Universidad Técnica Federico Santa María

Tesis de Magíster

Reconocimiento de modelo corporal a través de estímulos táctiles auto-generados en robot humanoide iCub

Autor:
Pablo Reyes Robles

Supervisor:
María-José Escobar, PhD.

Tesis presentada como requisito parcial para optar al grado de Magíster en Ciencias de la Ingeniería Electrónica

en el

Departamento de Ingeniería Electrónica

17 de octubre de 2025

CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☐ Memoria o trabajo de título ☒ Tesis de Postgrado

Título del trabajo: Reconocimiento de modelo corporal a través de estímulos táctiles auto-generados en robot humanoide iCub

Nombre del candidato(a): Pablo Gabriel Reyes Robles

Carrera / Grado: Magíster en Ciencias de la Ingeniería Electrónica

Campus: Casa Central **Departamento:** Departamento de Electrónica

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, María José Escobar Silva, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por **(marcar una opción)**:

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

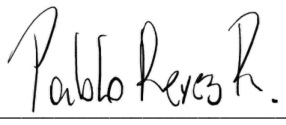
Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 18/12/2025 **Firma:** 

Estudiante o Candidato(a):

Fecha: 18/12/2025 **Firma:** 

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

Dedico este trabajo a mi familia, amigos, compañeros de proyectos y a la música. Cada uno cumplió un rol fundamental en mi vida y para lograr este trabajo. No podría estar escribiendo este texto sin uno de esos pilares.

Agradecimientos

Agradezco a mi profesora María José Escobar, siempre con las palabras certeras y con una confianza que no puedo dejar de atesorar. Un largo recorrido de coincidencias y curiosidades científicas, que me dio la oportunidad de cuestionar el saber de formas que no pude haber hecho sin contar con su tutoría.

Agradezco al equipo de Sysmic Robotics y a toda su gente a lo largo de su historia. No hubo mejor forma de aprender que estar ahí, fallando y reparando, estudiando y mejorando.

Índice general

Índice de figuras

Índice de tablas

UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

Abstract

Departamento de Ingeniería Electrónica

Magíster en Ciencias de la Ingeniería Electrónica

Reconocimiento de modelo corporal a través de estímulos táctiles auto-generados en robot humanoide iCub

by Pablo Reyes Robles

El reconocimiento del propio cuerpo a través del sentido del tacto es una habilidad fundamental en seres humanos, pero poco explorada en robótica humanoide. Esta tesis propone un enfoque bioinspirado para que un robot humanoide, específicamente el iCub, construya un modelo de su torso mediante autoexploración táctil utilizando su propio dedo pulgar como efector. Para ello, se hace uso de un algoritmo de meta-aprendizaje basado en plasticidad Hebbiana, optimizado mediante el algoritmo libre de gradientes Evolution Strategies (ES), que permite a redes neuronales aprender dinámicamente reglas de exploración durante el experimento. A su vez, se integra un modelo de superficie mediante Procesos Gaussianos (GP) para estimar la forma del torso y proporcionar una señal de recompensa intrínseca basada en la reducción de incertidumbre del modelo. Los experimentos se realizaron en un entorno simulado con el simulador Gazebo y el middleware YARP, replicando las condiciones del robot físico. Aunque la hipótesis principal no fue confirmada en su totalidad, los resultados obtenidos evidencian las capacidades del enfoque propuesto y permiten identificar mejoras metodológicas para futuras investigaciones, especialmente en la incorporación de información espacial a la señal de recompensa.

Capítulo 1

Introducción

El reconocimiento del modelo del propio cuerpo, es una tarea que el ser humano comienza a ejecutar incluso antes de haber nacido. Comprender su dimensionalidad, permite proyectar las sensaciones táctiles a un punto en el espacio que se encuentra embebido en nuestra piel. Gracias a este proceso, como seres humanos, somos capaces de interpretar instantáneamente dónde estamos siendo estimulados y podemos con seguridad indicar, con nuestras manos, el punto exacto donde está siendo excitado dicho estímulo. Este proceso natural nos permite reaccionar a estímulos como picazones, predecir instantáneamente si nosotros mismos nos estamos tocando o reaccionar a eventos generados por terceros.

En la actualidad, la mayoría de robots, tanto para investigación como para uso comercial o industrial, sub utilizan el sentido del tacto. En su lugar, se depende en gran medida de técnicas de visión por computador, sensores de distancia y odometría para estimar la posición del robot o partes de éste, tanto para desplazamientos como movimientos articulados de efectores. Sin embargo, el sentido del tacto juega un papel fundamental en nuestra relación con el ambiente y con otros seres vivos. Nos proporciona un sistema de percepción espacial muy sensible que funciona como una variable de control imprescindible para realizar tareas de manipulación complejas y es una herramienta vital en la interacción social. Por lo tanto, es esencial que los robots puedan aprovechar plenamente el sentido del tacto para mejorar su capacidad de interactuar y manipular su entorno.

Este trabajo aborda el problema de exploración y reconocimiento del propio cuerpo como un problema de *black-box optimization*, donde el agente debe ser capaz de maximizar el área explorada de una parte de su cuerpo, utilizando la plataforma robótica iCub para este objetivo. Este robot posee una gran cantidad de actuadores y sensores en su cuerpo, que le permiten desarrollar comportamientos humanoides y con ello profundizar en estudios sobre robótica cognitiva. Este trabajo aprovecha sensores táctiles que están dispuestos por todo el cuerpo del iCub, enfocándose en el pecho, el que debe ser estimulado con acciones exploratorias usando el pulgar de la mano izquierda del mismo robot como efector. La activación de puntos táctiles o *taxels* define la señal de refuerzo y la exploración se modela como optimización de caja negra; los detalles del entrenamiento y del controlador cartesiano se presentan en el Capítulo de Método.

Junto con el método de exploración recién descrito, se utiliza un modelo de regresión a partir de un proceso Gaussiano (GP) que estime la forma del pecho según ha sido descubierto por el propio agente. Se utiliza un modelo GP para representar la superficie del torso, su incertidumbre guía la exploración como recompensa intrínseca.

En esta tesis el foco no es maximizar rendimiento absoluto, sino examinar la aplicabilidad de principios bioinspirados a la exploración táctil, estableciendo *qué mecanismos funcionan, bajo qué condiciones y con qué evidencias observables.*

1.1. Hipótesis y objetivos

Hipótesis. *Los mecanismos bioinspirados de plasticidad Hebbiana, combinados con una señal de curiosidad basada en la reducción de incertidumbre de un Proceso Gaussiano (GP), son* ***aplicables*** *a la auto-exploración táctil en robótica humanoide en el sentido de que inducen* ***patrones de exploración dirigidos por información*** *(reducción sostenida de incertidumbre y descubrimiento de taxels novedosos) bajo restricciones cinemáticas y sensoriales realistas.*

Objetivo general. Evaluar la aplicabilidad de un esquema bioinspirado (plasticidad Hebbiana + GP) para auto-exploración táctil en un robot humanoide, caracterizando sus mecanismos, condiciones de validez y efectos observables.

Objetivos específicos.

1. Formalizar el pipeline: entradas (taxels + pose), salida (paso cartesiano) y señal intrínseca (reducción de incertidumbre GP).
2. Definir métricas de aplicabilidad: caída de incertidumbre, descubrimiento de taxels únicos.
3. Establecer un protocolo reproducible y obtener resultados temporales de desempeño.
4. Analizar condiciones que favorecen o degradan la aplicabilidad (sensibilidad de los elementos de simulación, restricciones geométricas).
5. Validar la aplicabilidad del método y su capacidad exploratoria en el área total.

1.2. Contribución de esta tesis

En este trabajo se optó por utilizar un algoritmo de optimización de caja negra para la tarea de exploración, buscando generar un experimento similar a los vistos en tareas de aprendizaje y navegación de un agente virtual. En este apartado, hay una gran variedad de sistemas de recompensa que pueden ser implementados, pudiendo utilizar reforzamientos externos como la cantidad de puntos nuevos que han sido descubiertos, así como la generación de recompensas internas que puedan depender de la capacidad de predicción de los propios estados futuros del agente o la incertidumbre del modelo basado en GP.

Los modelos propuestos presentan un problema de generalización cuando se intenta obtener la representación espacial de *taxels* cercanos que no fueron incluidos en el conjunto de entrenamiento, tarea que los modelos testeados no son capaces de resolver. En respuesta, este trabajo de tesis propone un método de exploración con rutinas de auto-tacto, a partir de la implementación de un algoritmo de *Meta-learning* basado en la teoría de plasticidad Hebbiana [1], que en pocas palabras, consiste en el aprendizaje de coeficientes que modulan pesos de una red neuronal de forma dinámica durante un experimento. Se utiliza *Evolution Strategies* (ES) [2] para la evolución de coeficientes Hebbianos, que es un algoritmo libre del cálculo de gradientes con

una importante influencia de métodos neuroevolutivos. Las redes neuronales que se encuentran evolucionando reciben las activaciones de los sensores táctiles como entrada y la pose actual del efector utilizado en la exploración, teniendo como salida un paso de movimiento de la pose en el espacio tridimensional de trabajo. En conjunto, la posición de los *taxels* explorados se incorpora de forma activa al GP para construir un modelo de la forma del cuerpo objetivo, además de entregar la incertidumbre actual de dicho modelo, lo que funciona como una recompensa interna que se incrementa en la medida que el área de exploración se maximiza.

La plataforma robótica que se utilizará para los experimentos es el ya mencionado iCub, un robot humanoide con una gran cantidad de actuadores y sensores. Este robot posee una piel artificial capacitiva en todo su cuerpo, además de brazos y manos articuladas. La tarea se llevará a cabo en un entorno simulado para proteger la integridad mecánica y eléctrica del robot, pudiendo comprobar el trabajo en el robot físico una vez concluyan exitosamente los experimentos. Especificaciones de la plataforma física y simulada serán detalladas en el siguiente capítulo.

1.3. Estructura del documento

El Capítulo 2 presenta el estado del arte en exploración táctil y modelado de superficies. El Capítulo 3 reúne los antecedentes generales: descripción del robot iCub, su piel artificial y los entornos de simulación. El Capítulo 4 detalla el método propuesto, integrando la evolución de reglas locales con *Evolution Strategies* y la estimación de superficies mediante Procesos Gaussianos. El Capítulo 5 describe el diseño experimental, incluyendo la configuración de ambientes, la rutina de aprendizaje y la ejecución de experimentos. El Capítulo 6 reporta los resultados experimentales. El Capítulo 7 discute los hallazgos y propone líneas de trabajo futuro. Finalmente, el Capítulo 8 presenta las conclusiones y el Apéndice A documenta una aplicación relacionada del enfoque.

Capítulo 2

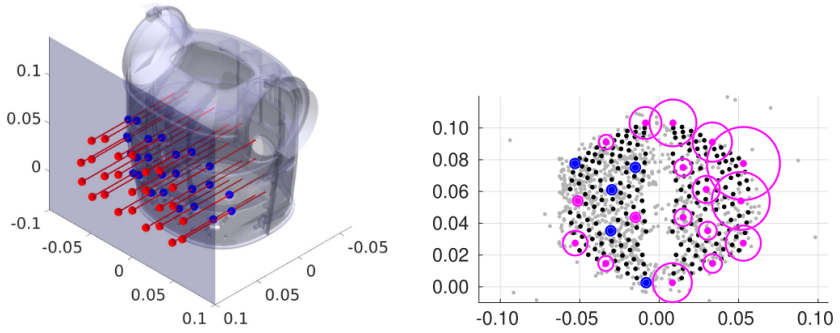
Estado del Arte

El desarrollo temprano sugiere que el auto-tacto cumple un rol estructurante en la construcción de modelos corporales: durante la gestación, se observan movimientos locales hacia áreas con alta sensibilidad (cara, plantas) y, a partir de la semana 20, patrones cinemáticos dependientes del objetivo, consistentes con conductas intencionales; tras el nacimiento, los patrones exploratorios de auto-tacto contribuyen a los primeros modelos propioceptivos y motores. [3–6]

En robótica humanoide, se han explorado mecanismos de auto-calibración por auto-tacto. Roncone [7] emplea iCub con posiciones de *taxels* predefinidas para corregir desajustes cinemáticos a partir de colisiones táctiles esperadas, estableciendo un puente operativo entre lectura táctil y ajuste de modelo. Por su parte, Nguyen [8] integra visión estéreo y posturas de cuello para predecir configuraciones de brazo que produzcan contacto, mostrando que la fusión visuo-propioceptiva-táctil permite inferir acciones de alcance y contacto de forma robusta.

En exploración dirigida del propio cuerpo, Gama [9] proponen en NAO —y luego Shcherban [10] en iCub— rutinas de auto-tacto orientadas a metas sobre regiones con piel artificial, combinando discretización dinámica del espacio de interés con criterios de curiosidad para no explorar a ciegas el espacio motor completo. El enfoque aprende correspondencias táctil–motor y reduce el problema a alcanzar metas táctiles sobre el torso.

Ligado a lo anterior, la literatura de motivación intrínseca/curiosidad en aprendizaje por refuerzo ha mostrado que recompensas internas pueden guiar la exploración



(A) Proyección del cuerpo a un plano.

(B) Resultados de exploración del espacio.

FIGURA 2.1: Propuesta de espacio a explorar y resultados del trabajo de Gama [9]. Puntos azules en 2.1b indican regresión de posición con mínimo error; circunferencias magenta regresión de posición y su desviación estándar.

cuando la señal extrínseca es escasa. Pathak [11] formula la curiosidad como error de predicción de un modelo forward medido en un espacio latente aprendido mediante un modelo inverso (representation learning auto-supervisado), con el fin de centrar la representación en factores accionables. La señal intrínseca proviene del modelo forward en el espacio latente, y el resultado es una exploración más eficiente en entornos visuales. En paralelo, la exploración orientada a metas (SAGG-RIAC) traslada la búsqueda al espacio de objetivos y prioriza por progreso de competencia, mostrando mayor eficiencia muestral que el motor babbling en tareas continuas y de alta dimensión; este marco es coherente con la selección de regiones del torso como metas táctiles.

Más específicamente en tareas manipulativas, se ha investigado motivación intrínseca basada en tacto: Vulin [12] introducen una recompensa intrínseca basada en fuerza de contacto y un replay priorizado por contacto que acelera la exploración en entornos de recompensa escasa, superando baselines con HER y demostrando que señales táctiles pueden estructurar la exploración incluso sin modelar explícitamente la forma del objeto. Este resultado refuerza la idea de que una señal intrínseca ligada al hecho de contactar (o a la incertidumbre espacial local) puede guiar más eficazmente la búsqueda.

Para estimar superficies a partir de contactos, los Procesos Gaussianos (GP) proporcionan una representación probabilística que maneja ruido y entrega incertidumbre, habilitando estrategias de exploración activa guiadas por dicha incertidumbre. En

robótica táctil se han usado GP para modelado implícito/explícito de superficies y para refinamiento de nubes de puntos obtenidas por visión, con aplicaciones a pose de objetos, agarre y estimación de fuerza. [13–19] En este sentido, la recompensa intrínseca basada en reducción de varianza del GP (empleada en este trabajo) es consistente con la literatura de exploración activa: incentiva explorar regiones poco muestreadas donde el modelo es más incierto.

Finalmente, la calidad del mapeo táctil-motor depende de la calibración métrica de la piel. Del Prete [20] estima la localización de *taxels* en iCub a partir de mediciones fuerza/par en la cadena cinemática y restricciones de superficie, alcanzando errores medios del orden de milímetros. Disponer de cartografía métrica reduce sesgos en la asociación pose–contacto, mejorando tanto métricas de cobertura como señales basadas en incertidumbre.

Se converge en tres lineamientos útiles para este trabajo: (i) emplear señales intrínsecas que ponderen espacialmente novedad/incertidumbre (curiosidad con componente espacial o contacto informativo), (ii) formular la exploración en el espacio de metas táctiles (regiones del torso), y (iii) asegurar una calibración métrica de la piel para que la relación pose–contacto soporte el modelado con GP. Estas ideas enmarcan y motivan la adopción de un esquema de optimización de caja negra con plasticidad hebbiana y recompensa intrínseca basada en GP, junto con las extensiones propuestas (modulación espacial de la intrínseca y entradas explícitas de cobertura al generador de acciones).

Capítulo 3

Antecedentes generales

3.1. Robot iCub

iCub es un robot humanoide de código abierto, diseñado y construido por el Istituto Italiano di Tecnologia (IIT)[1], para la investigación en robótica cognitiva e inteligencia artificial. El iCub tiene una altura de 104 cm. y pesa alrededor de 30 kg., mostrando proporciones similares a las de un niño de 5 años. Tiene la capacidad de gatear, caminar o equilibrarse, así como manipular objetos de manera sofisticada gracias al diseño fuertemente articulado de sus manos.

En la Tabla 3.1 se detalla la composición de sus 53 grados de libertad, que le dan la posibilidad de realizar una gran cantidad de acciones de movimiento, siempre que las restricciones físicas del mismo diseño lo permitan. Además cuenta con una gran cantidad de sensores como un par cámaras en la posición de sus ojos, micrófonos posicionados de forma lateral en su cabeza para simular audición, unidades de medición inercial (IMU) en su cabeza, encoders en todas sus articulaciones y sensores de torque en articulaciones principales para tareas de fuerza y equilibrio. El último grupo de sensores y más relevantes para este trabajo son los de carácter táctil, ubicados en la punta de sus dedos, palmas, brazos, pecho, piernas y planta de los pies. Un resumen de la capacidad sensorial del iCub se muestra en la Tabla 3.2.

El iCub posee en su cabeza un computador que centraliza información proveniente de distintos módulos distribuidos por su cuerpo. El *backbone* principal de comunicación

[1] https://icub.iit.it

FIGURA 3.1: Robot iCub

TABLA 3.1: Grados de libertad iCub

Componente	Grados de libertad	Notas
Ojos	3	Vergencia independiente e inclinación común
Cabeza	3	Posicionados en el cuello para inclinación, balanceo y rotación
Pecho	3	También para inclinación, balanceo y rotación
Brazos	7 (cada uno)	3 grados de libertad en el hombro, 1 en el codo y 3 en la muñeca
Manos	9 (cada una)	Pulgar, índice y medio con articulaciones interfalanges independientes, mientras que el anular y meñique se encuentran acoplados. El pulgar además puede rotar sobre la palma
Piernas	6 (cada una)	3 grados de libertad en la parte superior, 1 en la rodilla y 2 en el tobillo

Tabla 3.2: Sensores iCub

Sensor	Cantidad	Notas
Cámaras	2	Montados en los ojos para visión estéreo
Micrófonos	2	Micrófonos estéreos omnidireccionales para estudios de audición binaural
IMUs	1	Unidad compuesta de giroscopio de 3 ejes, acelerómetro de 3 ejes y magnetómetro de 3 ejes, ubicada en el cuello del iCub
Encoders	Para cada articulación	Encoders magnéticos absolutos, incrementales, de distinta resolución dependiente del tamaño de la articulación y necesidades de precisión
Fuerza y torque	6	Montados en la parte superior de los brazos y piernas, más 2 adicionales cerca de los tobillos para una mejor estimación de punto de momento cero (ZMP)
Táctiles	Más de 3000 puntos	Sensores táctiles capacitivos instalados en las yemas de dedos, palmas, brazos, pecho, piernas y plantas de pie

que recorre el cuerpo del iCub, un símil del sistema nervioso central, se lleva a cabo a través de una conexión bajo estándar Ethernet. Los módulos son placas de circuitos, que jerarquizan sus funciones para crear un *backbone* secundario basado en protocolo CAN, similar en este caso al sistema nervioso periférico. Esta red periférica se compone de módulos o conectores que coordinan la comunicación de aquellos que interactúan directamente con los actuadores o sensores ya descritos en esta sección, enviando y recibiendo instrucciones desde el *backbone* principal [21].

El desarrollo del software para el uso y manejo del robot iCub, tiene un carácter estrictamente modular. En la medida que el iCub experimente algún tipo de cambio estructural o de hardware, la arquitectura de software tiene la capacidad de incorporar dichos cambios sin experimentar reimplementación estructural de código.

La coordinación del flujo de datos de los elementos que componen el iCub es llevada a cabo por el *middleware* **Y**et **A**nother **R**obot **P**latform o YARP[22], que soporta un sistema de control de robots basado en una colección de programas que comunican las piezas de hardware distribuidas en el iCub de una manera *peer-to-peer*.

Su desarrollo y mantenimiento es llevado a cabo por el IIT para diversos proyectos en robótica, compartiendo una gran cantidad de similitudes con ROS[2] pero con un menor impacto en la industria y centros de investigación, focalizándose principalmente en funcionalidades específicas para la integración y control de robots humanoides. A diferencia de ROS, que ha sido adoptado ampliamente en diversas áreas de la robótica, YARP se centra en proporcionar una infraestructura robusta y flexible para la comunicación en tiempo real entre los módulos que conforman sistemas robóticos complejos, como el iCub. Esto le permite manejar de manera eficiente el procesamiento de señales sensoriales, el control de actuadores y la sincronización de múltiples componentes en entornos distribuidos.

3.1.1. Piel artificial

Como ha sido brevemente discutido en la sección previa, el iCub posee una piel artificial que cubre múltiples partes de su cuerpo. Esta piel está hecha de sensores táctiles capacitivos situados sobre placas de circuitos flexibles de forma triangular. Cada una de estas placas posee 10 puntos de tacto y se encuentran interconectadas siendo manejadas por un único microcontrolador.

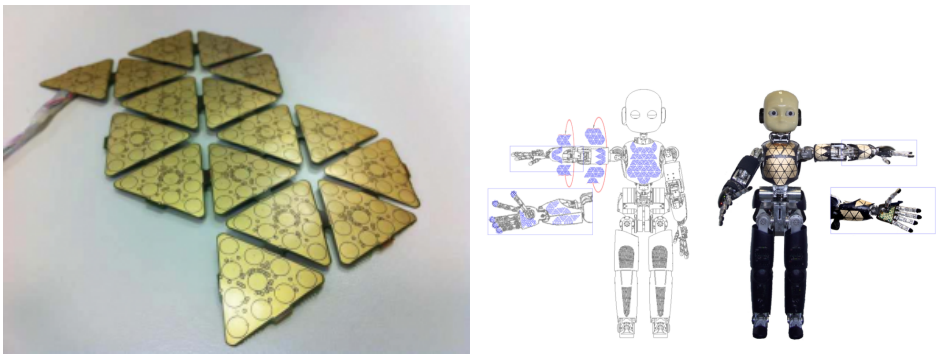
Los puntos de tacto se denominan *taxels* o píxeles táctiles, ubicando:

- 440 *taxels* en el torso
- 380 *taxels* en cada brazo
- 230 *taxels* en cada antebrazo
- 104 *taxels* en cada mano (44 en la palma y 12 en cada dedo)
- 720 *taxels* en cada pierna
- 150 *taxels* en la planta de cada pie

En la Figura 3.2a se muestra el diseño físico de las placas triangulares.

La captura de información de los sensores táctiles involucra centralizar los datos mediante un sistema de comunicación basado en una configuración heterogéneo de

[2] https://www.ros.org/



(A) Sensores táctiles capacitivos modulares tipo parche [23].

(B) Disposición de sensores sobre el iCub [24].

FIGURA 3.2: La red de sensores táctiles del iCub se encuentra organizada en parches. Se encuentran físicamente conectados entre ellos y son leídos por un único microntrolador.

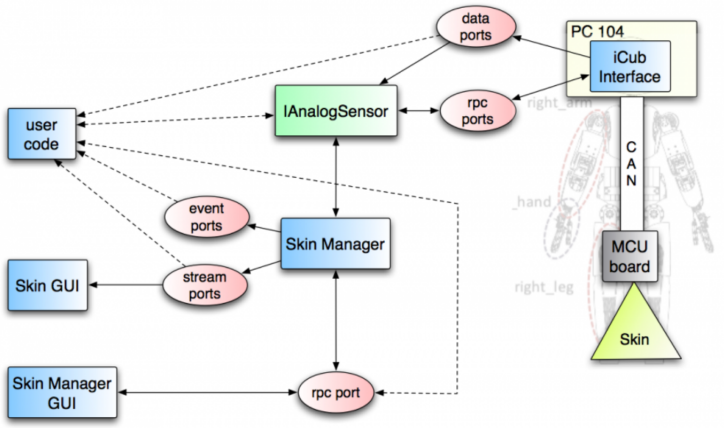


FIGURA 3.3: Flujo de datos táctiles. Imagen de [23].

cómputo. Cada parte del cuerpo posee un dispositivo que concentra las mediciones realizadas por cada parche de *taxels* como dato crudo, consistente en valores de 0 a 255 dependiendo del nivel de presión hecha sobre el *taxel*.

YARP posee un sistema de suscriptor/publicador, tanto para sus actuadores y sensores. En el caso despecífico de los *taxels*, los datos crudos de medición se envían a los puertos de YARP de forma `/icub/skin/part_name`, con `part_name` la parte asociada a medir. Luego un módulo integrado transforma los datos a punto flotante con los mismos rangos de 0.0 a 255.0 y genera un acceso para el código de usuario en el puerto `/icub/skin/part_name_comp`, con `part_name_comp` la parte específica de medición.

La Figura 3.3 muestra el recorrido de la información desde la piel capacitiva hasta las aplicaciones de usuario y las GUI:

1. **Skin → MCU board → iCub Interface (PC104).** Las placas microcontroladoras (MCU) adquieren los valores de los *taxels* y los envían por el bus `CAN` hacia el *iCub Interface* del PC104, que expone los datos al middleware YARP como puertos de datos (*data ports*). Estos puertos siguen la convención `/icub/skin/`*`part_name`* (ej., `/icub/skin/torso`). Además, por cada puerto de datos existe un puerto rpc (*rpc port*) emparejado para enviar comandos de *reset* a los microcontroladores.

2. **Lectura directa o vía `IAnalogSensor`.** El flujo puede consumirse directamente desde el puerto de datos o a través de la interfaz `IAnalogSensor` de YARP, que entrega la lectura analógica de sensores.

3. **`skinManager`.** Un módulo intermedio (`skinManager`) recibe los datos crudos y genera una segunda familia de puertos compensados con el sufijo `_comp`: `/icub/skin/`*`part_name`*`_comp`. Estos aplican compensación de deriva térmica y filtrado de alta frecuencia antes de publicarlos como puertos de *stream* continuos.

4. **Puerto de *eventos*.** Además, `skinManager` publica un evento de alto nivel (`skinContactList`) que entrega *contactos* agregados (centro de presión, fuerza estimada, enlace del robot), abstrayéndose de la disposición física de sensores individuales.

5. **Capa de usuario.** El *user code* puede suscribirse a: (i) los datos crudos (`/icub/skin/`*`part`*) cuando se requiere acceso de bajo nivel, (ii) los datos compensados (`/icub/skin/`*`part`*`_comp`), o (iii) la lista de contactos cuando interesa trabajar con unidades táctiles ya detectadas.

6. **GUI.** `iCubSkinGui` suele conectarse a los puertos compensados para visualización 2D de parches de piel; el `Skin Manager GUI` permite ajustar parámetros de filtrado, alternar visualización y monitorizar el estado de sensores.

La convención de nombres `/icub/skin/`*`part_name`* y `/icub/skin/`*`part_name`*`_comp` determina dos rutas de consumo: una de bajo nivel orientada a procesamiento propio

y otra pre-filtrada lista para control. El puerto *rpc* asociado habilita acciones de mantenimiento, mientras que el puerto de eventos expone contactos ya agregados para aplicaciones que no requieren operar con cada *taxel* individual.

3.1.2. Software y entornos de simulación

Además de tratarse de una compleja máquina de ingeniería mecánica y electrónica, el iCub posee una gran biblioteca de código que está disponible de forma abierta como un grupo de repositorios del ecosistema iCub[3].

Entre la oferta de herramientas disponibles, existe un simulador de desarrollo exclusivo para la plataforma iCub y que ha debido sido mantenido el equipo de desarrollo del ecosistema iCub. Alternativas de software de simulación cuentan con equipos dedicados a desarrollar software que sea estable, con motores de físicas especializados, multi-plataforma y fáciles de integrar con otros robots y objetos de simulación, como por ejemplo CoppeliaSim[4] o MuJoCo[5]. Bajo este escenario, el desarrollo del simulador exclusivo fue descontinuado del ecosistema iCub para dar paso a la utilización de alternativas más robustas y de mayor impacto. El software de simulación que mayor alcance tiene en la misma comunidad de iCub es Gazebo[6], desarrollado bajo un modelo *open-source*, provee motores de físicas para el estudio de dinámicas en los escenarios de estudio y soporte para la codificación de simulación de sensores y control de actuadores. En la actualidad el ecosistema de software del iCub provee conexión mediante YARP al simulador Gazebo, para poder interactuar con el robot simulado de la misma forma como si se tratase del robot físico.

3.2. Exploración mediante estímulos táctiles

Agentes biológicos muestran altos niveles de adaptación y velocidad de aprendizaje. A pesar de que los mecanismos subyacentes no han sido completamente descifrados, está bien establecido que la plasticidad sináptica tiene un rol fundamental en este asunto. Por ejemplo, muchos animales aprenden a caminar muy rápidamente una vez

[3] https://github.com/robotology
[4] https://www.coppeliarobotics.com/
[5] https://mujoco.org/
[6] https://gazebosim.org/home

han nacido sin una señal explícita de recompensa. Profundizando aún más en esta temática, es coherente pensar que desde un principio de los tiempos no hubo una señal de recompensa sobre los incontables comportamientos distintos que presentan las millones de especies a lo largo del mundo, esto puede explicarse por la capacidad exploratoria que han tenido los agentes biológicos durante la ejecución del complejo algoritmo de la selección natural.

En este trabajo se quiere explorar la capacidad de entrenamiento de un algoritmo que busque mecanismos de plasticidad que permitan a un agente adaptarse durante su tiempo de vida. En particular se utiliza el método de *Meta-Learning* a través de aprendizaje de reglas de aprendizaje Hebbianas en redes aleatorias de Najarro et al. [1], que muestra las capacidades de resolución en problemas típicos de tareas de aprendizaje reforzado, con la particular adaptación a situaciones no vistas durante el entrenamiento de forma orgánica gracias a la modulación dinámica de los pesos de la red durante la ejecución de la tarea. A continuación se hace un desglose de conceptos del método para entender la esencia del algoritmo:

- **Meta-Learning** o meta-aprendizaje corresponde al paradigma de *learning-to-learn*, cuyo objetivo es crear agentes que puedan aprender rápidamente a partir de la experiencia. Algunas de las formas que se ha abordado la resolución de este problema consiste en entregar explícitamente a la red la recompensa de la tarea [25] o realizar un pre-entrenamiento para posteriormente evaluar la habilidad de adaptarse a una nuevo problema [26].

- **Hebbian plasticity** o plasticidad Hebbiana está basada directamente en la ley de Hebb, propuesta por Donald O. Hebb [27] donde afirma que las conexiones sinápticas se fortalecen cuando dos o más neuronas se activan de forma contigua. Al asociarse el disparo de la célula presináptica con la actividad de la postsináptica tienen lugar cambios estructurales que favorecen la aparición de asambleas o redes neuronales. Englobando en la siguiente frase: *"Las células que se disparan juntas, permanecerán conectadas"*.

- El concepto de **Redes aleatorias** se relaciona a la capacidad de auto-organización que poseen los sistemas naturales [28, 29]. En el contexto de este trabajo se buscan mecanismos de plasticidad que permitan a los agentes adaptarse durante su *tiempo de vida* y que a partir de las reglas de aprendizaje Hebbiano,

sea capaz de auto-organizar una red inicialmente aleatoria para la resolución de una tarea así como, comprobar la capacidad de adaptarse a condiciones no vistas durante el entrenamiento. El trabajo de Mordvintsev [30] presenta un paralelismo interesante al explorar el crecimiento de autómatas celulares a través de reglas locales codificadas por una red neuronal, generando imágenes en 2D a partir de la auto-organización de los organismos.

Capítulo 4

Método propuesto

En este capítulo se presenta el método propuesto para integrar un algoritmo de caja negra con mecanismos de plasticidad Hebbiana y modelos de superficie basados en procesos gaussianos (GP), orientado al desarrollo de un sistema de autoreconocimiento corporal en un robot humanoide. El método se estructura en dos componentes principales: (i) una red neuronal con plasticidad Hebbiana evolutiva, encargada de generar las acciones motoras a partir de las activaciones táctiles y la pose del efector, y (ii) un modelo de superficie explícito mediante GP, que actúa exclusivamente como estimador probabilístico de la forma del torso y como sistema de recompensa intrínseca, cuantificando la reducción de incertidumbre en el modelo conforme el robot explora. En este esquema, los GP no generan acciones ni determinan poses de contacto, sino que informan al agente sobre qué regiones del cuerpo permanecen desconocidas en forma de recompensa al final de cada episodio, incentivando la reproducción de comportamientos exploratorios efectivos en generaciones posteriores. La Figura 4.1 ilustra el flujo general del método propuesto.

A continuación, se detalla el proceso completo: primero se describe la evolución de las reglas locales de aprendizaje en la red neuronal, que permiten la generación dinámica de pesos sinápticos y la adaptación del comportamiento motor; luego se aborda el uso de procesos gaussianos como estimadores de superficie, junto con su integración en el sistema de recompensas mixtas (externa e intrínseca). Finalmente, se presenta el algoritmo unificado y su representación esquemática, donde se ilustra el flujo general del método, desde la generación de acciones hasta la retroalimentación del modelo de superficie.

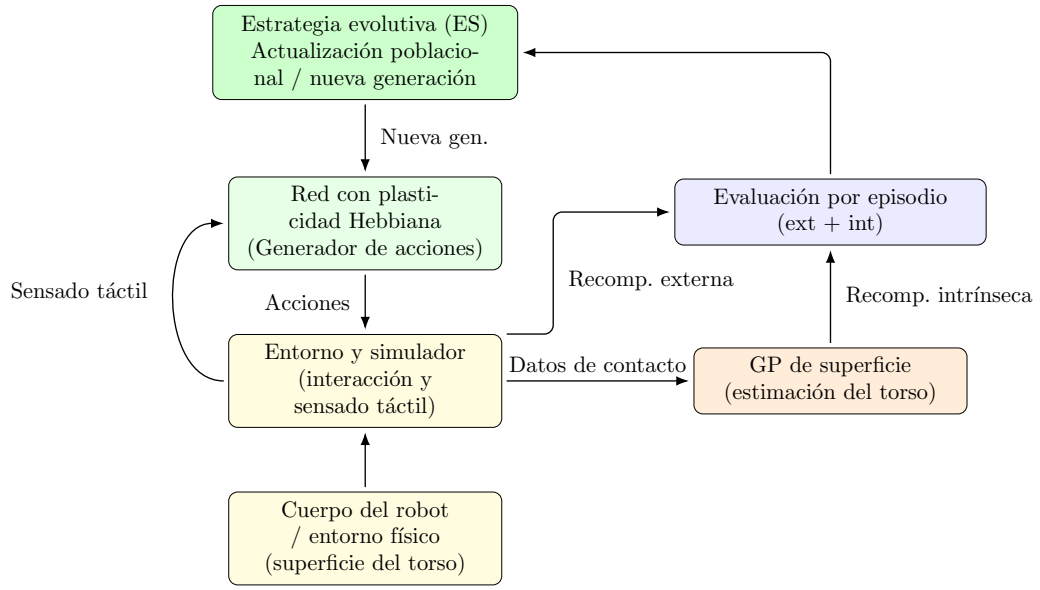


FIGURA 4.1: Esquema general del método propuesto. La red con plasticidad Hebbiana genera acciones; el entorno devuelve datos de contacto que alimentan un GP de superficie. Al final del episodio se combinan la recompensa externa y la intrínseca (del GP) en una evaluación; ES usa ese retorno para actualizar la población y producir la nueva generación de redes.

4.1. Evolución de reglas locales de aprendizaje

Los pasos principales del algoritmo pueden resumirse de la siguiente forma:

1. Se inicializa una población de redes neuronales con reglas de aprendizaje sináptico aleatorias.
2. Cada conexión es inicializada con pesos aleatorios y se evalúa su desempeño basado en la recompensa acumulada de la tarea.
3. Una nueva población es creada a partir de una **estrategia evolutiva** perturbando los parámetros de las reglas de aprendizaje en la dirección de una mayor recompensa acumulada.
4. Se vuelve al paso (2) con el objetivo de encontrar reglas más eficientes.

En detalle las reglas locales de aprendizaje de este método están inspiradas en mecánismos Hebbianos biológicos. Específicamente se usa un modelo Hebbiano ABCD

generalizado [31] para controlar la variación de los pesos de las conexiones entre las neuronas de una red *feedforward* de capas densas interconectadas. Los pesos inicializados de forma aleatoria son actualizados según:

$$\Delta w_{ij} = \eta_w \cdot (A_w o_i o_j + B_w o_i + C_w o_j + D_w), \tag{4.1}$$

donde w_{ij} es el peso entre la neurona i y j, η_w un parámetro de ritmo de aprendizaje local evolucionado, A_w parámetro de correlación evolucionado de actividad pre y post sináptica, B_w parámetro evolucionado de señal pre-sináptica, C_w parámetro evolucionado de señal post-sináptica, con o_i y o_j las activaciones pre-sinpáticas y post-sinpápticas respectivamente. D_w puede ser interpretado como un *bias* inhibitorio/excitatorio local de cada conexión en la red. Estas reglas de aprendizaje permiten a cada conexión en la red tener una propia regla de aprendizaje además de un ritmo de aprendizaje local.

4.1.1. Evolution Strategies

En este trabajo se adopta una variante simple de Evolution Strategies (ES)[2] como método de optimización de caja negra para ajustar los coeficientes de aprendizaje Hebbianos ($\mathbf{h}$) de un generador de acciones ($A_{\mathbf{h}}$). ES no requiere *backpropagation*; en su lugar, en cada generación se construye una población de candidatos perturbando la solución actual ($\mathbf{h}_t$) con ruido gaussiano de varianza controlada por (σ). Cada candidato se evalúa ejecutando una rutina de exploración en el simulador, donde el generador ($A_{\mathbf{h}}$) recibe como entrada las activaciones táctiles de los campos receptivos del torso ($\mathcal{S}$) y la pose del efector ($x_t \in \mathbb{R}^7$), y produce un paso cartesiano (Δx_t). La recompensa externa agrega, por episodio, los *taxels* nuevos descubiertos; con ello se obtiene un retorno (F_i) por candidato. Finalmente, los coeficientes se actualizan desplazándose hacia la dirección informada por la población (promedio ponderado) con tamaño de paso (α), según la ecuación 4.2 que se encuentra descrita al final del algoritmo.

Los pesos de las conexiones $\boldsymbol{w}$ y los coeficientes hebbianos $\mathbf{h}$ son inicializados a partir de una distribución uniforme según $\boldsymbol{w} \in U[-0.1, 0.1]$ y $\mathbf{h} \in U[0, 1]$.

Algorithm 1: Evolution Strategies (ES) (sin GP)

Input: Tamaño de paso α; desviación del ruido σ; población n; condición de continuación del episodio J; **campos receptivos del torso** $\mathcal{S}$; **pose inicial del efector** $x_0 \in \mathbb{R}^7$ (posición $p \in \mathbb{R}^3$ + orientación $q \in \mathbb{R}^4$); **parámetros iniciales** de la red h_0.

Output: Parámetros evolucionados $\mathrm{h}^\star$ de un **generador de acciones** $A_{\theta^\star} : (s_t, x_t) \mapsto \Delta x_t$.

Definiciones. A_θ es una red con plasticidad que, dada la lectura táctil s_t de $\mathcal{S}$ y la pose x_t, produce un paso cartesiano Δx_t.

La **recompensa externa** por paso r_t^{ext} se basa en *nuevos taxels* descubiertos.

Function `RunEpisode(`θ`):`

$x \leftarrow x_0$; $R \leftarrow 0$

while J **do**

$s_t \leftarrow$ leer activaciones táctiles desde $\mathcal{S}$

$\Delta x_t \leftarrow A_\theta(s_t, x)$

$x \leftarrow \mathrm{CartesianInterface}(x, \Delta x_t, \mathcal{W})$

$r_t^{\text{ext}} \leftarrow$ nuevos *taxels* descubiertos

$R \leftarrow R + r_t^{\text{ext}}$

end

return R

for $t = 0, 1, \ldots$ **do**

Samplear $\epsilon_1, \ldots \epsilon_n \sim \mathcal{N}(0, I)$

Cálculo de recompensa $F_i = F_i(\mathrm{h}_t + \sigma\epsilon_i)$ for $i = 1, \ldots, n$

Actualización $\mathrm{h}_{t+1} \leftarrow \mathrm{h}_t + \alpha \frac{1}{n\sigma} \sum_{i=1}^{n} F_i \epsilon_i$

end

El algoritmo indica que para cada generación se perturba la actual mejor solución descrita por $\mathbf{h}_t$ con ruido gaussiano $\epsilon_i = \mathcal{N}(0, 1)$ y se calcula el desempeño F_i de una población de n soluciones candidatas. Posteriormente se actualizan los coeficientes Hebbianos según la ecuación 4.2 que se encuentra descrita al final del algoritmo. Se aplica un condicionamiento J a los episodios que implicará los criterios de parada o por cuánto se ejecuta el experimento, por lo que mientras se cumpla dicha condición, la rutina exploratoria no será interrumpida.

El algoritmo adaptado es el que se indica en el Algoritmo 1.

$$\mathrm{h}_{t+1} \leftarrow \mathrm{h}_t + \alpha \frac{1}{n\sigma} \sum_{i=1}^{n} F_i \epsilon_i \tag{4.2}$$

Los pesos sinápticos ($\boldsymbol{w}$) y los coeficientes hebbianos ($\mathbf{h}$) se inicializan de manera uniforme como ($\boldsymbol{w} \sim U[-0.1, 0.1]$) y ($\mathbf{h} \sim U[0, 1]$). Nótese que el agente no observa la recompensa durante la ejecución paso a paso: el retorno se computa al nivel de episodio, consistente con el paradigma de caja negra.

4.2. Modelos de superficie mediante Procesos Gaussianos

La estimación de formas de objetos desconocidos en el espacio de trabajo del robot, es esencial para la interacción con su entorno. En un problema de manipulación, un robot debe ser capaz de realizar interacciones sofisticadas para llevar a cabo, por ejemplo, una tarea de agarre y mover un objeto en el espacio de forma precisa. Comúnmente, el robot cuenta con sensores basados en visión para estimar posición y formas de objetos, considerando las limitaciones como las oclusiones y las condiciones de iluminación variables hacen necesaria la inclusión de modalidades complementarias. La sensorización táctil, que proporciona datos como la fuerza de contacto y la distribución de la presión, ofrece un alternativa muy conveniente en la mejora de la estimación de formas. Sin embargo, incorporar eficazmente estos datos requiere un marco capaz de manejar la incertidumbre e integrar eficientemente nueva información.

La estimación de la forma mediante información táctil, ya sea fusionando información provista desde otras fuentes o sin información previa, requerirá de una tarea de exploración activa del actuador sobre el objeto. Solo una pequeña parte de la superficie del objeto puede ser tocada a la vez, teniendo que desplazar el actuador sobre distintas posiciones de la superficie para recuperar su forma.

Los procesos Gaussianos (GP) presentan una solución *adhoc* al problema de estimación de formas con sensores táctiles mediante exploración. Los GP son modelos Bayesianos no paramétricos que representan una distribución sobre funciones, caracterizados por una función media y una función de covarianza o *kernel* que define la similitud esperada entre salidas para entradas similares. Esta capacidad inherente para capturar la incertidumbre hace que los GP sean ideales para manejar datos ruidosos e incompletos, prevalentes en las interacciones robot y objeto del mundo real a través de sensores táctiles.

La estimación de superficies mediante GP se basa en una representación probabilística de una superficie geométrica. Se han utilizado en experimentos de exploración activa mediante modelos de superficies implícitas [13, 15], que se encuentran descritas en esencia por una función de forma

$$f : \mathbb{R}^3 \to \mathbb{R}; f(x) \begin{cases} = 0, & x \text{ está en la superficie} \\ > 0, & x \text{ está fuera de la superficie} \\ < 0, & x \text{ está dentro del objeto} \end{cases}$$

permitiendo representar eficientemente formas complejas con una única función. Sin embargo la complejidad de estos modelos radica en la extracción de información implícita del objeto, como las normales de la superficie o la curvatura.

Con estas complicaciones en mente, en este trabajo se optó por utilizar modelos de superficies explícitas, donde cada elemento de su representación define directamente la ubicación de la superficie. Zhengkun [14] propone un modelo explícito donde la superficie toma la forma $y_n = f(x_n, z_n)$, con (x_n, y_n, z_n) la coordenada del punto de contacto, $\mathbf{x}_n$, en el plano de referencia global. Luego el GP se especifica mediante la media $\mu(\tilde{\mathbf{x}})$ y la función de covarianza o *kernel* $k(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)$, donde $\tilde{\mathbf{x}}_i = (x_i, z_i)$. Se utiliza un *kernel* de base radial o cuadrado-exponencial, de forma

$$k(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j) = \sigma_f^2 e^{-\frac{1}{2}(\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j)^T \Lambda^{-1} (\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j)} + \sigma_w^2 \delta_{ij}$$

Con $\Lambda = \beta \mathbf{I}$ donde β es un fator de escalamiento del proceso y δ_{ij} es el delta de Kronecker. La media de la distribución inicial $\mu(\tilde{\mathbf{x}})$ se especifica como $\mu(\tilde{\mathbf{x}}) = 0$. Luego se define un set de entrenamiento $\mathbf{T} = [(\tilde{\mathbf{x}}_1, y_1), ..., (\tilde{\mathbf{x}}_N, y_N)]$. Para una nueva observación $\tilde{\mathbf{x}}_*$, la predicción de la distribución de un GP es una Gaussiana de media $\mu(\tilde{\mathbf{x}}_*)$ y de varianza $\sigma^2(\mu(\tilde{\mathbf{x}}_*))$, de forma:

$$\mu(\tilde{\mathbf{x}}_*) = \mathbf{k}_*^T \mathbf{K}^{-1} \mathbf{y}, \sigma^2(\mu(\tilde{\mathbf{x}}_*)) = k_{**} - \mathbf{k}_*^T \mathbf{K}^{-1} \mathbf{k}_*,$$

donde $\mathbf{k}_*$ es un vector con N entradas $k(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_*)$, $\mathbf{K}$ es una matriz con $\mathbf{K}_{ij} = k(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)$ y $k_{**} = k(\tilde{\mathbf{x}}_*, \tilde{\mathbf{x}}_*)$. La media posterior define una estimación de la forma del objeto y la varianza posterior define la incertidumbre con respecto al área no explorada.

La estrategia que se emplea en este trabajo al utilizar GP, consiste en incorporar un sistema de recompensa intrínseco que dependa de la disminución de la varianza del GP. Para el caso del modelo de superficie explícita, con $\tilde{\mathbf{x}}$ teniendo cotas establecida y sin observaciones iniciales, se tiene que la incertidumbre del modelo es máxima en cualquier punto de los límites. Por lo que en la medida que el área explorada se incremente, la varianza del modelo disminuirá pudiendo generar una recompensa que dependa directamente del incremento de conocimiento que existe de la superficie misma.

Un agente robótico que utiliza algoritmos de aprendizaje para la resolución de una tarea, a partir un sistema de recompensa basado en el conocimiento que va adquiriendo de la solución, puede entenderse como una rectificación intrínseca al problema y no generada por un observador externo.

Pathak [11] muestra que el agente, a partir del conocimiento que adquiere del ambiente, posee un sistema de recompensas que está sujeto a la capacidad de predicción de su propio estado siguiente a partir de su estado actual y la acción realizada. Luego el error de predicción de estado es utilizado como un sistema de señal de recompensa, que es intrínseco al problema y no definido de forma externa por el usuario del experimento. De esta forma la recompensa final será la suma ponderada de la recompensa externa y la interna, metodología cuyos resultados presentan una importante mejoría en tareas de exploración utilizando videojuegos como plataforma de pruebas.

Bajo la premisa anterior, en este trabajo se asocia la disminución de incertidumbre en el modelo de superficie del pecho del robot, como señal de recompensa intrínseca o interna, con el fin de incentivar al robot a buscar puntos táctiles sobre regiones que aún no han sido debidamente exploradas. De esta forma se incorpora

$$F^i = \gamma \frac{(\sigma_0^2 - \sigma_t^2)^2}{\sigma_0^2} \tag{4.3}$$

como definición de la recompensa interna asociada a la disminución de la varianza en el área definida del modelo desde el comienzo del experimento, al instante t que es evaluado al final cada iteración. Esta suposición está sujeta a que no hay datos de entrenamiento al comienzo de cada iteración, debido a que el robot aún no ha explorado ni una zona. Por tanto, al no tener datos para hacer un ajuste al modelo, la varianza será máxima en toda la región delimitada para la superficie objetivo.

Algorithm 2: ES adaptado con recompensa intrínseca basada en GP para control cartesiano Δx

Entradas: Tamaño de paso α; desviación del ruido σ; población n; condicionamiento episodio J; campos receptivos del torso $\mathcal{S}$; pose inicial del efector $x_0 \in \mathbb{R}^7$; coeficientes hebbianos iniciales $\mathbf{h}_0$; *kernel* del GP $k(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j)$ y sus hiperparámetros; límites de $\tilde{\mathbf{x}}$ para estimación explícita de superficie; ponderador γ para la recompensa intrínseca. No existen puntos iniciales en cada conjunto de entrenamiento $\mathbf{T}_i$.

Salidas: Coeficientes evolucionados $\mathbf{h}^\star$ y generador de acciones $A_{\mathbf{h}^\star} : (s_t, x_t) \mapsto \Delta x_t$.

Definiciones: Para el candidato i, $\mathbf{T}_i$ es el conjunto de entrenamiento del GP. Sea σ_0^2 la varianza posterior promedio inicial del GP (máxima sin datos) y σ_T^2 la varianza promedio al final del episodio. La recompensa externa total del episodio es $F_i^e = \sum_\tau \texttt{taxels\_nuevos}(\tau)$ y la recompensa intrínseca es

$$F_i^i = \gamma \, \frac{(\sigma_0^2 - \sigma_T^2)^2}{\sigma_0^2}.$$

El retorno del candidato es $F_i = F_i^e + F_i^i$.

for $t = 0, 1, \ldots$ **do**
- Muestrear $\epsilon_1, \ldots, \epsilon_n \sim \mathcal{N}(0, I)$
- **for** $i = 1$ **to** n **do**
  - `// Rutina de exploración + ajuste del GP para el candidato` i
  - Fijar $\mathbf{h}^{(i)} \leftarrow \mathbf{h}_t + \sigma \, \epsilon_i$; $\mathbf{T}_i \leftarrow \emptyset$; inicializar GP en $\mathcal{G}$
  - $x \leftarrow x_0$; $F_i^e \leftarrow 0$; calcular σ_0^2 en $\mathcal{G}$
  - **while** J **do**
    - $s_\tau \leftarrow$ leer activaciones táctiles desde $\mathcal{S}$
    - $\Delta x_\tau \leftarrow A_{\mathbf{h}^{(i)}}(s_\tau, x)$
    - $x \leftarrow \text{CartesianInterface}(x, \Delta x_\tau, \mathcal{W})$
    - Obtener la pose de contacto $\tilde{\mathbf{x}}_\tau$ y la medición y_τ;
    - $\mathbf{T}_i \leftarrow \mathbf{T}_i \cup \{(\tilde{\mathbf{x}}_\tau, y_\tau)\}$
    - Ajustar el GP con $\mathbf{T}_i$; $F_i^e \leftarrow F_i^e + \texttt{taxels\_nuevos}(\tau)$
  - **end**
  - Calcular σ_T^2 en $\mathcal{G}$; $F_i^i \leftarrow \gamma \dfrac{(\sigma_0^2 - \sigma_T^2)^2}{\sigma_0^2}$
  - $F_i \leftarrow F_i^e + F_i^i$
- **end**
- `// Actualización ES`
- $\mathbf{h}_{t+1} \leftarrow \mathbf{h}_t + \alpha \dfrac{1}{n\sigma} \sum_{i=1}^n F_i \, \epsilon_i$

end

El Algoritmo 2 muestra la adaptación del esquema ES cuando se incorpora una recompensa intrínseca proveniente del GP; la Fig. 4.2 sintetiza el flujo completo. (i) En el *bucle externo* (bloques verdes) ES parte de los coeficientes actuales $\mathbf{h}_t$, muestrea candidatos $\mathbf{h}^{(i)} = \mathbf{h}_t + \sigma\epsilon_i$ y, tras evaluar su retorno, actualiza $\mathbf{h}$ según (ecuación 4.2). (ii) En el *bucle interno del episodio* (bloques blancos), para cada candidato $\mathbf{h}^{(i)}$, el generador de acciones $A_{\mathbf{h}^{(i)}}$ recibe las activaciones táctiles s_t y la pose cartesiana x_t, y produce la acción Δx_t; la interfaz cartesiana aplica la acción y entrega la nueva pose x_{t+1}. Con la pose de contacto $\tilde{\mathbf{x}}_t$ y la medida y_t se actualiza el GP de superficie explícita. (iii) Al finalizar el episodio se computa el retorno del candidato $F_i = F_i^{\mathrm{e}} + F_i^{\mathrm{i}}$: la recompensa externa F_i^{e} acumula `taxels_nuevos`, mientras que la recompensa intrínseca F_i^{i} depende de la reducción de incertidumbre del GP, conforme a (ecuación 4.3). (iv) Finalmente, ES promedia la información de la población y actualiza $\mathbf{h}_{t+1}$ con (ecuación 4.2). En resumen: las acciones las genera la red $A_{\mathbf{h}}$; el GP no genera poses ni acciones, sino que estima la superficie y aporta la señal de recompensa intrínseca al final del episodio; las poses de contacto provienen de la ejecución cartesiana de Δx_t durante la interacción.

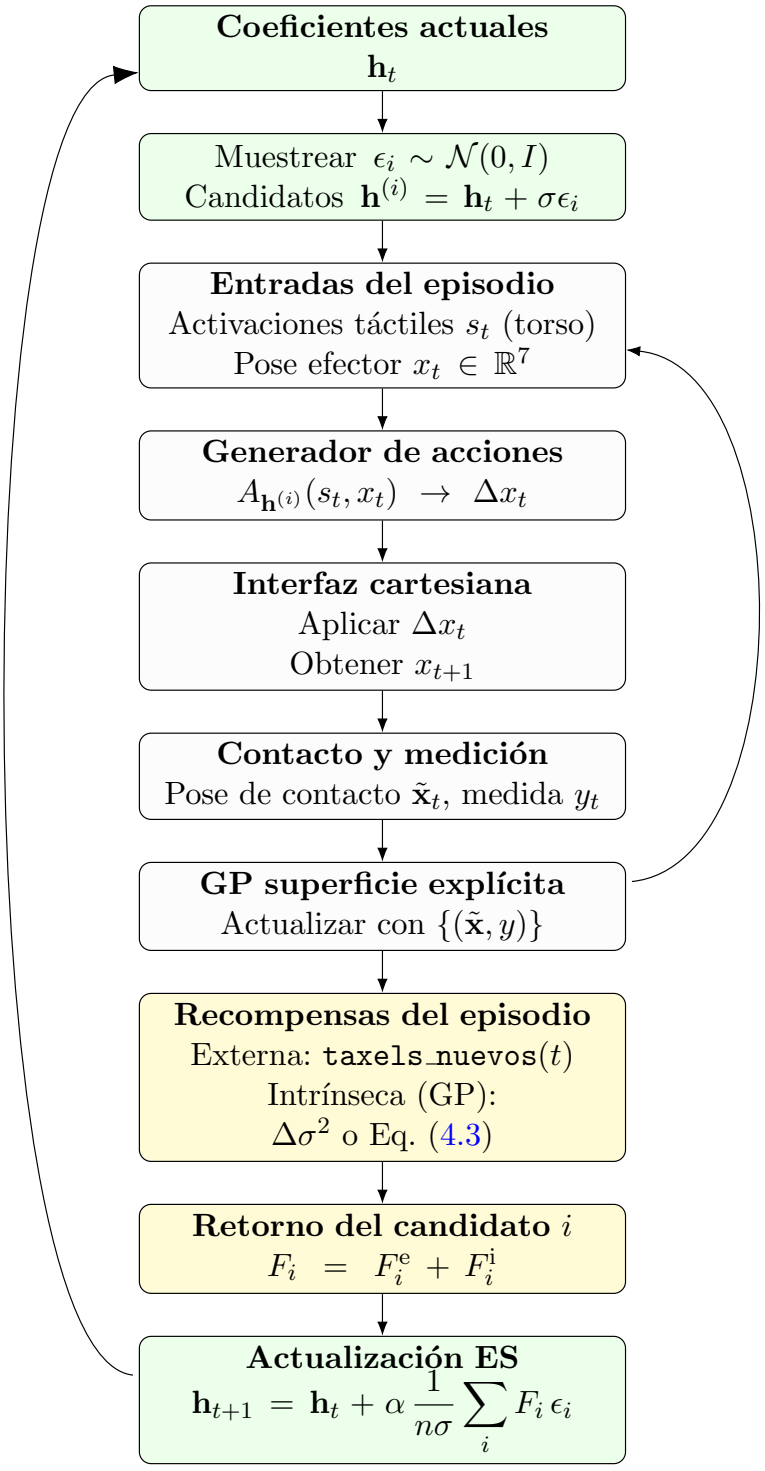


FIGURA 4.2: Diagrama de bloques del método propuesto. Verde: *bucle externo* ES (candidatos y actualización de $\mathbf{h}$). Blanco: *bucle interno* del episodio (sensado táctil, generador $A_{\mathbf{h}}$, aplicación cartesiana de Δx_t, contacto, actualización del GP. Amarillo: cómputo de recompensas externa e intrínseca para obtener F_i).

Capítulo 5

Diseño experimental

5.1. Configuración ambientes

El experimento planteado para este trabajo consiste en utilizar al robot para que descubra un modelo de superficie de su propio torso mediante la exploración táctil. Se usa un modelo de aprendizaje reforzado basado en plasticidad Hebbiana y un método de motivación intrínseca, para verificar su efecto en la capacidad exploratoria de la propuesta. Realizar este entrenamiento en el iCub real es arriesgado, debido a la fragilidad de algunas partes del robot, específicamente las manos que son el efector principal de las rutinas a ejecutar. Debido a esto, se propone utilizar un entorno simulado que pueda conectarse con el ecosistema de software del iCub.

La librería YARP permite la comunicación entre el ordenador y el robot, permitiendo la transferencia de datos y la ejecución de comandos. YARP funciona como un árbitro de comunicación, permitiendo la flujo de información entre diferentes módulos físicos que se encuentran en el robot, considerando que hay controladores, sensores y actuadores distribuidos en el robot descentralizando el procesamiento.

Debido a que el iCub es un robot altamente complejo y es riesgoso someterlo a un proceso de entrenamiento extensivo que pueda dañar el robot, se decidió realizar un entrenamiento virtual en el simulador Gazebo. El simulador Gazebo es un entorno de simulación 3D que permite la simulación de robots y ambientes. Gazebo es un software de código abierto que permite la simulación de robots en un entorno virtual, permitiendo la simulación de sensores y actuadores.

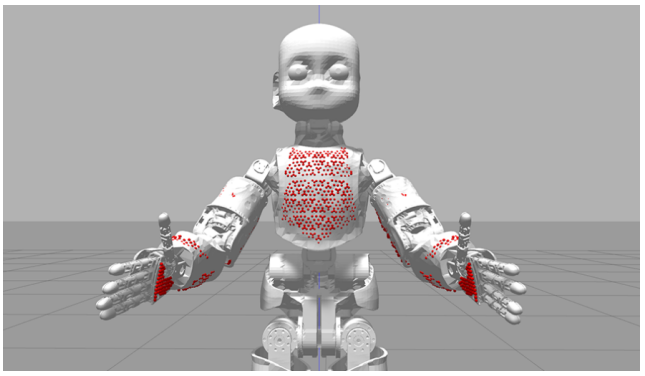


FIGURA 5.1: iCub en simulador Gazebo. El modelo 3D posee un sistema de *taxels* (puntos de color rojo) artificiales para los experimentos basados en estimulación táctil.

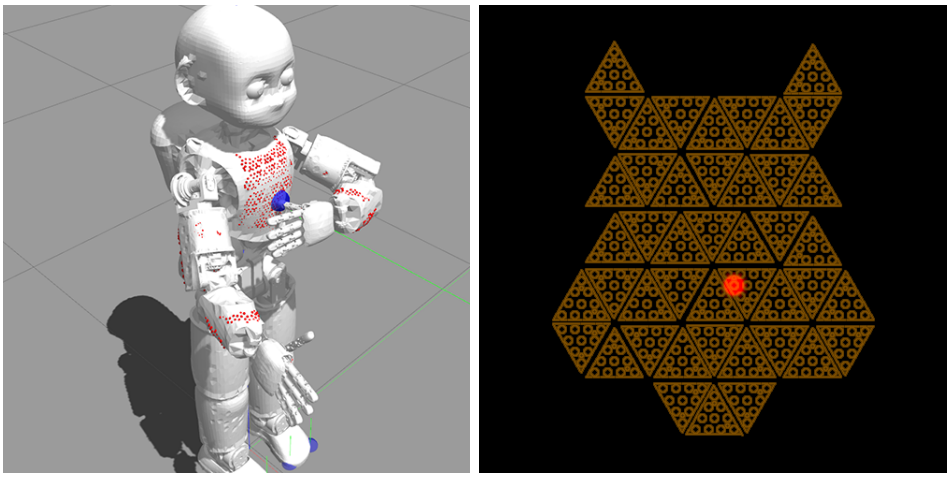
El ecosistema de software de iCub cuenta con una interfaz de comunicación entre el simulador Gazebo y el robot iCub a través de YARP. Esta interfaz permite la simulación de los sensores y actuadores del robot iCub en el simulador Gazebo, operando sobre las unidades de hardware virtual de la misma forma que el robot real. De esta forma, una vez se obtenga un modelo que haya sido entrenado de forma virtual, el traslado al robot real consiste solo en seleccionar el robot objetivo adecuado como parámetro al momento de arrancar la rutina, considerando que algún tipo de ajuste o re-entrenamiento debe realizarse para incorporar el comportamiento físico y mecánico del iCub real [8, 18].

En la Figura 5.1 se muestra el modelo del robot 3D en el simulador Gazebo. En este ambiente se han implementado controladores para cada uno de las articulaciones y movimientos del robot, así como puntos de sensado táctil o *taxels*. Como ha sido mencionado, la conexión al robot en simulación debe realizarse mediante YARP, como si se tratase del robot real. Para esto se utiliza una librería desarrollada en el trabajo de [10], que permite conectarse al servidor de YARP utilizando Python como lenguaje de acceso a la interfaz. Además la librería implementa una primera versión de ambiente para el Gym de OpenAI[1], una plataforma estándar para la investigación en tareas de aprendizaje reforzado.

El espacio de acciones del ambiente está basado en posiciones angulares de las articulaciones del brazo del robot, con tres grados de libertad para el hombro, uno para el codo y dos para la muñeca, manteniendo los dedos fijos. Esta estrategia que

[1] https://openai.com/research/openai-gym-beta

está diseñada para establecer relaciones entre las activaciones de *taxels* y una configuración de ángulos en el brazo del robot. En este trabajo se buscó establecer una cadena de acciones que esté sujeta al desplazamiento continuo del actuador sobre el pecho, que es el dedo pulgar en una posición perpendicular a la mano. La Figura 5.2 muestra la configuración experimental al inicio de la ejecución de la rutina de exploración.



(A) Posición de partida del iCub en los experimentos de exploración.

(B) Módulo de visión mediante interfaz gráfica de los *taxels* en el torso. Las activaciones de los mismos están conectadas a los estímulos en el simulador.

FIGURA 5.2: Configuración experimental del iCub al inicio de la rutina de exploración, con representación gráfica de la activación del primer *taxel* en el torso, mediante visor gráfico `icubSkinGui`. Imágenes y preparación de ambiente de [10].

Para cumplir con el cambio de modelo de acciones, gran parte de este trabajo se basó en modificar la configuración inicial del robot virtual para habilitar el control del brazo directamente en el espacio operacional, en vez de una configuración de ángulos. Se busca que la punta del dedo alcance cierta pose en el espacio cartesiano, a partir de una combinación de un punto tridimensional que esté vinculado a este efector final. Para esto se utiliza el módulo *Cartesian Interface*, provisto en el ecosistema de software del iCub, que posee un optimizador y controlador interno para encontrar las soluciones adecuadas a la configuración de ángulos del brazo, respecto a la pose objetivo del efector. De esta forma se pretende realizar una exploración continua basada en el movimiento del dedo pulgar sobre el torso del robot, realizando pasos de desplazamiento respecto a la pose del efector durante el experimento. De esta

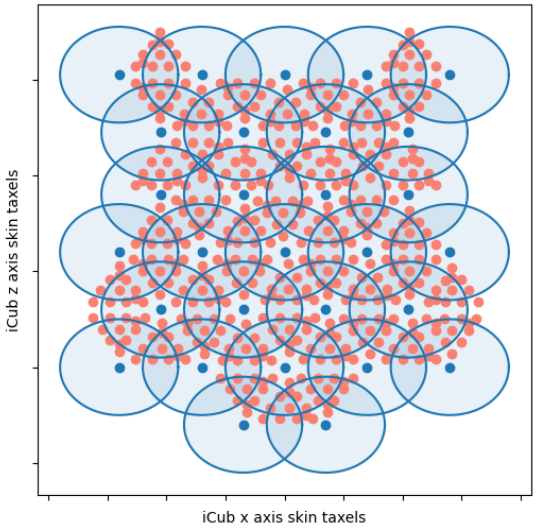


FIGURA 5.3: Sistema de *taxels* (puntos de color rojo) del torso y filtros Gaussianos (círculos azules) que simulan campos receptivos para la activación ponderada de señales de los *taxels*.

forma también será posible identificar el punto de contacto, pues la pose del efector es conocida gracias al controlador interno.

El iCub posee 440 *taxels* en su torso, que en su mayoría se mantendrán en un estado inactivo o de valor cero, y aquellos que se vean estimulados por el efector pasarán a estar activos o de valor uno. Dado que que el efector es la punta del dedo, en un mismo instante solo una pequeña fracción de *taxels* se encontrarán activos si el torso es alcanzado, por tanto tener un espacio escaso de esta magnitud para el algoritmo puede imposibilitar la convergencia del algoritmo de aprendizaje. Por este motivo se adopta un método de reducción de dimensionalidad de este espacio desde un punto de vista bioinspirado, al superponer 29 filtros Gaussianos sobre la región total de *taxels* en el torso, los que actuarán como campos receptivos generando una respuesta ponderada a la activación de los *taxels* estimulados. La Figura 5.3 muestra gráficamente los filtros Gaussianos superpuestos y el área que abarcan para su activación.

Tabla 5.1: Arquitectura y espacios del generador de acciones (véase sección 4.1).

Componente	**Tamaño**	**Descripción**
Capas densas intermedias	$64 \to 32$ nodos	Dos capas totalmente conectadas con modulación hebbiana.
Parámetros de pesos	4,576	Número total de pesos entrenables (sin contar coeficientes hebbianos).
Coeficientes hebbianos (ABCD)	22,880	Coeficientes dinámicos optimizados por ES bajo el mecanismo ABCD.
Espacio de observación	$\dim(u_t) = 36$	29 campos receptivos del torso + 7 componentes de pose del efector (posición y orientación).
Espacio de acciones	$\dim(\Delta x_t) = 7$	Paso cartesiano del efector en espacio 3D (posición/orientación).

Tabla 5.2: Hiperparámetros de entrenamiento (ES) y del modelo de superficies con GP (véase sección 4.2).

Parámetro	**Valor**	**Descripción**
Tamaño de población (ES)	$n = 25$	Redes candidatas por generación.
Generaciones (ES)	100	Iteraciones de evolución.
Tamaño de paso (ES)	$\alpha = 0.2$	Modulación de actualización sobre $\mathbf{h}$.
Desviación del ruido (ES)	$\sigma = 0.15$	Escala de perturbación para candidatos.
Varianza señal (RBF)	$\sigma_f^2 = 1$	Amplitud del kernel RBF del GP.
Varianza de ruido	$\sigma_w^2 = 10^{-4}$	Ruido gaussiano independiente del GP.
Escala espacial	$\beta = 10^{-3}$	Factor de longitud/escala en $\Lambda = \beta\mathbf{I}$.
Peso recompensa intrínseca	$\gamma = 20$	Ponderador en $F^i = \gamma \frac{(\sigma_0^2 - \sigma_t^2)^2}{\sigma_0^2}$.

5.2. Rutina de aprendizaje

Utilizando la librería mencionada en la sección 5.1 se instancia el ambiente de entrenamiento de forma directa y ágil, las modificaciones han sido implementadas en la fuente y se encuentran incluidas en la inicialización de la ambiente. El algoritmo de aprendizaje se instancia de forma paralela utilizando las observaciones y ejecución de acciones provistas en la interfaz del objeto ambiente, de esta forma los parámetros de los algoritmos se reconfiguran de forma independiente en el estudio del experimento.

Los hiperparámetros de las redes neuronales de pesos dinámicos descritas en la sección 4.1 están definidas en la Tabla 5.1. Luego los hiperparámetros del algoritmo ES y de los modelos de superficie con GP están definidos en la Tabla 5.2.

La elección de estos parámetros se fundamenta en pruebas empíricas de los resultados de estimación de superficies y la evolución de la varianza promedio en los modelos

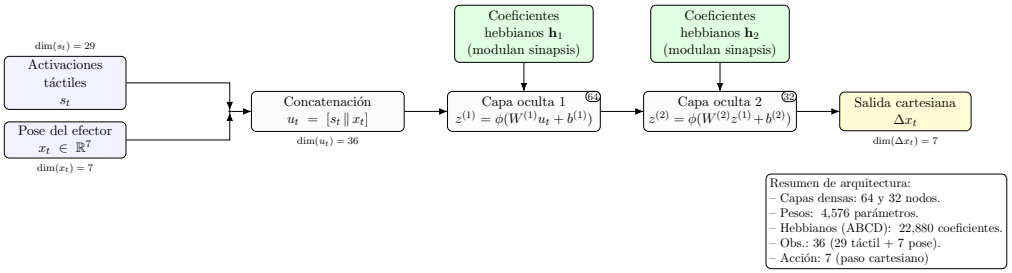


FIGURA 5.4: Red/generador de acciones $A_{\mathbf{h}}$, dos capas moduladas por $\mathbf{h}$, y salida cartesiana Δx_t.

obtenidos, de manera que en la medida que nuevos puntos se incorporen al GP ésta última disminuya y no ocurra *overfitting* en la estimación. Con esta configuración, la media obtenida del modelo mediante los GP tiene una curva suave en su superficie, similar a la del pecho del iCub.

Finalmente en la Figura 4.2 muestra de forma gráfica la configuración del generador de acciones o red neuronal, que ejecuta las acciones sobre el actuador que explora el torso del robot iCub.

Se ejecuta el servidor YARP y se lanza el simulador Gazebo con el modelo del iCub con *taxels* virtualizados. En esta etapa ya es posible visualizar las componentes del robot y sus puertos en la lista de conexiones disponibles, ejecutando los programas que conectan el ambiente y el Algoritmo 2 implementado para lanzar las rutinas de aprendizaje.

En este punto y desde la posición inicial presentada en la Figura 5.2a, el iCub aplicará los cambios de pose del dedo pulgar o actuador a partir de la salida del generador de acciones (ver Figura 4.2), ejecutando tantos steps como sean posibles según el condicionamiento del episodio y los criterios de parada. Estos últimos corresponden a:

- Presencia de colisiones entre partes del cuerpo del iCub que no sean el efector y el torso del robot.
- Ausencia de contacto entre el efector y algún sensor táctil del torso por más de **10 steps** en el episodio.

- Mientras exista contacto exploratorio del efector sobre el torso del robot, el experimento continuará y se reiniciará el criterio de parada por ausencia recurrente de tacto entre el efector y torso.

El desempeño o recompensa externa para el algoritmo a implementar se compone de la suma total de puntos que han sido alcanzados al menos una vez por efector durante la rutina, con $\mathrm{p}(t) = [p_0(t), p_1(t), ..., p_n(t)]$ un vector cuyo largo n es la cantidad de *taxels* disponibles en el torso, donde

$$p_k(t) \begin{cases} = 1, & \text{el taxel } k \text{ se fue alcanzado en la generación } t \\ = 0, & \text{e.o.c.} \end{cases}$$

Luego, según la definición del Algoritmo 2 indica que la recompensa total de un agente al final de la generación t se mide según la Ecuación 5.1, con la recompensa interna y externa se describen en la Ecuaciones 4.3 de la sección 4.2 y 5.2 respectivamente.

$$F(t) = F^e(t) + F^i(t) \tag{5.1}$$

$$F^e(t) = \sum_{k=0}^{n} p_k(t) \tag{5.2}$$

Para cada generación se ejecutará el episodio exploratorio para cada una de las redes neuronales de la población y se aplicarán las variaciones de coeficientes según el Algortimo 2 una vez .

5.3. Ejecución de experimentos

El ecosistema de software de iCub está implementado con el objetivo de actuar sobre el robot físico. La base temporal de los cálculos está rígidamente planteada en la definición de los modelos de control, imposibilitando la aceleración de los experimentos a partir de los relojes que actúan como base temporal para el servidor de YARP. Por este motivo y bajo la configuración experimental planteada, cada rutina

toma cerca de **2 días** en culminar completamente, donde hay una sola instancia de Gazebo disponible.

Para flexibilizar las rutinas de entrenamiento y crear instancias paralelas en un mismo equipo modularmente, se implementa un contenedor de Docker con todo el software necesario: el ecosistema de software de iCub[2], el algoritmo de aprendizaje adaptado para el problema[3] y la interfaz de comunicación entre el ambiente de aprendizaje y el servidor YARP[4]. Utilizando un servidor con 40 cores en CPU y una GPU Tesla P100 de Nvidia, se pueden instanciar 10 experimentos en paralelo para obtener más resultados durante el mismo tiempo de ejecución.

Todas las ejecuciones siguen exactamente el mismo protocolo (mismo entorno, límites del workspace, presupuesto de pasos por episodio, criterios de parada y métrica de retorno).

[2]https://github.com/robotology/robotology-superbuild
[3]https://github.com/pabloreyesrobles/HebbianMetaLearning/tree/icub-skin
[4]https://gitlab.com/pablo_rr/code-icub-gazebo-skin

Capítulo 6

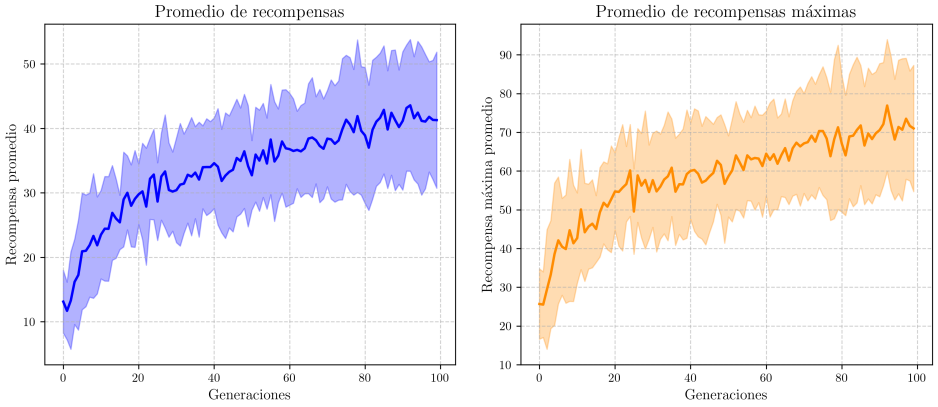
Resultados experimentales

En esta sección se presentan los hallazgos obtenidos a partir del desarrollo de la investigación. Tal como será detallado a continuación, la hipótesis planteada inicialmente no fue confirmada. Los resultados de los modelos de aprendizaje y datos recopilados proporcionan información importante para identificar las causas que inhiben una exploración extensa en la espacialidad de la región objetivo, junto con una evaluación de las herramientas utilizadas y cómo el software propio de la plataforma iCub influyó en los resultados. A partir de esto es posible proponer mejoras metodólogicas para investigaciones futuras en esta línea.

6.1. Resumen de experimentos

La Figura 6.1 exhibe el comportamiento de las recompensas totales a lo largo del entrenamiento. En el caso de la Figura 6.1a se muestran las recompensas promediadas de toda la población de redes neuronales actuando en el proceso de aprendizaje, mientras que la Figura 6.1b muestra el comportamiento de las mejores redes por generación. No se percibe una diferencia importante en la forma de crecimiento de las curvas de recompensas, pero sí hay una evidente mejoría cuantitativa en el caso de las recompensas máximas. Esta diferencia se plantea para establecer un criterio de selección de solución ante un posible uso real de los modelos resultantes y también para reflejar la naturaleza del algoritmo, donde las mejores soluciones son las que

dictan el proceso evolutivo y reconfiguración de las redes en la medida que avanzan las generaciones.



(A) Recompensas promedio de toda la población de redes en ejecución.

(B) Recompensas promediadas de las redes que exhibieron el mejor comportamiento por generación.

FIGURA 6.1: Recompensas totales generadas por el algoritmo de aprendizaje durante la exploración de la superficie táctil.

La Figura 6.2 muestra el comportamiento de la recompensa intrínseca promedio de toda la población de redes neuronales durante el entrenamiento. La forma de crecimiento es similar a la de las recompensas totales exhibidas en la Figura Figura 6.1, sin embargo se puede notar que a partir de la generación 30 aproximadamente no hay cambios importantes del punto de vista cuantitativo. Por otro lado, en términos de escala puede no ser significativo para la recompensa final, lo que tiene un efecto directo en el comportamiento evolutivo de los modelos, ya sea por un asunto netamente de magnitud que podría ser regulado a través del parámetro γ (ver ecuación 4.3), como que el comportamiento de dicha recompensa no tiene un sentido práctico para la solución.

Considerando que el torso dispone de 440 *taxels* y que la recompensa externa depende del número de *taxels* explorados, los valores finales de la Figura 6.1 indican que, bajo la configuración actual, el método propuesto **no supera el 20 % de cobertura** (esto es, $\lesssim 88$ *taxels*). Si bien la hipótesis de este trabajo no exige superar métodos alternativos, los propios resultados evidencian limitaciones para maximizar el área descubierta. Entre las causas plausibles se consideran:

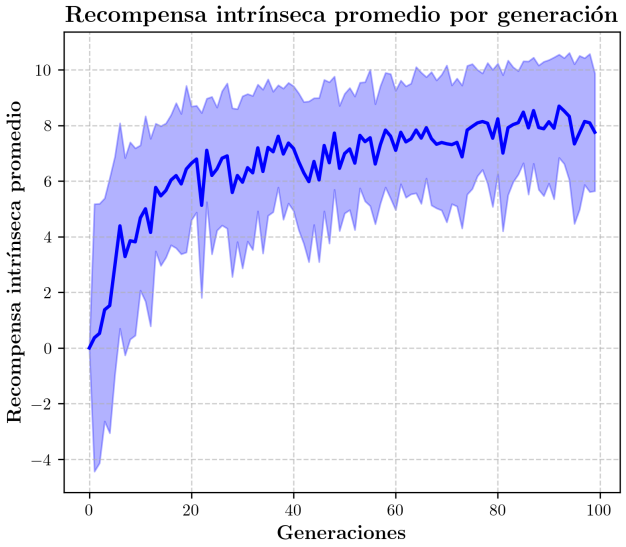


FIGURA 6.2: Recompensa intrínseca promedio de toda la población de redes neuronales del proceso de aprendizaje.

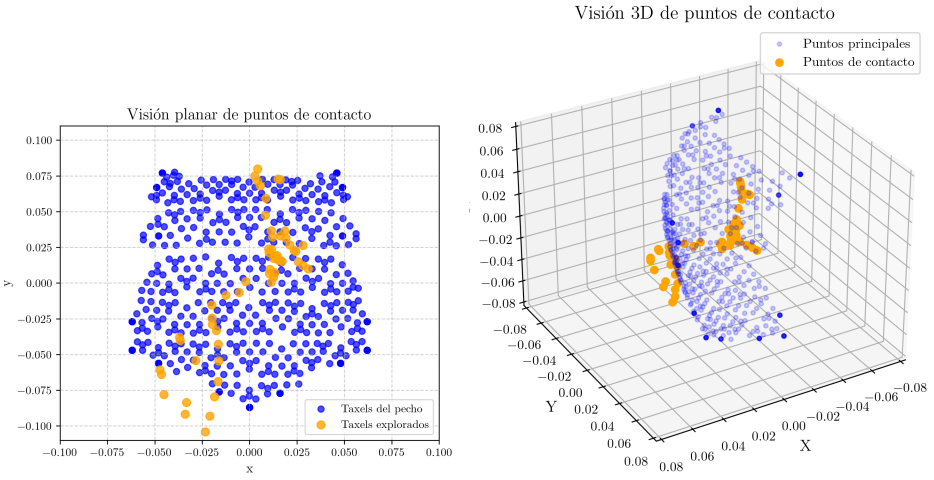
- **Exploración poco dirigida.** La exploración aleatoria y generación de movimientos continuos en el espacio de trabajo conlleva una alta probabilidad de infringir las condiciones del episodio. Relajar los criterios de parada (p. ej., aumentando su duración) no garantiza mantener el contacto efectivo ni una cobertura más amplia. Aunque las recompensas sugieren mayor tiempo de estimulación sobre el torso, la direccionalidad de la exploración sigue siendo débil y la cobertura espacial continúa siendo difícil de optimizar para este esquema.
- **Desajuste de hiperparámetros y arquitectura.** La combinación actual de hiperparámetros de ES y la configuración de la red (tamaños de capa, modulación hebbiana, escalas de entrada) puede no ser la idónea. La validación empírica necesaria para ajustar estos elementos se vio limitada por los altos tiempos de entrenamiento y las restricciones técnicas de la simulación, lo que probablemente impidió converger hacia una configuración más efectiva.

La disminución de la varianza del modelo de superficie es una idea atractiva para favorecer la identificación de la forma del torso del iCub, sin embargo, en la configuración actual esta señal se incorpora al aprendizaje únicamente de manera cuantitativa, sin codificar explícitamente su dimensionalidad espacial. A continuación se plantean dos extensiones que complementarían el método:

- **Recompensa intrínseca modulada por "curiosidad" espacial.** Ajustar dinámicamente la recompensa en función de la historia de exploración por regiones, asignando mayor valor a áreas *poco o nada muestreadas* a lo largo de las generaciones. En la práctica, puede implementarse como un factor de ponderación espacial $w(\tilde{\mathbf{x}})$ que crece donde la cobertura acumulada es baja (o donde la varianza local del GP se mantiene alta). Se prevé un impacto positivo tanto en la recompensa final como en la dinámica evolutiva, aunque esta extensión, si bien analizada, no fue implementada en este trabajo.
- **Entradas explícitas de cobertura al generador de acciones.** Incluir en el generador de acciones un mapa de regiones ya exploradas (p. ej., una codificación binaria o densa por parches del torso) además de las activaciones táctiles instantáneas. Esta señal contextual permitiría a la red distinguir entre zonas ya documentadas y aún por explorar, orientando los pasos cartesianos hacia áreas informativamente más útiles.

La Figura 6.3 presenta una dimensionalidad física del análisis previo de recompensas, mostrando puntos descubiertos durante una de las rutinas exploratorias con mayor recompensa, en una vista planar y tridimensional. Se observa que la cantidad de *taxels* o regiones descubiertas en la rutina es muy limitada a respecto del espacio completo disponible en el torso del robot. Se observaron discrepancias entre las posiciones de *taxels* explorados y sus ubicaciones teóricas. Este desajuste se atribuye a movimientos residuales del efector en el instante de captura de la pose, que pueden producir una colisión involuntaria de otra parte del efector con *taxels* adyacentes, desplazando así los puntos sensados. Este fenómeno fue considerado durante el diseño experimental y se concluyó que una solución directa, aunque no trivial, que consiste en agregar al dedo efector un *taxel* dedicado para asegurar el punto de contacto. En el robot físico iCub existen *taxels* de menor sensibilidad en la punta de los dedos, por lo que esta alternativa resulta factible en un escenario real.

Como fue presentado en la Sección 5.1, el ecosistema de software del iCub provee de un sistema complejo de control para la gran cantidad de componentes que lo integran, pero que se encuentra en constantes iteraciones y mejoras. En particular el sistema de sensores táctiles está en una etapa temprana de desarrollo e integración a los entornos simulados, lo que se tradujo en múltiples errores de ejecución, experimentos fallidos por errores fuera del control de usuario y discrepancias en el



(A) Vista planar de puntos explorados respecto a los *taxels* del torso del iCub.

(B) Vista tridimensional de puntos explorados respecto a los *taxels* del torso del iCub.

FIGURA 6.3: Vistas de puntos explorados de un modelo resultante.

rescate de información concluyente. En relación con lo anterior, la decisión de utilizar el módulo *Cartesian Interface* con un desplazamiento del punto de actuación a la punta del dedo pulgar izquierdo del iCub, presentó una interesante oportunidad de manipulación y control sobre los movimientos en las rutinas de exploración. Sin embargo, posterior a la obtención de resultados se identificó que las soluciones de posición presentadas por el controlador cartesiano, no respondían de forma coherente a todas las señales de control manual comandadas, lo que puede explicar la inhabilidad del robot de alcanzar algunas zonas de su torso mediante este método en las rutinas de exploración automáticas.

A pesar de la limitada información espacial en las rutinas exploratorias, el proceso Gaussiano en ejecución incorpora cada punto durante la ejecución para calcular la varianza del mismo, que disminuye con la incorporación de cada punto descubierto, reflejado en la recompensa interna mostrada en la Figura 6.2. La Figura 6.4 muestra un resultante de superficie a partir del proceso Gaussiano con los mismos puntos fueron explorados en el experimento de la Figura 6.3. Se observa como la superficie tipo hoja modifica su forma respecto a la distribución de puntos. A modo de ejercicio y para demostrar la implicancia del aumento de información se replica a modo de espejo los puntos desde una vista frontal, como se ve en la Figura 6.5 lo

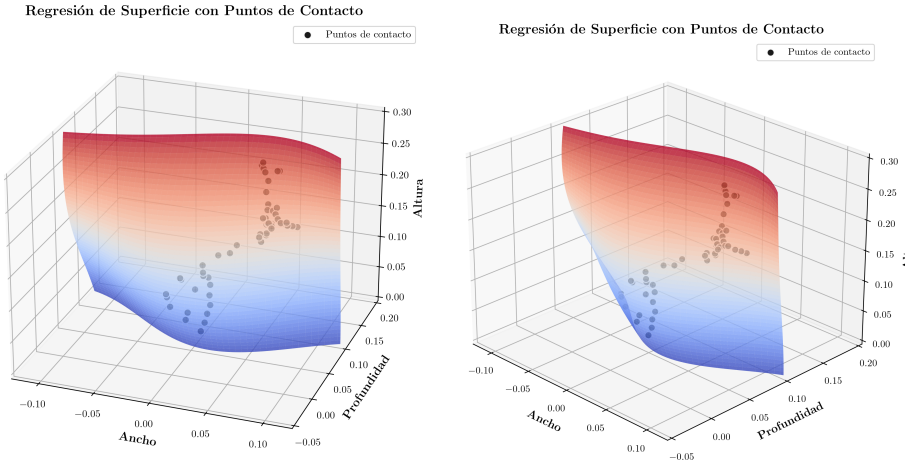


FIGURA 6.4: Superficie construida a partir del proceso Gaussiano y los puntos explorados.

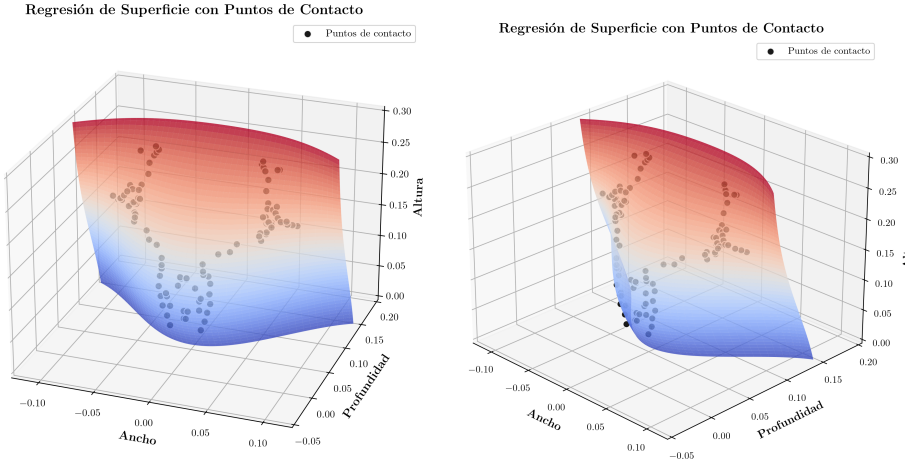


FIGURA 6.5: Superficie construida a partir del proceso Gaussiano y los puntos explorados, con una replicación frontal.

que modifica la construcción del modelo de superficie y permite tomar una forma más representativa a la información provista por el proceso exploratorio.

Capítulo 7

Discusión y propuestas de trabajo futuro

7.1. Discusión

Los resultados muestran que la solución algorítmica satisface parcialmente la hipótesis planteada: el enfoque es aplicable, pero exhibe limitaciones en la capacidad exploratoria final (coberturas inferiores al umbral deseado). Una elección más cuidadosa de hiperparámetros (tamaños de capas, escalas de entrada, reglas hebbianas y parámetros de ES), así como ajustes en la definición de la señal intrínseca, podría aportar mejoras. No obstante, tras un conjunto amplio de corridas y descartes de larga duración, la evidencia sugiere que el método requiere una revisión estructural.

En particular, se identificaron tres factores relevantes: (i) la recompensa intrínseca basada exclusivamente en disminución global de varianza es informativa pero espacialmente indiferenciada; (ii) la exploración carece de un sesgo explícito hacia zonas poco muestreadas; y (iii) existen efectos experimentales (contactos residuales y pequeños desajustes de pose) que introducen ruido en la señal de retorno. Bajo este diagnóstico, se valoran dos mejoras complementarias: incorporar modulación espacial en la recompensa intrínseca y añadir entradas explícitas de cobertura al generador de acciones, de modo que la red distinga entre áreas ya exploradas y pendientes.

La elección de reglas de plasticidad hebbianas se sustenta en su capacidad de adaptación a situaciones no vistas durante el entrenamiento. Como antecedente, en el Apéndice A se documenta un estudio no publicado con un cuadrúpedo (ArgoV2) cuyo desempeño inspiró el presente enfoque. Con todo, no es razonable responsabilizar en exclusiva al algoritmo de aprendizaje: la topología de aplicación y varias decisiones de integración adoptadas en este trabajo son perfectibles.

La elección de la plataforma iCub obedece a la adquisición del robot por parte de la universidad y al interés de impulsar un primer proyecto de investigación sobre la plataforma y al sistema de sensores táctiles que ésta ofrece. El software provisto por la comunidad iCub se encuentra en permanente evolución, por lo que es esperable que existan componentes en desarrollo activo y documentación en progreso.

Respecto del ecosistema iCub, se constató que *Cartesian Interface* no es necesariamente descartable, pero requiere un estudio específico de sus capacidades y soluciones para el experimento planteado. No existe documentación clara sobre si el *workspace* efectivo incluía el torso bajo la cinemática y restricciones configuradas, lo que pudo limitar trayectorias viables. De cara a nuevas corridas, se recomienda delimitar formalmente el volumen alcanzable (con la definición del *tool frame* en la punta del dedo empleado) y verificar numéricamente soluciones del solver cartesiano en la vecindad del torso.

Finalmente, se observaron discrepancias entre posiciones sensadas y teóricas de *taxels*, atribuibles a movimientos residuales del efector al momento de registrar la pose y a colisiones no deseadas de partes adyacentes del efector. Este fenómeno se mitigará con (i) ventanas cortas de estabilización antes de muestrear, (ii) descarte de contactos indeseados por reglas geométricas locales y (iii) la adición de un *taxel* dedicado en el dedo efector con, lo que hace factible asegurar el punto de contacto.

7.2. Propuestas de trabajo futuro

7.2.1. Exploración de hiperparámetros del algoritmo de plasticidad Hebbiana

Explorar la topología de la red y la sintonía de ES ofrece un espacio inmediato de mejora. Búsqueda de tamaños de capas/activaciones que reduzcan el acoplamiento

con los coeficientes hebbianos. La reducción de complejidad en la optimización hebbiana puede acelerar la adaptación generacional. A la luz del ejemplo del ArgoV2, el enfoque mantiene potencial de adaptación a largo plazo.

7.2.2. Modificación del sistema de recompensa interna

Se propone introducir **curiosidad espacial**: ponderar la recompensa intrínseca con un factor $w(\tilde{\mathbf{x}})$ mayor en regiones con baja cobertura acumulada o varianza local alta. Este sesgo puede añadir direccionalidad a la exploración y podría reducir la dependencia de la recompensa externa; incluso cabe considerar una condición sin recompensa externa si el término intrínseco induce el comportamiento deseado [11]. De forma complementaria, incorporar al generador un mapa compacto de cobertura como entrada adicional permitiría a la red diferenciar entre zonas ya documentadas y aún por explorar.

7.2.3. Incorporación de visión

El iCub dispone de cámaras estéreo que habilitan estimación tridimensional basada en *computer vision*. Aunque para el torso la visibilidad puede ser limitada, en otras zonas del cuerpo la fusión táctil–visual podría resultar valiosa para guiar la exploración y validar el mapeo de superficie. Existen antecedentes de uso de visión para estimar cinemática corporal [8] y de exploración basada en curiosidad [18], que, si bien difieren del presente enfoque, aportan lineamientos útiles.

7.2.4. Mitigación de artefactos de contacto y pose

Implementar una ventana de estabilización de pose previo a registrar contacto e instrumentar el dedo efector con un *taxel* dedicado.

Capítulo 8

Conclusión

En esta investigación, se abordó el problema del reconocimiento del modelo corporal en robots humanoides mediante el uso de estímulos táctiles auto-generados. Se propuso un enfoque basado en estrategias evolutivas y modelos de aprendizaje reforzado con mecanismos de plasticidad Hebbiana para optimizar la exploración del propio cuerpo en el robot iCub. Además, se implementó un modelo de regresión mediante Procesos Gaussianos con el objetivo de reconstruir una representación probabilística de la superficie explorada.

Los resultados obtenidos demostraron que, si bien la propuesta permitió una exploración parcial del torso del robot, el modelo de recompensa basado en la reducción de incertidumbre no logró incentivar suficientemente la cobertura total del área de exploración. Se identificaron limitaciones en la capacidad del algoritmo para generalizar la representación espacial de taxels no explorados, lo que sugiere la necesidad de incorporar mecanismos adicionales para mejorar la eficiencia exploratoria, como la integración de heurísticas espaciales o métodos de curiosidad intrínseca.

A pesar de estas limitaciones, los hallazgos de este estudio representan un avance en la utilización de exploración táctil en robots humanoides para la construcción de modelos corporales. En futuras investigaciones, se recomienda explorar ajustes en la arquitectura del modelo de aprendizaje, optimización de hiperparámetros y la integración de múltiples modalidades sensoriales, como *computer vision*, para complementar la percepción táctil.

Apéndice A

Aplicación de ES con plasticidad hebbiana en caminatas

Previo a este trabajo de tesis, para sustentar y decidirse por el método de plasticidad Hebbiana presentado por Najarro [1], se realizó un trabajo no publicado de autoría propia con dicho algoritmo en otro problema de simulación con plataformas robóticas. Se sometió la plataforma robótica ArgoV2 con 12 grados de libertad (ver Figura A.1) en el entorno de simulación CoppeliaSim [32], a una tarea de aprendizaje de caminatas, bajo la misma estructura del experimento realizado por Reyes [33]. Los resultados obtenidos con el método de plasticidad Hebbiana excedió los obtenidos en el trabajo citado, que utilizaba el método neuroevolutivo ES-HyperNEAT [34] como algoritmo de aprendizaje.

En la Figura A.2a se muestran los resultados de desempeño o recompensa de la plataforma robótica sujeta a un escenario de aprendizaje de caminatas a lo largo de 100 generaciones, donde mientras más alta la recompensa, mayor la distancia alcanzada por el ArgoV2 en el escenario. Se ve una saturación en el aprendizaje a lo largo del entrenamiento, donde se alcanza rápidamente un peak de recompensa en las primeras 20 generaciones. Por otra parte, en la Figura A.2b se muestran los resultados del mismo experimento, esta vez con el algoritmo de plasticidad Hebbiana usado en este trabajo. Esta vez se extendió el tiempo de entrenamiento al no obtener una saturación apreciable en el sistema de recompensas, en contraste con el lento desarrollo temprano evidente al compararlo con los métodos basados en HyperNEAT. Esto

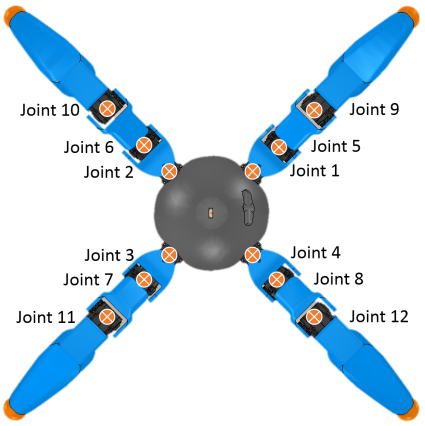
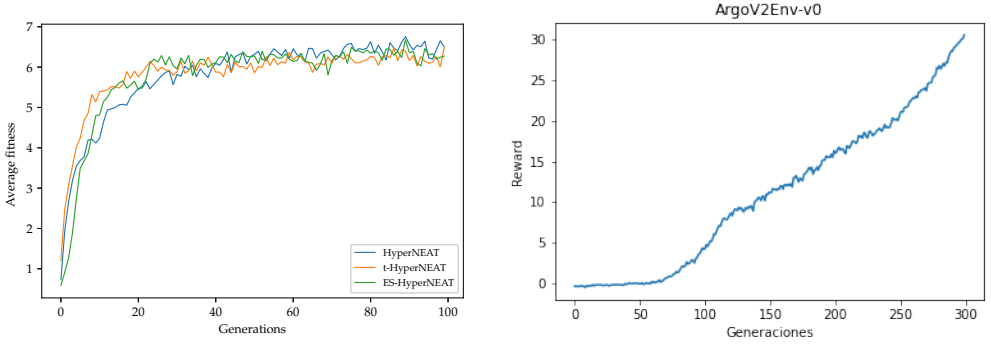


FIGURA A.1: Plataforma robótica ArgoV2.



(A) Recompensas obtenidas en problema de aprendizaje de caminatas con métodos basados en HyperNEAT.

(B) Recompensa obtenida en problema de aprendizaje de caminatas con métodos de plasticidad Hebbiana. Imagen extraída de Reyes [33].

FIGURA A.2: Resultados de trabajo no publicado usando algoritmo de plasticidad Hebbiana.

último puede deberse a la cantidad de parámetros que el algoritmo debe optimizar, pero que a largo plazo no mostró una cota en la recompensa obtenida y por tanto, a la distancia que es capaz de recorrer el ArgoV2 antes de que se cumpla alguna condición de término de experimento, como caídas o colisiones indeseables.

Bibliografía

[1] E. Najarro and S. Risi, "Meta-learning through hebbian plasticity in random networks," 2020. [Online]. Available: https://arxiv.org/abs/2007.02686

[2] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, "Evolution strategies as a scalable alternative to reinforcement learning," 2017. [Online]. Available: https://arxiv.org/abs/1703.03864

[3] A. Piontelli, *Development of normal fetal movements: The last 15 weeks of gestation.* Springer Milano, 2010.

[4] S. Zoia, L. Blason, G. D'Ottavio, M. Bulgheroni, E. Pezzetta, A. Scabar, and U. Castiello, "Evidence of early development of action planning in the human foetus: a kinematic study," *Exp. Brain Res.*, vol. 176, no. 2, pp. 217–226, Jan. 2007.

[5] M. Hoffmann, L. K. Chinn, E. Somogyi, T. Heed, J. Fagard, J. J. Lockman, and J. K. O'Regan, "Development of reaching to the body in early infancy: From experiments to robotic models," in *2017 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, 2017, pp. 112–119.

[6] M. Hoffmann, "The role of self-touch experience in the formation of the self," 2017.

[7] A. Roncone, M. Hoffmann, U. Pattacini, and G. Metta, "Automatic kinematic chain calibration using artificial skin: Self-touch in the icub humanoid robot," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 2305–2312.

[8] P. D. Nguyen, M. Hoffmann, U. Pattacini, and G. Metta, "Reaching development through visuo-proprioceptive-tactile integration on a humanoid robot - a deep learning approach," in *2019 Joint IEEE 9th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, 2019, pp. 163–170.

[9] F. Gama, M. Shcherban, M. Rolf, and M. Hoffmann, "Goal-directed tactile exploration for body model learning through self-touch on a humanoid robot," *IEEE Transactions on Cognitive and Developmental Systems*, pp. 1–1, 2021.

[10] M. Shcherban, "Efficient exploration of body surface with tactile sensors on humanoid robots," in *Master's thesis*, 2021.

[11] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," 2017.

[12] N. Vulin, S. Christen, S. Stevsic, and O. Hilliges, "Improved learning of robot manipulation tasks via tactile intrinsic motivation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, p. 2194–2201, Apr. 2021. [Online]. Available: http://dx.doi.org/10.1109/LRA.2021.3061308

[13] S. Dragiev, M. Toussaint, and M. Gienger, "Gaussian process implicit surfaces for shape estimation and grasping," *2011 IEEE International Conference on Robotics and Automation*, pp. 2845–2850, 2011.

[14] Z. Yi, R. Calandra, F. Veiga, H. Hoof, T. Hermans, Y. Zhang, and J. Peters, "Active tactile object exploration with gaussian processes," 10 2016.

[15] C. de Farias, N. Marturi, R. Stolkin, and Y. Bekiroglu, "Simultaneous tactile exploration and grasp refinement for unknown objects," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3349–3356, apr 2021. [Online]. Available: https://doi.org/10.1109%2Flra.2021.3063074

[16] Y. Bekiroglu, M. Björkman, G. Zarzar Gandler, J. Exner, C. H. Ek, and D. Kragic, "Visual and tactile 3d point cloud data from real robots for shape modeling and completion," *Data in Brief*, vol. 30, p. 105335, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352340920302298

[17] Q. Le, D. Kamm, A. Kara, and A. Ng, "Learning to grasp objects with multiple contact points," 06 2010, pp. 5062 – 5069.

[18] S. M. Nguyen, S. Ivaldi, N. Lyubova, A. Droniou, D. Gérardeaux-Viret, D. Filliat, V. Padois, O. Sigaud, and P.-Y. Oudeyer, "Learning to recognize objects through curiosity-driven manipulation with the icub humanoid robot," in *2013 IEEE Third Joint International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, 2013, pp. 1–8.

[19] K. Kangur, M. Giesel, J. Harris, and C. Hesse, "Visuo-tactile integration in texture perception: A replication and extension study," 04 2022.

[20] A. Del Prete, S. Denei, L. Natale, F. Mastrogiovanni, F. Nori, G. Cannata, and G. Metta, "Skin spatial calibration using force/torque measurements," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 3694–3700.

[21] "iCub Wiring Documentation," https://icub-tech-iit.github.io/documentation/icub_wiring/icub2_x/, accessed on 2024-03-30.

[22] G. Metta, P. Fitzpatrick, and L. Natale, "Yarp: Yet another robot platform," *International Journal of Advanced Robotic Systems*, vol. 3, no. 1, p. 8, 2006. [Online]. Available: https://doi.org/10.5772/5761

[23] "iCub Physical Tactile Sensors," https://icub-tech-iit.github.io/documentation/tactile_sensors/, accessed on 2024-03-30.

[24] M. Hoffmann, Z. Straka, I. Farkaš, M. Vavrečka, and G. Metta, "Robotic homunculus: Learning of artificial skin representation in a humanoid robot motivated by primary somatosensory cortex," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 2, pp. 163–176, 2018.

[25] J. X. Wang, Z. Kurth-Nelson, D. Tirumala, H. Soyer, J. Z. Leibo, R. Munos, C. Blundell, D. Kumaran, and M. Botvinick, "Learning to reinforcement learn," *CoRR*, vol. abs/1611.05763, 2016. [Online]. Available: http://arxiv.org/abs/1611.05763

[26] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," *CoRR*, vol. abs/1703.03400, 2017. [Online]. Available: http://arxiv.org/abs/1703.03400

[27] D. O. Hebb. Wiley, D. O. Hebb. Wiley, 1949., "The organization of behavior; a neuropsychological theory," 1949.

[28] G. W. Lindsay, M. Rigotti, M. R. Warden, E. K. Miller, and S. Fusi, "Hebbian learning in a random network captures selectivity properties of the prefrontal cortex," *Journal of Neuroscience*, vol. 37, no. 45, pp. 11 021–11 036, 2017. [Online]. Available: https://www.jneurosci.org/content/37/45/11021

[29] Scott Camazine, Jean-Louis Deneubourg, Nigel R Franks, James Sneyd, Eric Bonabeau, and Guy Theraula, "Self-organization in biological systems, volume 7," 2003.

[30] A. Mordvintsev, E. Randazzo, E. Niklasson, and M. Levin, "Growing neural cellular automata," *Distill*, 2020, https://distill.pub/2020/growing-ca.

[31] A. Soltoggio, P. Durr, C. Mattiussi, and D. Floreano, "Evolving neuromodulatory topologies for reinforcement learning-like problems," in *2007 IEEE Congress on Evolutionary Computation*, 2007, pp. 2471–2478.

[32] E. Rohmer, S. P. N. Singh, and M. Freese, "Coppeliasim (formerly v-rep): a versatile and scalable robot simulation framework," in *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2013, www.coppeliarobotics.com.

[33] P. Reyes and M.-J. Escobar, "Neuroevolutive algorithms for learning gaits in legged robots," *IEEE Access*, vol. 7, pp. 142 406–142 420, 2019.

[34] S. Risi and K. O. Stanley, "An enhanced hypercube-based encoding for evolving the placement, density, and connectivity of neurons," *Artificial Life*, vol. 18, no. 4, pp. 331–363, 2012.