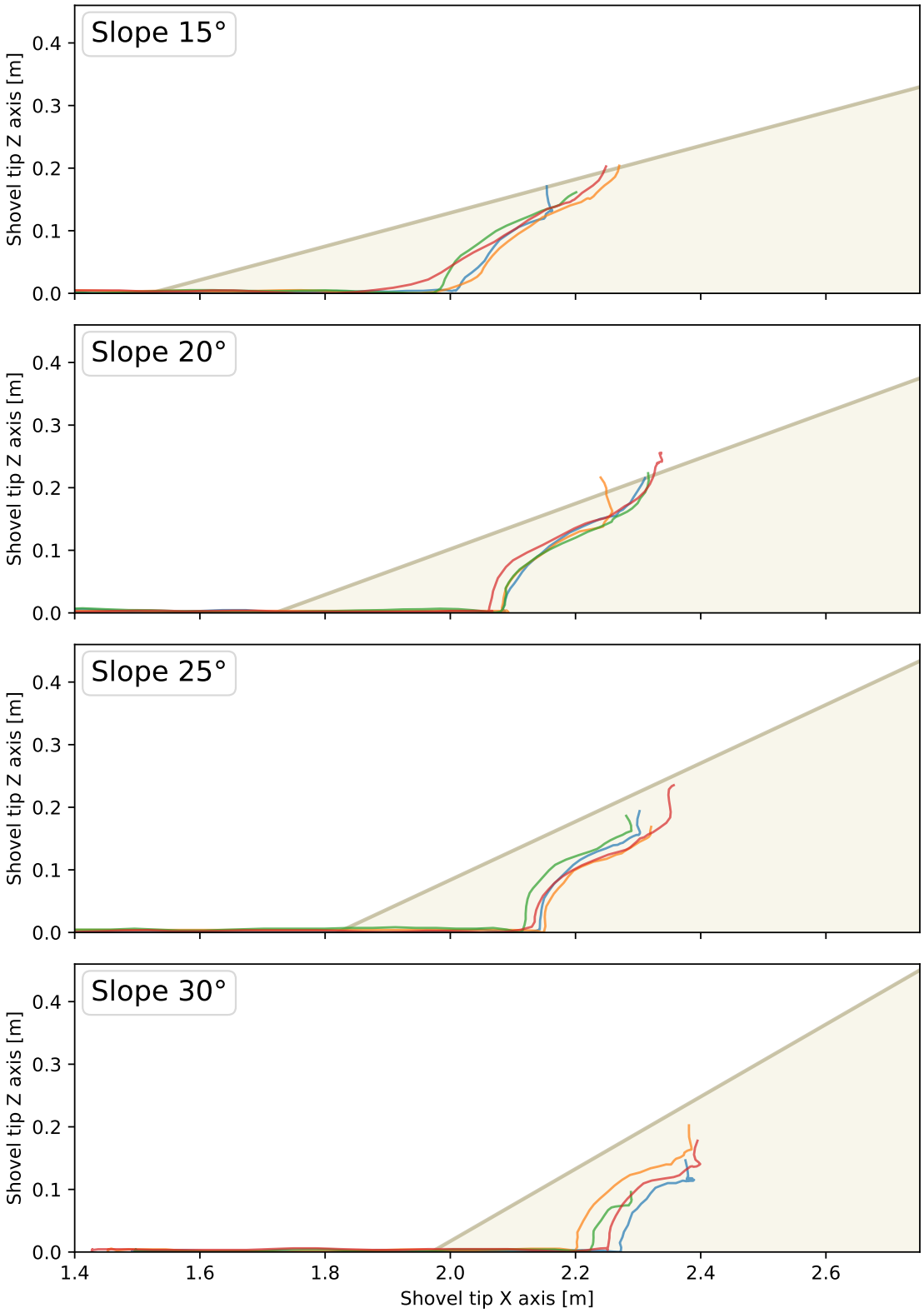

Figura 5.23: Trayectorias distintos carguíos realizados por algoritmo Tampier en material no homogéneo.

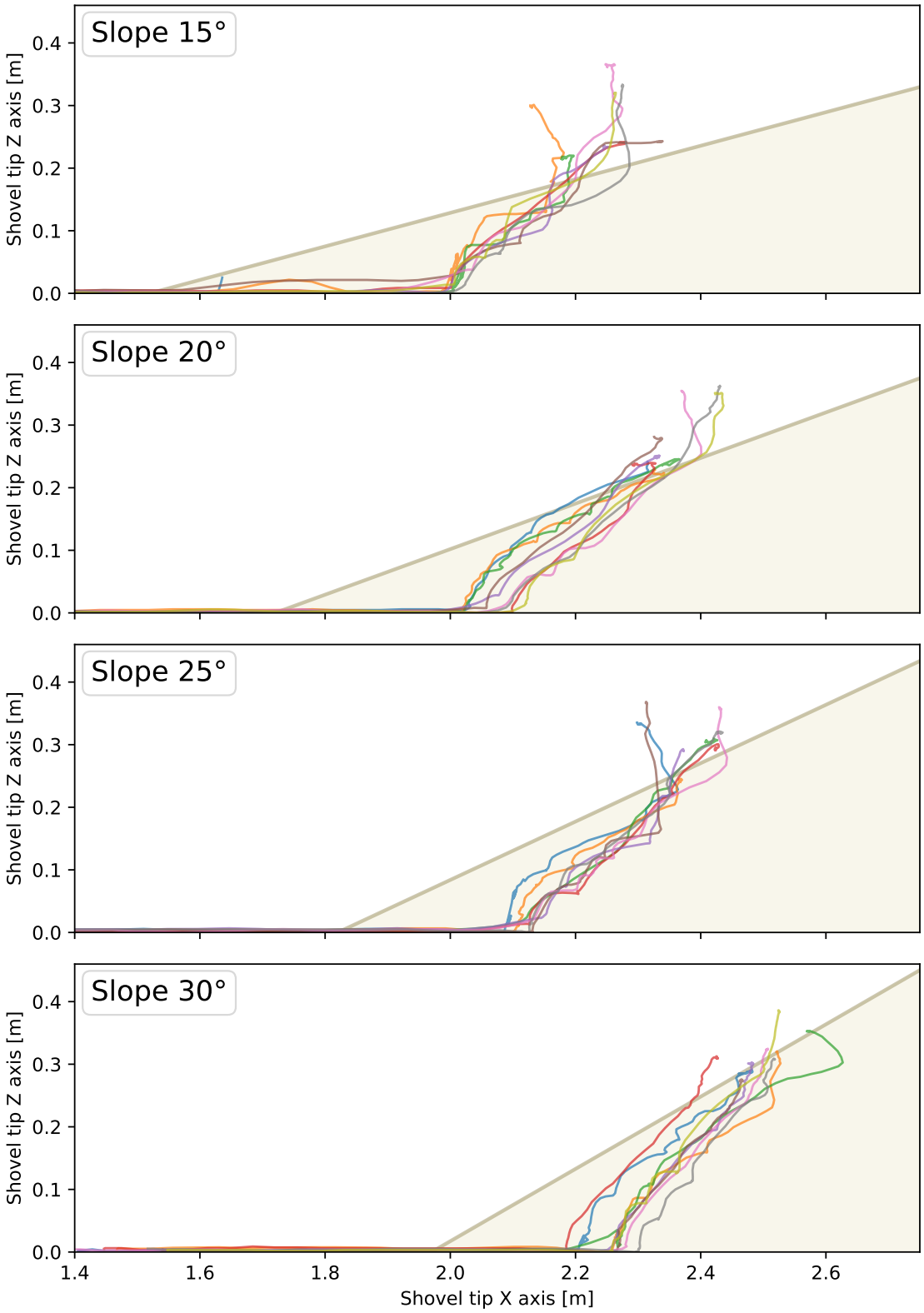

Figura 5.24: Trayectorias distintos carguíos realizados por agente Teleop en material no homogéneo.

5.2.4. Carguíos con errores en observación de pendiente

La política es entrenada con dos observaciones que entregan información directa sobre la configuración del ambiente: la pendiente de la pila α y la distancia entre la punta del balde y el límite de profundidad d_{end}. La observación d_{end} está directamente relacionada con la observación α debido a que la zona final se construye considerando la pendiente de la pila de material. Debido a esto, es importante que la política aprendida sea robusta frente a errores en la observación de la pendiente de la pila al momento de cargar.

Para evaluar cuán robusta es la política aprendida frente a estos errores de observación, se realizan distintos carguíos en donde la observación de la pendiente contiene errores artificiales de medición. Para simular los errores de medición, se realizan carguíos en pilas de material con una pendiente de 15 y 30 grados, pero la observación de la pendiente es cambiada por el valor de otra pendiente. El error en la observación varía entre los 5 grados y los 15 grados. Estos experimentos son realizado para ambas políticas RLContinua y RLDiscreta.

La Tabla 5.13 muestra los resultados obtenidos de los carguíos realizados en una pila de material homogéneo y con una pendiente de 30 grados. La columna de "Error" indica la diferencia entre la observación que recibe el agente y la pendiente real de la pila. En el caso de la Tabla 5.13, en donde la pendiente es de 30 grados, las observaciones varían entre los 15 (error de -15 grados) y 30 grados (error de 0 grados).

Tabla 5.12: Resultados de carguíos realizados por todos los controladores en pila de material homogéneo con pendiente de 30 grados, con error en observación de pendiente.

	Error [deg]	RLContinuo	RLDiscreto
Material cargado [kg]	-15	23.86 ± 0.7	25.94 ± 0.8
	-10	27.41 ± 1.22	25.61 ± 1.61
	-5	26.48 ± 1.38	22.16 ± 5.42
	0	27.24 ± 0.79	28.12 ± 0.91
Tiempo [s]	-15	9.71 ± 2.22	9.93 ± 0.04
	-10	4.62 ± 1.25	9.9 ± 0.05
	-5	7.35 ± 6.56	8.28 ± 2.39
	0	3.98 ± 0.2	4.72 ± 0.36
Drift delantero[%]	-15	98.19 ± 1.36	8.42 ± 1.54
	-10	41.66 ± 14.15	26.99 ± 33.28
	-5	62.32 ± 28.7	31.07 ± 34.75
	0	94.96 ± 2.59	16.32 ± 3.84
Drift trasero [%]	-15	97.18 ± 1.05	96.6 ± 1.67
	-10	94.59 ± 1.86	95.99 ± 1.22
	-5	94.96 ± 2.59	92.33 ± 6.87
	0	89.24 ± 6.6	88.5 ± 6.1

Se puede observar de los resultados de la Tabla 5.12 que para ambas políticas, con respecto a la cantidad de material cargado, el error en la observación afecta negativamente y provoca que se cargue menos material: una baja del 12.4 % para la política RLContinua y una baja del 7.7 % para

la política RLDiscreta. Cuando el error es máximo, es decir que la pendiente real es de 30 grados pero la política recibe una observación de 15 grados, todos los carguíos terminan por timeout con la máquina atascada tratando de avanzar sin levantar el balde para terminar el episodio. Esto se puede observar en la Tabla 5.12, en donde los tiempos de carguío aumentan y el resbalamiento de las ruedas alcanza casi el máximo posible. Con respecto al tiempo de carguío, estos también se ven afectados negativamente, provocando que los carguíos sean terminados por timeout. Con respecto, al resbalamiento de las ruedas, se tiene que para la política RLContinua, para ambos ejes aumenta casi al máximo el porcentaje de resbalamiento, mientras que para la política RLDiscreta, el resbalamiento en el eje trasero aumenta casi al máximo mientras que el resbalamiento para el delantero disminuye. Esta disminución en el caso de la política RLDiscreta se debe a que la máquina se queda atascada con el balde de material lleno y tratando de avanzar, lo que genera la suficiente tracción para que el eje delantero no pueda girar pero con los motores tratando de avanzar. Es por estos casos que se implementó el timeout a la maniobra de carguío, para proteger los motores.

Las Figuras 5.25 y 5.26 muestran un ejemplo de las acciones ejecutadas durante un carguío por ambas políticas RL en la pila de material homogéneo con pendiente 30 grados, tanto para un error de observación nulo y con 15 grados de error.

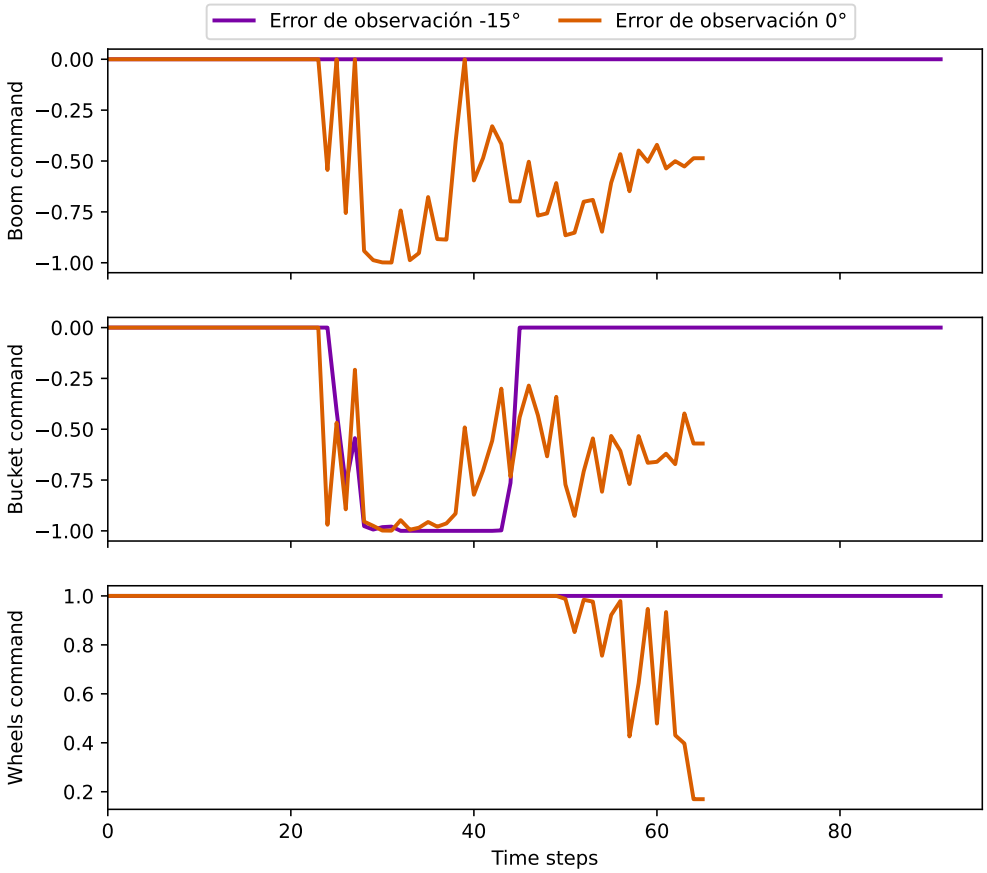

Figura 5.25: Acciones ejecutadas por la política RLContinua durante un carguío y con errores en la observación de pendiente, en pila de material homogéneo y con pendiente de 30 grados.

Se puede observar en la Figura 5.25 las diferencias entre un carguío normal y un carguío con error en la observación de pendiente ejecutados por la política RLContinua. En el caso en que hay

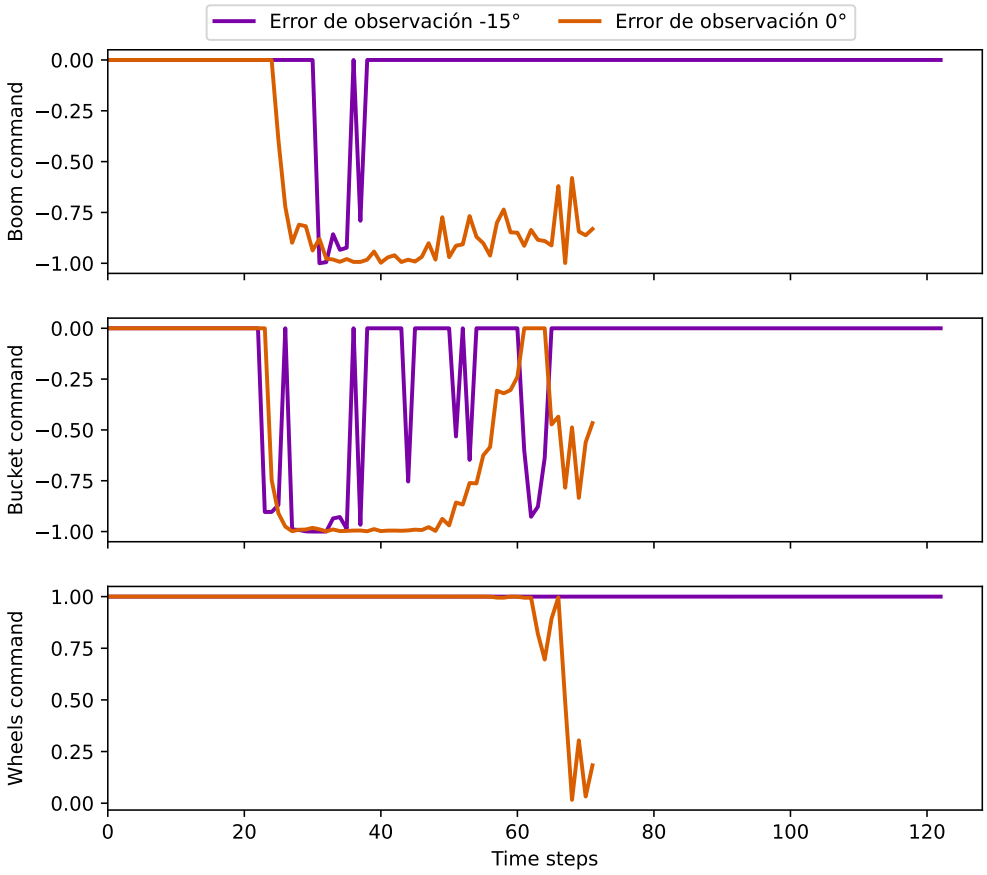

Figura 5.26: Acciones ejecutadas por la política RLDiscreta durante un carguío y con errores en la observación de pendiente, en pila de material homogéneo y con pendiente de 30 grados.

error en la observación de la pendiente, se tiene el caso donde el carguío fue terminado debido a que la política solo aceleraba contra la pila de material sin activar el brazo, lo que genera que se atasque y las ruedas resbalen. En la Figura 5.26 se tiene el mismo caso para la política RLDiscreta, en que no se termina la maniobra y se atasca la máquina, tratando de acelerar constantemente contra la pila de material.

Luego, la Tabla 5.13 muestra los resultados obtenidos de los carguíos realizados en una pila de material homogéneo y con una pendiente de 15 grados. En este caso, las observaciones varían entre los 15 (error de 0 grados) y los 30 grados (error de +15 grados).

En el caso de la política RLContinua, se puede observar en la Tabla 5.13 que si bien el tiempo de carguío y el resbalamiento de las ruedas en ambos ejes disminuyó al tener error en la observación de la pendiente, la cantidad de material cargado también disminuyó. En el caso de la política RLDiscreta, la cantidad de material y el resbalamientos de las ruedas en el eje trasero disminuyeron de forma similar que para el caso de la política RLContinua, mientras que el tiempo de carguío tuvo un leve variación y el resbalamiento en el eje delantero aumentó.

Las Figuras 5.27 y 5.28 muestran un ejemplo de las acciones ejecutadas durante un carguío por ambas políticas RL en la pila de material homogéneo con pendiente 15 grados, tanto para un error de observación nulo y con 15 grados de error.

Tabla 5.13: Resultados de carguíos realizados por todos los controladores en pila de material homogéneo con pendiente de 15 grados, con error en observación de pendiente.

	Error [deg]	RLContinuo	RLDiscreto
Material cargado [kg]	+15	20.76 ± 2.66	20.59 ± 0.85
	+10	23.13 ± 1.23	21.77 ± 0.45
	+5	23.77 ± 1.09	23.95 ± 0.82
	0	22.4 ± 2.36	23.28 ± 1.43
Tiempo [s]	+15	3.07 ± 0.3	4.38 ± 0.66
	+10	3.56 ± 0.22	4.02 ± 0.29
	+5	3.78 ± 0.27	4.1 ± 0.26
	0	4.06 ± 0.23	4.38 ± 0.5
Drift delantero[%]	+15	13.66 ± 6.95	41.63 ± 17.34
	+10	15.9 ± 11.7	19.08 ± 2.9
	+5	40.25 ± 18.14	7.82 ± 7.4
	0	47.99 ± 11.32	26.45 ± 5.75
Drift trasero [%]	+15	17.73 ± 16.87	41.63 ± 17.34
	+10	32.02 ± 9.62	24.76 ± 9.19
	+5	45.75 ± 4.94	38.34 ± 3.02
	0	35.12 ± 19.59	52.59 ± 4.55

Ambas Figuras 5.27 y 5.28 muestran que para ambas políticas RL, cuando se tiene error en la observación de pendiente, toda la maniobra de carguío demoró menos que cuando no tiene error en la observación. Se tiene además que las acciones tomadas por ambas políticas RL durante el carguío son similares entre el caso con error y sin error en la observación.

Los resultados obtenidos en esta sección permiten evaluar el impacto que tiene el error de observación de pendiente en el desempeño de los carguíos. Se tiene de los resultados que cuando la pila de material tiene una pendiente de 15 grados y se agrega error a la observación para que la política vea una mayor pendiente, si bien el la cantidad de material cargado disminuye todos los carguíos fueron terminados. Mientras que en el caso en que la pila de material tiene una pendiente de 30 grados y se agrega error a la observación para que la política vea una menor pendiente, los carguíos se ven afectados considerablemente, hasta que casi todos los carguíos terminan por timeout.

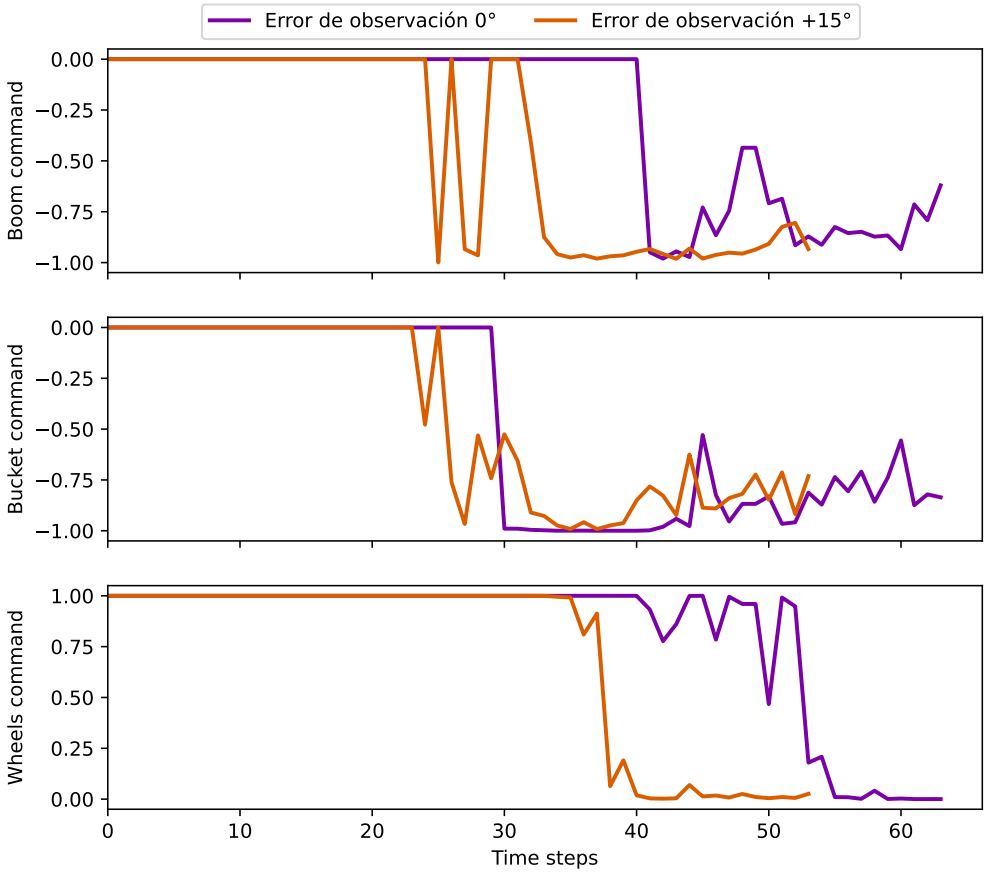

Figura 5.27: Acciones ejecutadas por la política RLContinua durante un carguío y con errores en la observación de pendiente, en pila de material homogéneo y con pendiente de 15 grados.

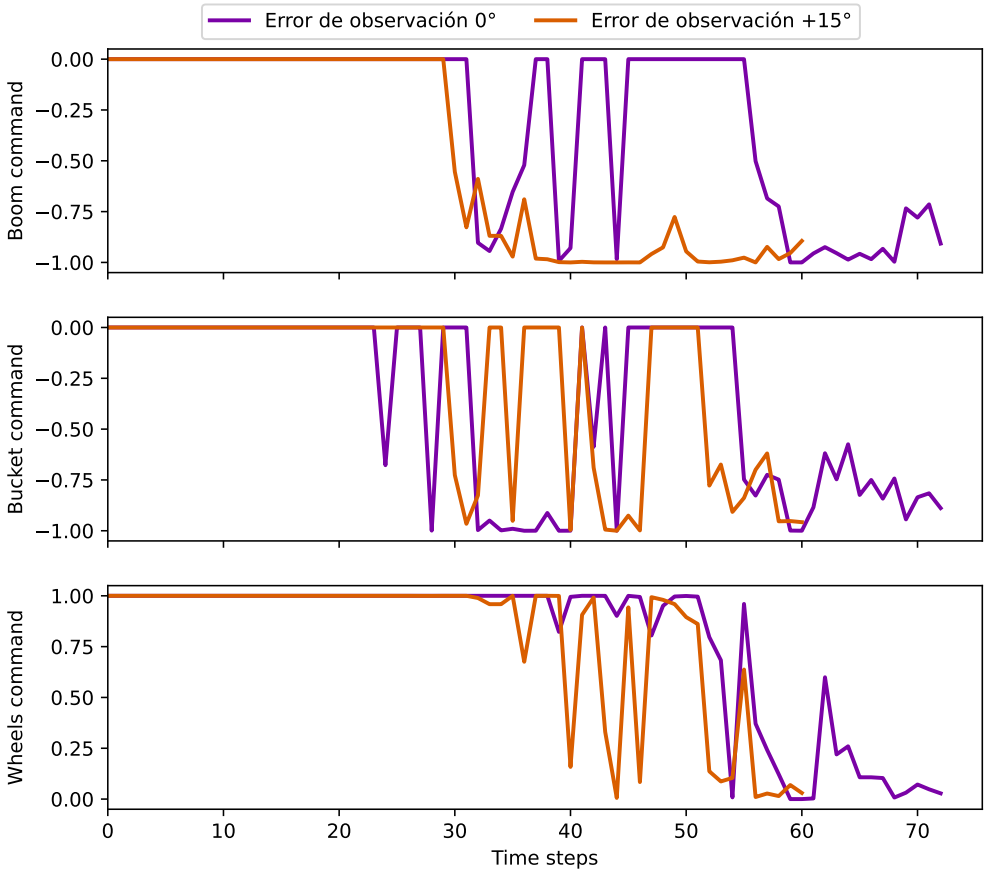

Figura 5.28: Acciones ejecutadas por la política RLDiscreta durante un carguío y con errores en la observación de pendiente, en pila de material homogéneo y con pendiente de 15 grados.

Capítulo 6

Discusión

En esta sección se presentan distintos puntos de discusión sobre el diseño final del agente RL y los resultados obtenidos en los experimentos descritos anteriormente.

6.1. Simulación del punto de extracción

En los puntos de extracción de la minería subterránea, si bien el tipo de material y sus características intrínsecas no cambian significativamente entre carguíos, la forma de la pila y su granulometría varían debido a la estrategia de extracción utilizada (generalmente por tronadura). En este trabajo, para simular el material se utiliza una extensión del modelo analítico FEE, puesto que al ser un modelo analítico permite una integración más simple y rápida del modelo. Debido a que las políticas RL son entrenadas mediante múltiples interacciones con el ambiente, en este caso alcanzando los 300000 pasos de tiempo para entrenar una política, la velocidad de la simulación es crucial para que los entrenamientos duren lo menor posible. De esta forma, mientras más rápida sea la simulación más rápido es el entrenamiento, lo que permite iterar mejor en el diseño y otros beneficios.

Si bien una de las principales ventajas de la simulación del material al utiliza la FEE extendida es la velocidad de la simulación, también tiene desventajas a tomar en cuenta. Una de las principales limitaciones que tiene la aplicación de este modelo es que no utiliza físicas de contacto entre el balde y el material. La fuerza total que la pila de material ejerce sobre el LHD es calculada únicamente en base a la pose de la punta del balde dentro de la pila. En consecuencia, existen situaciones en las que el fondo del balde está en contacto con material en la pila pero la punta está afuera por lo que no se ejerce fuerza sobre el balde. Debido a la exploración de pares estado-acción durante el entrenamiento del agente RL, es posible que algunas experiencias se vean afectadas por esto y por lo tanto son experiencias más alejadas de la realidad. Una opción para solucionar este problema es utilizar simuladores más potentes como por ejemplo el utilizado en [24] pero esto podría aumentar los costos computacionales del entrenamiento.

En este trabajo se implementó el modelo extendido de la FEE junto al sistema de *voxels* para simular la pila de material. El principal objetivo del sistema de *voxels* es poder variar la densidad del material por secciones de la pila y que así el ambiente de entrenamiento entregue escenarios

más realistas. Estos escenarios más realistas se enfocan en los casos de material tronado, en donde se tiene una gran variedad de tamaño de rocas. Las distintas densidades de material en cada *voxel* buscan simular zonas en la pila en donde la punta del balde puede pasar con distintos niveles de dificultad. Los resultados obtenidos en la Sección 5 indican que la simulación del material permite entrenar políticas RL para realizar carguíos autónomos. Tanto en la pila de material homogéneo como en la pila de material no homogéneo, las políticas RL alcanzaron mejores resultados que el algoritmo Tampier y la teleoperación de la máquina. Esto indica que la simulación del ambiente puede ser utilizada para entrenar políticas RL para carguío autónomo.

6.2. Modelamiento del problema de aprendizaje reforzado

Con respecto a las recompensas, su diseño es una parte fundamental para el entrenamiento de una política con aprendizaje reforzado. En este trabajo de tesis, el diseño de las recompensas tiene como enfoque permitir explorar un amplio espectro de trayectorias posibles introduciendo la menor cantidad de conocimiento experto. La recompensa por trayectorias (definida en 4.2.4) ejemplifica este enfoque abordado, al limitarse a encerrar las trayectorias posibles que puede seguir la punta del balde sin entregar una trayectoria guía. No obstante, para lograr que la política lograra ejecutar los carguíos de forma exitosa, se agregaron recompensas que entregan directamente a la política información: la recompensa por atascamiento, la recompensa por inactividad y la recompensa por incentivo a enterrar el balde. Estas son consideradas recompensas complementarias, puesto que fueron agregadas con el único objetivo de corregir que la política aprendía a quedarse quieta dentro de la pila para minimizar la recompensa. Previo al diseño final del agente presentado en este trabajo, se entrenaron múltiples agentes sin la inclusión de las recompensas complementarias, pero las políticas aprendidas no realizaban carguíos exitosos. Considerando esto, el diseño de la recompensa tiene espacio para iterar y mejorar, con el fin de eliminar las recompensas complementarias.

Luego, la simulación de la interacción entre el balde y el material juega un rol importante en el diseño de las recompensas utilizadas. En primer lugar, la recompensa por enterrar el fondo del balde existe únicamente debido a la falta de físicas de contacto para el balde completo. Luego, la recompensa por resbalamiento también depende de cuán cercano a la realidad es la simulación, ya que determina la fuerza que genera la pila de material y así el esfuerzo que deben ejecutar las ruedas. Debido a esto, la recompensa por resbalamiento es diseñada con conocimiento experto para facilitar el aprendizaje de esta interacción entre ruedas y fuerza ejercida. El conocimiento experto corresponde a la indicación explícita de levantar el brazo para reducir el resbalamiento.

Considerando todo lo anterior, en el diseño final modelamiento del problema propuesto, balancear las recompensas que incentivan al agente a terminar rápidamente el episodio y las recompensas relativas al material cargado es crítico para evitar que el agente fracase al realizar carguíos. El mejorar la simulación del material permitiría modificar las recompensas complementarias y así alcanzar un diseño más estable que pueda cargar material.

Con respecto a las observaciones, una observación utilizada en otros trabajos de automatización de maquinaria minera que no fue incorporada en este trabajo es la presión de los cilindros hidráulicos que controlan el brazo. Esta observación permitiría entregar al agente información inmediata sobre la resistencia que está ejerciendo la pila de material sobre el balde. No obstante, las observaciones de las velocidades de las distintas articulaciones y la observación de la velocidad de la máquina también entregan información inmediata sobre la resistencia que ejerce la pila de

material.

Luego, la zona de observación y sus observaciones también hacen parte del diseño del agente RL. Considerando que el agente solo ejecuta la etapa excavación de material, la política aprendida debe ser incorporada en un sistema completo que pueda ejecutar carguíos y depositar el material en un lugar objetivo, tal como el sistema completo propuesto en [61]. En primer lugar, para facilitar la implementación del sistema, se busca utilizar observaciones que pueden ser adquiridas directamente de la máquina, como por ejemplo la velocidad de la máquina y el estado de sus articulaciones. Luego, la zona de observación también toma en consideración la implementación en el mundo real. La posición de la zona de observación solo depende de donde inicia la pila de material y la dimensión de la zona es constante para todas las pilas, por lo que para realizar una excavación, se debe tener algún módulo que ubique la zona de observación al detectar el inicio de la pila y así obtener las observaciones respectivas para el agente. Se necesita también un módulo que entregue la pendiente de la pila de material, debido a que es una de las observaciones de la política RL que está relacionada con el ambiente y que, como se pudo observar en la Sección 5.2.4, los errores en esta observación pueden generar que los carguíos no sean exitosos. Estas consideraciones son importantes para que la política entrenada pueda ser implementada en un LHD real que ejecute carguíos en ambientes reales.

6.3. Políticas de carguío aprendidas

Los distintos experimentos realizados tanto en simulación como en la realidad permiten evaluar distintos aspectos sobre las políticas RL aprendidas. Para comparar el desempeño de las políticas RL entrenadas en este trabajo, se considera la cantidad de material cargado, el tiempo de carguío y el resbalamiento de las ruedas como métricas principales.

En primer lugar, en el ambiente de simulación y como muestran los resultados de la Sección 5.1.3, la política RLDiscreta logra un mejor desempeño que la política RLContinua en todas las métricas evaluadas. Un factor que aporta a esto es que el agente RLDiscreto solo puede mover las articulaciones a su máxima velocidad, por lo que realiza los carguíos más rápidos y aquellas recompensas que penalizan por no mover las articulaciones son menos propensas a activarse. Luego, en el mundo real tanto en una pila de material homogéneo como en una pila de material no homogéneo, la política RLDiscreta logra mejores resultados que la política RLContinua. Con esto, agregar la dinámica completa de la máquina objetivo permite disminuir el *reality-gap* y así lograr mejores carguíos. Los detalles de las dinámica de la máquina son detallados en la Sección 5.1.3.

Si bien los resultados de la Sección 5.2.2 y la Sección 5.2.3 muestran que ambas políticas RL logran realizar carguíos de material, es necesario comparar con otros agentes para evaluar el éxito de la solución propuesta. Por un lado, al comparar las políticas RL y el algoritmo de Tampier, tanto en material homogéneo como en el material no homogéneo, se tiene que ambas políticas logran superar los resultados del algoritmo de Tampier. La principal mejora es con respecto al resbalamiento de las ruedas, en donde ambas políticas logran reducir el resbalamiento en ambos ejes. Por ejemplo, la política RLDiscreta logra reducir más de tres veces el resbalamiento en el eje delantero que el algoritmo Tampier. Por otro lado, al comparar las políticas RL y el agente Teleop, se tiene que en la pila de material homogéneo, ambas políticas RL lograron superar los resultados del agente Teleop, sin embargo, en la pila de material no homogéneo, el agente Teleop obtiene mejores resultados salvo por el tiempo de carguío. Una de las ventajas que tiene un operador, es

que puede detectar más fácilmente situaciones indeseadas como el resbalamiento de las ruedas o que la máquina esté atascada, por lo que puede reaccionar antes y tomar medidas que permitan cargar exitosamente. Además, el agente Teleop tiene como información extra la visión del punto de extracción mediante la cámara instalada en la máquina, y al ver la pila de material puede estimar la dificultad del carguío puesto que puede ver el tamaño de algunas rocas y así planear una estrategia de carguío. Estos resultados muestran que ambas políticas RL son capaces de realizar carguíos y superar el desempeño de otros algoritmos o teleoperadores.

Para que las políticas RL logren realizar los carguíos de forma exitosa al nivel del algoritmo Tampier y el agente Teleop, es importante que las observaciones sean correctas. La observación de la pendiente es una de las dos observaciones que entregan información directa sobre el ambiente al agente, con la distancia entre la punta del balde y el límite de profundidad siendo la otra observación. Los experimentos realizados con errores en la observación de la pendiente permiten evaluar cuánto error en esta observación puede haber antes de que los carguíos se vean afectados y comiencen a fallar. El caso en que la pendiente de la pila de material es de 15 grados y el error entrega observaciones de hasta 30 grados que se presenta en la Tabla 5.13. En este se muestra que un error en la observación de pendiente que aumenta la pendiente percibida genera una disminución en la cantidad de material cargado hasta un mínimo del 66.6 % de la capacidad del balde, sin embargo, todos los carguíos son finalizados y la máquina no se queda atascada. En contraste, se tiene el caso en que la pendiente de la pila de material es de 30 grados y el error entrega observaciones de hasta 15 grados, el cual se muestra en la Tabla 5.12, en donde al alcanzar el máximo error la máquina se queda atascada en todos los carguíos, por lo que el tiempo de carguío y el resbalamiento de las ruedas aumenta significativamente.

De esta forma, se tiene que ambas políticas RL son capaces de realizar carguíos exitosos incluso cuando la observación de la pendiente presenta errores, no obstante, desde los 15 grados de error existen casos en que las políticas no son capaces de terminar la maniobra de carguío. Es importante que la política sea robusta a errores en las observaciones, para que la política pueda funcionar en ambientes difíciles como lo es una mina subterránea con poca visibilidad y con polvo. En el trabajo de [62] se muestra un ejemplo del proceso de estimación de la forma de la pila.

Finalmente, desde el punto de vista general de diseñar un controlador utilizando RL y comparando con otras metodologías clásicas de control, se destacan las siguientes ventajas y desventajas. Por un lado, al utilizar un controlador basado en RL, para introducir cualquier cambio a la política, por ejemplo para ajustar el modelo al momento de desplegar el controlador en el mundo real, se debe entrenar nuevamente toda la política, provocando largos tiempos de iteración. Por otro lado, controladores basados en conocimiento experto, como el algoritmo de Tampier presentado en este trabajo, pueden ser ajustados al momento de despliegue, lo que permite ajustar de mejor manera el controlador al mundo real. En el caso del agente Teleop, sus operadores con suficiente tiempo y experiencia, pueden disminuir su nivel de resbalamiento y aumentar la cantidad de material cargado, puesto que con cada carguío van perfeccionando su técnica y ganando experiencia. Por lo tanto, considerando todos los resultados obtenidos en este trabajo, un controlador RL es una opción viable para excavar material utilizando un LHD de forma autónoma.

Capítulo 7

Conclusión

En este trabajo se aborda el problema de excavación autónoma con LHDs mediante la propuesta de un sistema de control basado en aprendizaje reforzado profundo. Este agente RL está diseñado para poder ser integrado directamente en sistemas de carguío completo: como el propuesto en [62] o en [61] en donde se reemplaza el módulo que ejecuta la etapa de excavación.

La política es entrenada únicamente en simulación, la cual tiene como base la función analítica propuesta en [79] pero fue modificada para que fuese compatible con máquinas de carguío frontal como un LHD. Esta forma de simular la interacción entre el material y el balde del LHD tiene como ventaja un bajo costo computacional pero la ausencia de físicas de contacto genera problemas. Estos fueron abordados principalmente mediante el diseño de recompensas específicas que permiten al agente aprender una política que pueda funcionar en la realidad. Se efectuaron extensivas pruebas en simulación que comprueban la estabilidad del enfoque propuesto y que el agente es exitoso al cargar en simulación.

Luego, mediante una variedad de experimentos realizados en el mundo real, se comprueba el funcionamiento del sistema RL propuesto. En primer lugar, se hacen experimentos para evaluar el desempeño del agente en puntos de extracción con distintas configuraciones y con material de distintas granulometrías. Los resultados indican que el agente logra ejecutar carguíos exitosos para todos las configuraciones de la pila de material y para todos las granulometrías del material utilizadas. No obstante, al agregar material con una granulometría mayor, el sistema obtiene peores resultados, cargando una menor cantidad de material. En segundo lugar, el agente tiene dos observaciones que entregan información sobre el punto de extracción: la pendiente de la pila de material, y la distancia entre la punta del balde y el límite de profundidad de la zona final. La ejecución de múltiples experimentos en donde se agrega una cantidad determinada de error a la observación de la pendiente permiten concluir que el agente puede cargar de forma efectiva hasta con 15 grados de error en la medición y valores superiores a esto provocarían que agente no pueda cargar. Por último, se realizan carguíos con el sistema propuesto en [62] y con operadores mediante teleoperación para tener resultados de carguíos y comparar con el desempeño del agente RL. El desempeño del sistema propuesto en este trabajo en la mayoría de los casos logra obtener mejores resultados en la cantidad de material cargado y en el tiempo de carguío comparado con los otros dos sistemas.

El resbalamiento de las ruedas es un factor importante en el uso de los LHDs puesto que repre-

senta una parte importante de los costos de mantención de la máquina. Para todos los experimentos mencionados anteriormente, se evalúa el porcentaje de resbalamiento por eje de las ruedas del LHD. El sistema propuesto alcanza un menor porcentaje de resbalamiento que el sistema propuesto en [62] pero no que los carguíos realizados con teleoperación.

Considerando los resultados anteriores, se puede concluir que los principales objetivos de este trabajo fueron cumplidos de forma satisfactoria. Se identificaron dos direcciones de investigación para extender el aporte realizado por este trabajo: probar el sistema desarrollado en un ambiente real, es decir, en una mina con un LHD y material tronado, junto con otras máquinas de carguío de material, y utilizar una simulación más realista para evaluar el desempeño del diseño de este trabajo y evaluar posibles cambios al diseño de la función de recompensa.

7.1. Trabajo futuro

Múltiples direcciones de investigación pueden ser abordadas para explorar nuevos espacios de uso y mejorar los resultados obtenidos en este trabajo.

Primero, el diseño de la solución propuesta en este trabajo puede ser probado para el entrenamiento de agentes RL para otras máquinas como cargadores frontales o cargadores skid-steer, y así evaluar la flexibilidad del diseño y su eficacia para entrenar agentes de excavación en distintas situaciones. De esta misma forma, se podría evaluar el desempeño con agentes entrenados con distintos algoritmos como por ejemplo SAC, PPO o TD3 y agentes entrenados con alguna base de conocimiento utilizando como ejemplo *Imitation Learning*.

Luego, la simulación de material es un punto crítico para mejorar los resultados. Reemplazar la simulación con algún otro modelo que sea más cercano a la realidad permitiría ajustar el diseño del agente RL, lo que permite eliminar recompensas como la de entierre del fondo del balde. Un primer paso para complementar la simulación implementada es agregar elementos discretos a la implementación de la FEE, metodología utilizada en [88]. Otra opción es cambiar completamente la simulación y explorar otras implementaciones como las utilizadas en [24, 72]. Esta nueva implementación de la simulación también permitiría agregar las observaciones de la presión hidráulica de la máquina.

Por último, futuros experimentos deben ser realizados con diferentes límites de trayectorias más acotados para evaluar si la recompensa por trayectorias es capaz de guiar la trayectoria de la punta de la pala sin la necesidad de las recompensas complementarias. De la misma forma, reducir el conocimiento experto entregado por las recompensas también es importante para evaluar el desempeño del agente cuando no recibe conocimiento experto sobre los distintos aspectos más importantes de la operación de LHDs, como el resbalamiento de las ruedas.

Bibliografía

[1] M. Acevedo, M. Baczynska, P. Bingoto, G. Callaway, K. Hoffman, and O. Ramsbottom, "The raw-materials challenge: How the metals and mining sector will be at the core of enabling the energy transition." Available at https://www.mckinsey.com/industries/metals-and-mining/our-insights/the-raw-materials-challenge-how-the-metals-and-mining-sector-will-be-at-the-core-of-enabling-the-energy-transition (accessed 22/06/2023), 2022.

[2] J. H. Hodgkinson and M. H. Smith, "Climate change and sustainability as drivers for the next mining and metals boom: The need for climate-smart mining and recycling," *Resources Policy*, vol. 74, 2021.

[3] Y. Ghorbani, G. T. Nwaila, S. E. Zhang, J. E. Bourdeau, M. Cánovas, J. Arzua, and N. Nikadat, "Moving towards deep underground mineral resources: Drivers, challenges and potential solutions," *Resources Policy*, vol. 80, 2023.

[4] J. Ruiz-del Solar, M. Mascaró, and C. S. Quiroz Leiva, "Automation of unit and auxiliary operations in block/panel caving: Challenges and opportunities," in *Proceedings of MassMin2020*, (Santiago, Chile), pp. 9–11, 2020.

[5] R. Tatiya, *Surface and Underground Excavations: Methods, Techniques and Equipment*. CRC Press/Balkema, 2013.

[6] Caterpillar, "Underground mining load haul dump (lhd) loaders, R2900G." Available at https://www.cat.com/en_US/products/new/equipment/underground-hard-rock/underground-mining-load-haul-dump-lhd-loaders/18234906.html (24/07/2023), 2023.

[7] Caterpillar, "Underground mining load haul dump (lhd) loaders, R1600H." Available at https://www.cat.com/en_US/products/new/equipment/underground-hard-rock/underground-mining-load-haul-dump-lhd-loaders/18509152.html (12/09/2023), 2023.

[8] R. Filla, "Simulating operability of wheel loaders: Operator models and quantification of control effort," *Fachtagung Baumaschinentechnik, TU Dresden*, 2012.

[9] R. Filla, M. Obermayr, and B. Frank, "A study to compare trajectory generation algorithms for automatic bucket filling in wheel loaders," in *3rd Commercial Vehicle Technology Symposium*, pp. 588–605, 2014.

[10] Consejo Minero, "Reporte Anual," tech. rep., Consejo Minero, Santiago, Chile, 2022.

[11] ICSG, "World Copper Factbook," tech. rep., ICSG, Lisbon, Portugal, 2022.

[12] C. Jamasmie, "Codelco kicks off underground mining at chuquicamata." Available at https://www.mining.com/chiles-codelco-kicks-off-underground-mining-chuquicamata/ (accessed 24/06/2023), 2019.

[13] Codelco, "División el teniente." Available at https://www.codelco.com/operaciones/el-teniente/nosotros/division-el-teniente (26/07/2023), 2023.

[14] Corporación Alta Ley, "Roadmap: Digitalización para una minería 4.0," tech. rep., Corporación Alta Ley, Santiago, Chile, 2022.

[15] ASI, "Autonomous haulage." Available at https://asirobots.com/mining/autonomous-haulage/ (26/06/2023), 2023.

[16] Caterpillar, "Scalable solutions from the leader in mining automation." Available at https://www.cat.com/en_US/by-industry/mining/autonomy-leadership.html (26/06/2023), 2023.

[17] Epiroc, "Automation and information management." Available at https://www.epiroc.com/en-na/innovation-and-technology/automation-and-information-management (26/06/2023), 2023.

[18] Hitachi, "Autonomous haulage system (AHS)." Available at https://www.hitachicm.com/global/en/solutions/solution-linkage/ahs/ (26/06/2023), 2023.

[19] Komatsu, "Autonomous haulage system." Available at https://www.komatsu.com.au/innovation/autonomous-haulage-system (26/06/2023), 2023.

[20] D. Ali and S. Frimpong, "Artificial intelligence, machine learning and process automation: existing knowledge frontier and way forward for mining sector," *Artificial Intelligence Review*, vol. 53, pp. 6025–6042, Dec 2020.

[21] H. V. Nguyen and H. M. La, "Review of deep reinforcement learning for robot manipulation," *2019 Third IEEE International Conference on Robotic Computing (IRC)*, pp. 590–595, 2019.

[22] E. Marchesini and A. Farinelli, "Discrete deep reinforcement learning for mapless navigation," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10688–10694, 2020.

[23] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. baker, M. Lai, A. Bolton, Y. Chen, T. P. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of go without human knowledge," *Nature*, vol. 550, pp. 354–359, 2017.

[24] S. Backman, D. Lindmark, K. Bodin, M. Servin, J. Mörk, and H. Löfgren, "Continuous control of an underground loader using deep reinforcement learning," *Machines*, vol. 9, no. 10, p. 216, 2021.

[25] Q. Lu, Y. Zhu, and L. Zhang, "Excavation reinforcement learning using geometric representation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4472–4479, 2022.

[26] T. Osa and M. Aizawa, "Deep reinforcement learning with adversarial training for automated excavation using depth images," *IEEE Access*, vol. 10, pp. 4523–4535, 2022.

[27] R. Bellman, "A markovian decision process," *Indiana University Mathematics Journal*, vol. 6, pp. 679–684, 1957.

[28] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[29] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7559–7566, 2017.

[30] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, *et al.*, "Mastering chess and shogi by self-play with a general reinforcement learning algorithm," *arXiv preprint arXiv:1712.01815*, 2017.

[31] D. Ha and J. Schmidhuber, "Recurrent world models facilitate policy evolution," *Advances in neural information processing systems*, vol. 31, 2018.

[32] T. Anthony, Z. Tian, and D. Barber, "Thinking fast and slow with deep learning and tree search," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[33] V. Feinberg, A. Wan, I. Stoica, M. I. Jordan, J. E. Gonzalez, and S. Levine, "Model-based value estimation for efficient model-free reinforcement learning," *arXiv preprint arXiv:1803.00101*, 2018.

[34] S. Racanière, T. Weber, D. Reichert, L. Buesing, A. Guez, D. Jimenez Rezende, A. Puigdomènech Badia, O. Vinyals, N. Heess, Y. Li, *et al.*, "Imagination-augmented agents for deep reinforcement learning," *Advances in neural information processing systems*, vol. 30, 2017.

[35] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, pp. 1928–1937, PMLR, 2016.

[36] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[37] N. Sanghi, "Deep q-learning," in *Deep Reinforcement Learning with Python: With PyTorch, TensorFlow and OpenAI Gym*, pp. 155–206, Berkeley, CA: Apress, 2021.

[38] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[39] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2016.

[40] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*, pp. 1861–1870, PMLR, 2018.

[41] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *arXiv preprint arXiv:2005.01643*, 2020.

[42] N. Jaques, A. Ghandeharioun, J. H. Shen, C. Ferguson, A. Lapedriza, N. Jones, S. Gu, and R. Picard, "Way off-policy batch deep reinforcement learning of implicit human preferences in dialog," *arXiv preprint arXiv:1907.00456*, 2019.

[43] G. Kahn, P. Abbeel, and S. Levine, "Badgr: An autonomous self-supervised learning-based navigation system," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1312–1319, 2021.

[44] J. Hansen, F. Hogan, D. Rivkin, D. Meger, M. Jenkin, and G. Dudek, "Visuotactile-rl: learning multimodal manipulation policies with deep reinforcement learning," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 8298–8304, IEEE, 2022.

[45] R. Ramamurthy, P. Ammanabrolu, K. Brantley, J. Hessel, R. Sifa, C. Bauckhage, H. Hajishirzi, and Y. Choi, "Is reinforcement learning (not) for natural language processing?: Benchmarks, baselines, and building blocks for natural language policy optimization," *arXiv preprint arXiv:2210.01241*, 2022.

[46] Z. Xu, B. Liu, X. Xiao, A. Nair, and P. Stone, "Benchmarking reinforcement learning techniques for autonomous navigation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9224–9230, IEEE, 2023.

[47] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, pp. 279–292, 1992.

[48] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[49] S. Dadhich, U. Bodin, and U. Andersson, "Key challenges in automation of earth-moving machines," *Automation in Construction*, vol. 68, pp. 212–222, 2016.

[50] R. Filla and B. Frank, "Towards finding the optimal bucket filling strategy through simulation," in *Proceedings of the 15th Scandinavian International Conference on Fluid Power Held at Linköping University, Linköping, Sweden*, pp. 7–9, 2017.

[51] Y. Chen, G. Shi, C. Tan, and Z. Wang, "Machine learning-based shoveling trajectory optimization of wheel loader for fuel consumption reduction," *Applied Sciences*, vol. 13, no. 13, 2023.

[52] Y. Chen, H. Jiang, G. Shi, and T. Zheng, "Research on the trajectory and operational performance of wheel loader automatic shoveling," *Applied Sciences*, vol. 12, no. 24, 2022.

[53] Y. Meng, H. Fang, G. Liang, Q. Gu, and L. Liu, "Bucket trajectory optimization under the automatic scooping of lhd," *Energies*, vol. 12, no. 20, 2019.

[54] B. wei Cao, X. Liu, W. Chen, K. Yang, and P. Tan, "Skid-proof operation of wheel loader based on model prediction and electro-hydraulic proportional control technology," *IEEE Access*, vol. 8, pp. 81–92, 2020.

[55] J. A. Marshall, P. F. Murphy, and L. Daneshmend, "Toward autonomous excavation of fragmented rock: Full-scale experiments," *IEEE Transactions on Automation Science and Engineering*, vol. 5, pp. 562–566, 2008.

[56] A. A. Dobson, J. A. Marshall, and J. Larsson, "Admittance control for robotic loading: Design and experiments with a 1-tonne loader and a 14-tonne load-haul-dump machine," *Journal of Field Robotics*, vol. 34, no. 1, pp. 123–150, 2016.

[57] H. A. Fernando, J. A. Marshall, H. Almqvist, and J. Larsson, "Towards controlling bucket fill factor in robotic excavation by learning admittance control setpoints," in *Field and Service Robotics: Results of the 11Th International Conference*, pp. 35–48, Springer, 2018.

[58] H. A. Fernando, J. A. Marshall, and J. Larsson, "Iterative learning-based admittance control for autonomous excavation," *Journal of Intelligent & Robotic Systems*, vol. 96, pp. 493 – 500, 2019.

[59] K. Aoshima, M. Servin, and E. Wadbro, "Simulation-based optimization of high-performance wheel loading," *arXiv preprint arXiv:2107.14615*, 2021.

[60] S. Dadhich, U. Bodin, F. Sandin, and U. Andersson, "Machine learning approach to automatic bucket loading," *2016 24th Mediterranean Conference on Control and Automation (MED)*, pp. 1260–1265, 2016.

[61] D. Cárdenas, P. Loncomilla, F. Inostroza, I. Parra-Tsunekawa, and J. R. del Solar, "Autonomous detection and loading of ore piles with load–haul–dump machines in room & pillar mines," *Journal of Field Robotics*, 2023.

[62] C. Tampier, M. Mascaro, and J. Ruiz-del Solar, "Autonomous loading system for load-haul-dump (lhd) machines used in underground mining," *Applied Sciences*, vol. 11, no. 18, 2021.

[63] F. Núnez, S. Navarro, A. Aguado, and A. Cipriano, "State estimation based model predictive control for lhd vehicles," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 1448–1453, 2008.

[64] T. Nayl, G. Nikolakopoulos, and T. Gustafsson, "Path following for an articulated vehicle based on switching model predictive control under varying speeds and slip angles," in *Proceedings of 2012 IEEE 17th International Conference on Emerging Technologies & Factory Automation (ETFA 2012)*, pp. 1–7, IEEE, 2012.

[65] G. Bai, Y. Meng, L. Liu, W. Luo, Q. Gu, and K. Li, "A new path tracking method based on multilayer model predictive control," *Applied sciences*, vol. 9, no. 13, p. 2649, 2019.

[66] G. Bai, L. Liu, Y. Meng, W. Luo, Q. Gu, and B. Ma, "Path tracking of mining vehicles based on nonlinear model predictive control," *Applied Sciences*, vol. 9, no. 7, p. 1372, 2019.

[67] E. Halbach, J.-K. Kämäräinen, and R. Ghabcheloo, "Neural network pile loading controller trained by demonstration," *2019 International Conference on Robotics and Automation (ICRA)*, pp. 980–986, 2019.

[68] W. Yang, N. Strokina, N. Serbenyuk, R. Ghabcheloo, and J.-K. Kämäräinen, "Learning a pile loading controller from demonstrations," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4427–4433, 2020.

[69] W. Yang, N. Strokina, N. Serbenyuk, J. Pajarinen, R. Ghabcheloo, J. Vihonen, M. M. Aref, and J.-K. Kämäräinen, "Neural network controller for autonomous pile loading revised," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2198–2204, 2021.

[70] S. Dadhich, F. Sandin, U. Bodin, U. Andersson, and T. Martinsson, "Field test of neural-network based automatic bucket-filling algorithm for wheel-loaders," *Automation in Construction*, vol. 97, pp. 1–12, 2019.

[71] S. Dadhich, F. Sandin, U. Bodin, U. Andersson, and T. Martinsson, "Adaptation of a wheel loader automatic bucket filling neural network using reinforcement learning," *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–9, 2020.

[72] O. Azulay and A. Shapiro, "Wheel loader scooping controller using deep reinforcement learning," *IEEE Access*, vol. 9, 2021.

[73] Y. Zhu, L. Wang, and L. Zhang, "Excavation of fragmented rocks with multi-modal model-based reinforcement learning," *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6523–6530, 2022.

[74] S. Wang, Y. Yin, Y. Wu, and L. Hou, "Modeling and verification of an acquisition strategy for wheel loader's working trajectories and resistance," *Sensors (Basel, Switzerland)*, vol. 22, 2022.

[75] D. Schmidt, M. Proetzsch, and K. Berns, "Simulation and control of an autonomous bucket excavator for landscaping tasks," *2010 IEEE International Conference on Robotics and Automation*, pp. 5108–5113, 2010.

[76] M. Tariq, A. Gustafson, and H. Schunnesson, "Training of load haul dump (lhd) machine operators: a case study at LKAB's Kiirunavaara mine," *Mining Technology*, pp. 1–16, 2023.

[77] P. Egli, D. Gaschen, S. Kerscher, D. Jud, and M. Hutter, "Soil-adaptive excavation using reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, pp. 9778–9785, 2022.

[78] D. Holz, A. Azimi, M. Teichmann, and S. Mercier, "Real-time simulation of mining and earth-moving operations: A level set-based model for tool-induced terrain deformations," in *30th International Symposium on Automation and Robotics in Construction and Mining ISARC*, 2013.

[79] O. Luengo, S. Singh, and H. Cannon, "Modeling and identification of soil-tool interaction in automated excavation," in *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No. 98CH36190)*, vol. 3, pp. 1900–1906, IEEE, 1998.

[80] A. R. Reece, "The fundamental equation of earth-moving mechanics," *Proceedings of the Institution of Mechanical Engineers, Conference Proceedings*, vol. 179, no. 6, pp. 16–22, 1964.

[81] N. P. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, pp. 2149–2154 vol.3, 2004.

[82] S. Levine, N. Wagener, and P. Abbeel, "Learning contact-rich manipulation skills with guided policy search (2015)," *arXiv preprint arXiv:1501.05611*, 2015.

[83] Geotech Data, "Geotechnical parameters." Available at `https://www.geotechdata.info/parameter` (31/12/2023), 2023.

[84] Fine, "Table of ultimate friction factors for dissimilar materials." Available at `https://www.finesoftware.eu/help/geo5/en/table-of-ultimate-friction-factors-for-dissimilar-materials-01/` (31/12/2023), 2023.

[85] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, p. 5, Kobe, Japan, 2009.

[86] Linux Foundation, "Pytorch." Available at `https://pytorch.org/` (29/09/2023), 2023.

[87] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, p. 2280–2292, 06 2014.

[88] D. Holz, A. Azimi, and M. Teichmann, “Advances in physically-based modeling of deformable soil for real-time operator training simulators,” in *2015 international conference on virtual reality and visualization (ICVRV)*, pp. 166–172, IEEE, 2015.