

# Dyna-1: Desarrollo, Construcción y Pruebas Experimentales de un Robot Cuadrúpedo de Diseño Abierto

Tadeo Casiraghi\*, Gabriel Torre\*†, Roberto Bunge\*†, Ignacio Mas\*†

† CONICET, Argentina

\* Universidad de San Andrés, Departamento de Ingeniería, LINAR, Buenos Aires, Argentina

‡ Universidad de Buenos Aires, Facultad de Ingeniería, Instituto de Ingeniería Biomédica

**Resumen**—Presentamos un nuevo diseño de cuadrúpedo de 12 grados de libertad, centrado en reducir el costo y la complejidad de fabricación y maximizar el uso de componentes disponibles regionalmente. Para facilitar el ensamblado, las piernas son modulares, y los motores están lo mas cerca posible del cuerpo para reducir su inercia. Se utilizan motores brushless junto a controladores Odrive que permiten un control simple con su interfaz por CAN. El cuerpo contiene los motores de los hombros y toda la electrónica necesaria para su funcionamiento. Para controlar el cuadrúpedo se implementa el sistema operativo ROS2. Todos los procesos cruciales para el funcionamiento del robot se corren en la computadora a bordo, y los demás en una computadora externa. Se demuestra experimentalmente que cada pierna es capaz de saltar hasta 30 cm con un peso similar a un cuarto del cuerpo. Se ensayó el robot con peso agregado de hasta 4 kg y se observa buen rendimiento. Se creó un simulador en Pybullet para evaluar algoritmos de autonomía de manera eficiente. El costo total del robot es de aproximadamente 3000 USD. Todo el software y los modelos del hardware son de diseño abierto y pueden encontrarse en un repositorio público.

## I. INTRODUCTION

Los robots cuadrúpedos utilizados en la investigación de la locomoción autónoma y la aplicación de algoritmos de aprendizaje automático suelen ser costosos y requieren piezas hechas a medida o incluso motores personalizados [1] [2]. Otros robots diseñados específicamente para ser de bajo costo [3] utilizan piezas y procesos de manufactura que pueden ser difíciles de reproducir o muy costosos en otras regiones del mundo, como las impresiones de Multijet (MJP), impresión 3D metálica, o mecanizado CNC de 5 ejes. Robots comerciales como los de Unitree cuestan aproximadamente 3000 USD, pero esto no incluye el precio que se le agrega para poder acceder al control de bajo nivel de los actuadores o el costo de envío e importación desde China, lo que suma por lo menos otros 6000 USD. Por sobre esto, al ser un producto comercial es más difícil de arreglar, y no permite modificaciones o agregados que no sean de carácter “oficial”.

Este artículo presenta el diseño de un robot cuadrúpedo como se ve en la Fig. 1, cuyo enfoque es reducir la complejidad del proceso de manufactura, maximizar el uso de componentes disponibles en la región, y reducir el costo total sin sacrificar la destreza y la potencia del movimiento. Los requerimientos iniciales del diseño son:

- 40 cm entre ejes de piernas: Luego de un análisis previo técnico-económico se decidió que este tamaño sería el mayor posible sin aumentar considerablemente el precio.



Figura 1. Fotografía del Dyna-1: plataforma de investigación con 12 grados de libertad. En la imagen se energiza al sistema con una fuente de pared, pero también puede utilizarse con baterías 8S. Se editó la imagen para remover el fondo y resaltar el robot.

- Locomoción con trote y galope: El robot debería ser capaz de moverse con los pasos típicos de cuadrúpedos.
- Capacidad de saltar un mínimo de 2 cm: Con esta cota de salto se cerciora que el robot tiene un mínimo de destreza.
- Galope máximo a 1m/s: Robots como el mini cheetah de MIT pueden galopar hasta 4 m/s. Una cota de 1 m/s asegura que el diseño, a pesar de su reducido costo, pueda moverse a una velocidad razonable.
- 12 grados de libertad: Al tener 3 grados de libertad por pierna el robot sería capaz de movimientos complejos en cualquier dirección.

Tanto el hardware y el software del cuadrúpedo son abiertos. Puede encontrarse toda la información pertinente al diseño en el repositorio público [4].

## II. ESPECIFICACIONES DE DISEÑO

En el Cuadro I se pueden observar atributos distintivos del Dyna-1. El valor de carga útil informado no incluye el peso de las baterías. Al utilizar el cuadrúpedo con baterías montadas se reducirá la carga útil efectiva. A continuación presentamos las características principales que definen al diseño del robot.

Parámetros	Valor	Unidades
Largo Brazo	19	cm
Largo Antebrazo	19	cm
Peso del Cuerpo (sin baterías)	3.7	kg
Peso Total (sin baterías)	8	kg
Tiempo estimado de autonomía con dos baterías 4S de 4000 mah en serie	30	min
Peso de baterías	840	g
Carga útil (efectiva / sin batería)	4	kg
Carga útil (con batería)	3.16	kg
Costo aproximado	3000	USD

Cuadro I  
VALORES CARACTERÍSTICOS DEL CUADRÚPEDO

**II-1. Morfología de las piernas:** Diseños de cuadrúpedos como el del Minitaur [5] utilizan para las piernas mecanismos de 5 vínculos. Estos permiten simplificar o eliminar la reducción mecánica de los motores y la inercia de las piernas, pero pueden tener problemas con el calentamiento de los motores y el contacto con obstáculos o cambios de terreno. Otros diseños implementan una estructura de 3 o más miembros en serie con vinculación fija [6] o independiente. Estos son más bio-miméticos y permiten una mayor complejidad mecánica y mayor adaptación pasiva al terreno. Sin embargo es esta misma complejidad que fuerza a que el diseño sea más costoso e intrincado para armar y controlar. El estilo más común es el de piernas con 2 miembros como hace uso este diseño. Se consigue así un balance entre capacidad de movimiento y complejidad de diseño, armado y costo. Para el Dyna-1, es posible cambiar de una configuración con todos los codos apuntando para atrás a una en la que apuntan al centro rotando el tercer miembro de las piernas de atrás. Sin embargo, dada a la popularidad de los robots de Unitree [7], la configuración por defecto es la que se ve en la Fig. 1.

**II-2. Selección de motores y controladores:** Como motores se escogieron los brushless GARTT ML5010 siguiendo el ejemplo del “Open Dynamic Robot Initiative” [8]. Estos motores livianos en conjunto con los controladores Odrive [9] proporcionan un control preciso y simple. Estos controladores de motores brushless tienen un ciclo de control interno de tres capas: torque, velocidad y posición [10]. Cada capa tiene un controlador proporcional (P) o proporcional-integral (PI) y se realimentan entre si y con los datos que devuelve el motor y el encoder magnético AS5047P montado en cada eje. Los parámetros de estos controladores fueron calibrados para mejorar la actuación de los motores de cada eje mientras se mueven sin contacto con el suelo, el caso más inestable del movimiento del cuadrúpedo.

**II-3. Transmisión de potencia:** Dado que los motores brushless están diseñados para girar a altas rpm y entregar torques bajos, se utilizan correas sincrónicas para multiplicar el torque, como también transferir la potencia desde el motor hasta el eje de cada módulo. Se utilizan dos correas en serie, cada una proporcionando una reducción de 1:3, obteniendo así una reducción total de 1:9.

**II-4. Modularidad:** Para simplificar la manufactura y construcción del robot se diseñaron las piernas de forma tal que se repitan los componentes. Con ese objetivo, se crearon los módulos que multiplican y desplazan el torque de los motores. En la Fig. 2 se observa una vista explotada del diseño nominal del módulo.

Como mencionamos en una publicación previa sobre el diseño de la pierna [11], reducir la inercia facilita el control, y a su vez reduce el torque necesario para cada movimiento. En base a eso, para el diseño de los módulos se posiciona a los motores (el componente más pesado) lo más cerca posible al cuerpo. En la Fig. 3 se pueden observar los segmentos y sus respectivos ejes.

**II-5. Cuerpo:** El cuerpo del robot contiene los motores de los hombros (con sus reducciones correspondientes), la computadora central, la IMU (BNO055 de Adafruit), los controladores Odrive y todo el sistema de distribución de energía. Las baterías se sostienen en la parte inferior. En la Fig. 4 se pueden observar las dimensiones de diseño nominal. La rigidez de la estructura se obtiene con cuatro tubos de fibra de carbono y placas de acrílico. En la Fig. 5 puede verse el esquema de cableado del Dyna-1. La IMU se coloca en el centro inferior del cuerpo.

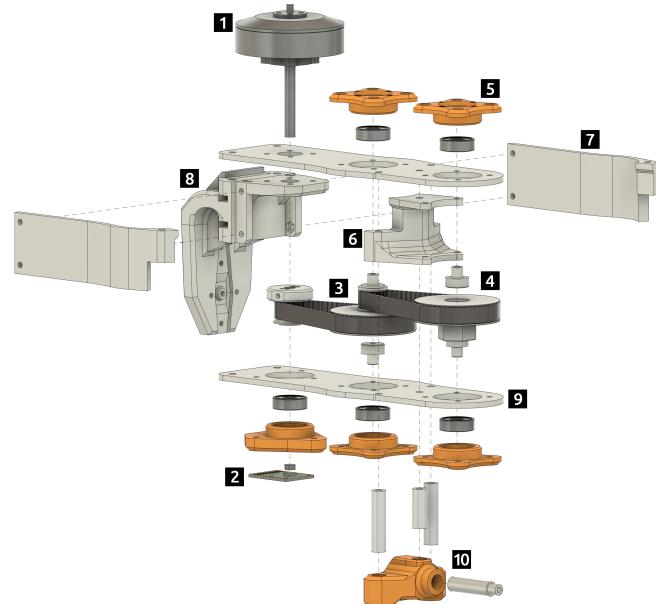


Figura 2. Ensamblaje de módulo nominal. **1:** Motor sin escobillas Gartt ML5010 300KV con eje modificado para mayor alcance. **2:** Encoder magnético AS5047P montado en el eje del motor. **3:** Correa de distribución Gates Powergrip GT3 de 225 dientes y 9 mm. Cada correa proporciona una reducción de 3x, sumando un total de 9x. **4:** Diseño de polea nominal. Se utiliza un perno M3 como su núcleo para proporcionar rigidez. Dependiendo del uso, este eje también puede ser parte del soporte trasero de los módulos siguientes. **5:** Soporte de eje con rodamientos de bolas 626 ZZ. **6/7:** Estas partes mantienen el módulo unido y añaden rigidez torsional. **8:** El soporte trasero tiene múltiples diseños de conexión dependiendo del uso del módulo. **9:** Las placas superior e inferior proporcionan la estructura sobre la cual se construye el módulo. **10:** El eje adicional se utiliza cuando el módulo se une al cuerpo principal.



Figura 3. Segmentos y ejes de movimiento de la pierna del Dyna-1. El segmento L1 controla el eje A1 que corresponde a la ab/aducción. Este segmento está integrado en el diseño del cuerpo. Los segmentos L2 y L3 controlan los ejes A2 (hombro) y A3 (codo) respectivamente.

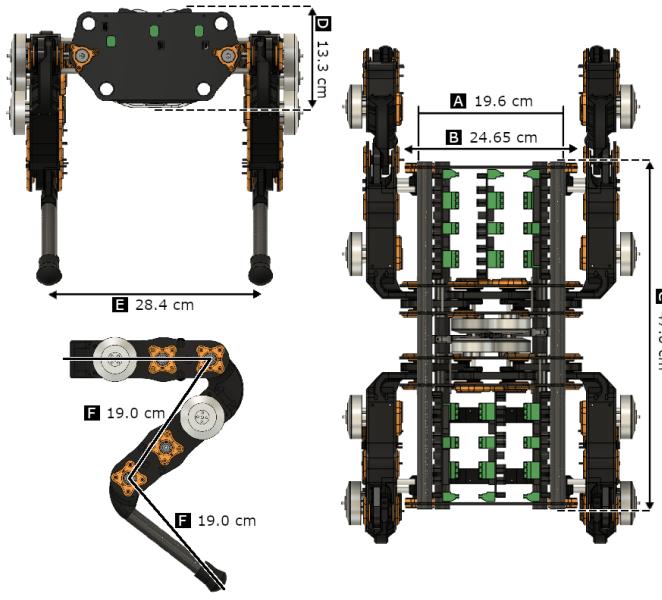


Figura 4. Dimensiones de diseño sin baterías agregadas (Las baterías y las piezas que las sostienen agregan 3.5 cm al alto del cuerpo [D] al montarse por debajo del mismo). **A:** Ancho entre ejes de piernas. **B:** Ancho de cuerpo. **C:** Largo de cuerpo y largo entre ejes teóricos de las piernas. **D:** Alto del cuerpo. **E:** Ancho entre puntas de piernas. **F:** Largos de links teóricos.

**II-6. Manufactura:** La construcción del Dyna-1 se basa fuertemente en impresión 3D. La mayoría de las piezas se hacen con PLA. Las placas que le dan rigidez y estructura al cuerpo son de acrílico cortadas con láser, pero pueden ser cortadas con CNC o a mano. También se probó con Poliestireno de Alto Impacto, pero dada la facilidad de cortar el acrílico con láser se optó por no usar ese material. De esta manera, la gran mayoría de los componentes del robot son fáciles de encontrar o fabricar en mercados locales o con una impresora 3D.

### III. CONTROL

Para controlar al robot se utiliza el *middleware* de ROS2 [12]. ROS2 (Robot Operating System 2) es un sistema operativo para robots que proporciona un marco de trabajo para el desarrollo de software en robótica, incluyendo una colección de herramientas y bibliotecas que tienen como objetivo simplificar la tarea de crear un comportamiento complejo y robusto para múltiples plataformas y lenguajes de código. ROS2 simplifica la comunicación entre procesos del robot con una estructura de nodos que pueden publicar o suscribirse a tópicos. En la Fig. 6 pueden observarse los nodos de control que se implementan en la arquitectura básica del Dyna-1.

Dado que el cuadrúpedo se piensa usar como plataforma de evaluación de algoritmos de locomoción, como también para correr procesos de detección de imágenes o procesar información de un lidar, solo aquellos nodos que sean cruciales para el buen funcionamiento del robot se ejecutan en la computadora a bordo. El resto de los nodos se ejecutan en una computadora externa y se conectan entre sí por medio de wifi.

#### III-A. Computadora a Bordo

La computadora a bordo es una Nvidia Jetson TX2 y ejecuta una serie de nodos que se encargan de la comunicación con los controladores de motores, de la interpretación de las mediciones de los motores y del comando de movimientos deseados. Estos nodos son:

**III-A1. Can Bridge:** Este nodo se encarga de la comunicación por medio del Bus de CAN con los 6 controladores de motores. Los Odrive están programados para mandar por este medio las mediciones de los encoder y las corrientes de los motores cada 10 ms. Es decir que el nodo publica en el tópico “RX” 2400 veces por segundo. Por otro lado también recibe los comandos deseados por el tópico “TX”. Estos pueden ser de posición, velocidad o torque deseado, como también comandos de cambio de estructura de control o encendido y apagado de motores.

**III-A2. Motores:** El nodo de motores recibe todas las mediciones de posición y para cada motor calcula el ángulo del la articulación en función a los ejes propuestos para la pierna. Luego, junto a las mediciones de corriente, publica dos mensajes con todos los datos empaquetados. También recibe los pedidos de cambio de corriente máxima, cambio de estado de motores, y comandos de posición angular.

**III-A3. Dynabot real Interface:** Este nodo se encarga de recibir todos los comandos por parte del joystick y la máquina de estados del robot. En base a eso y al estado actual del robot que recibe del nodo de Motores, calcula y envía los comandos de posición que se requieren.

**III-A4. IMU:** El nodo de IMU lee las señales que manda el sensor de mediciones iniciales por I2C y las publica como tópico.

#### III-B. Computadora Externa

En la computadora externa se corren los nodos que no sean cruciales para el funcionamiento del robot, y que tienen que

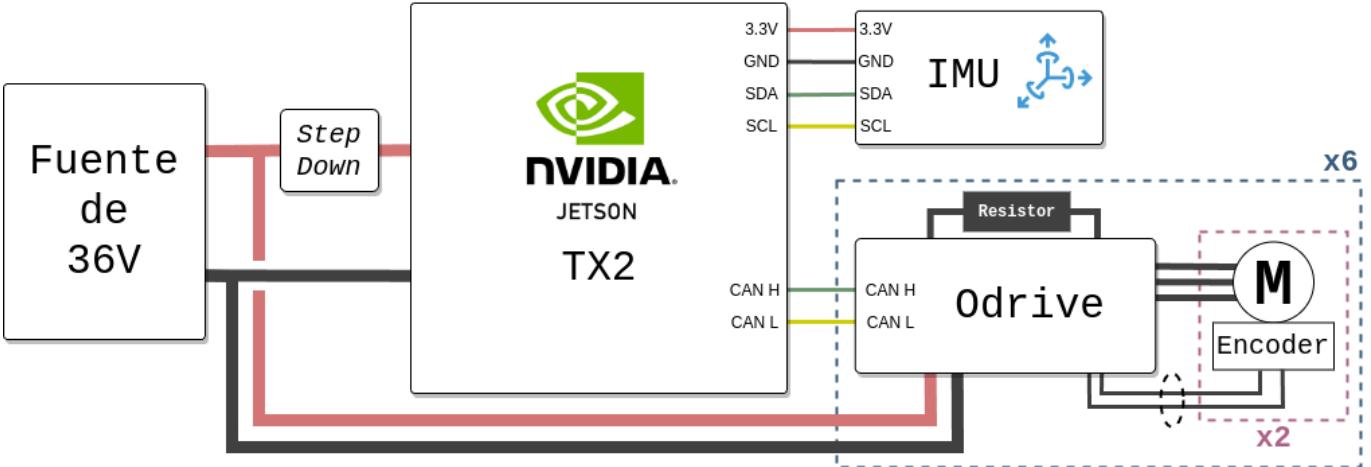


Figura 5. Conexión eléctrica del Dyna-1. El bus de CAN conecta los 6 Odrive a la TX2, y cada driver controla 2 motores. Los encoder de cada motor se conectan directamente al Odrive, y la IMU se conecta a la TX2 por I2C. Tanto la computadora central como los controladores de los motores son alimentados desde una única fuente de energía. Dado que la Jetson TX2 no puede recibir 36V de entrada se utiliza un conversor Step Down.

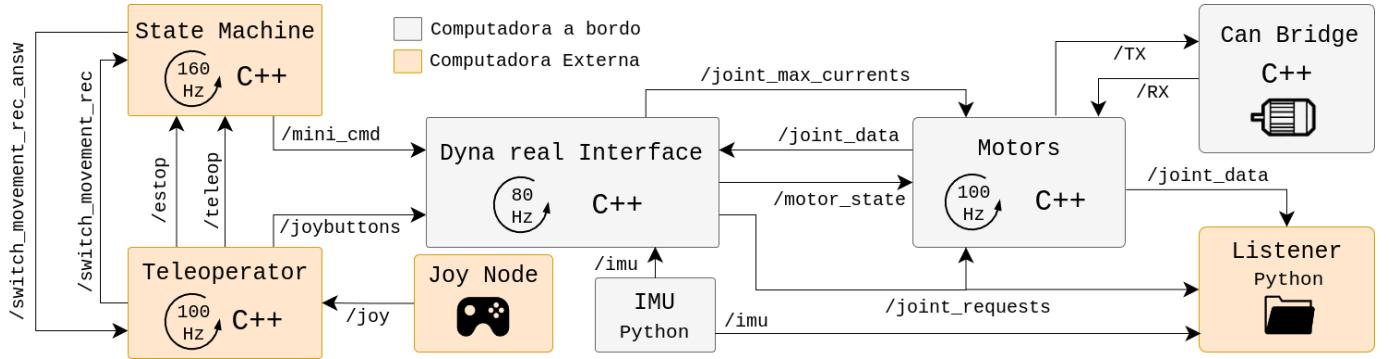


Figura 6. Nodos de ROS2 con sus correspondientes tópicos de publicación y subscripción. La comunicación entre los nodos que se corren en la computadora a bordo y los de la computadora externa se realiza mediante una red privada de wifi.

ver con el monitoreo de variables, funciones de teleoperación y control de modos de funcionamiento. Estos son:

*III-B1. Listener:* Este nodo se encarga de guardar todas las mediciones de posición y corriente de los Odrive, los comandos de posición, y también los valores de la IMU. Todos los datos son guardados en archivos csv para su posterior análisis.

*III-B2. Joy Node:* Este nodo se incluye con las bibliotecas provistas por la instalación básica de ROS2. Traduce los comandos del control a publicaciones periódicas de los botones y joysticks.

*III-B3. Teleoperator:* El nodo de teleoperación recibe los comandos del control remoto y los interpreta para mandar a la máquina de estados y al interfaz con el robot. Se encarga de informar el pedido de cambio de tipo de movimiento (Fijo a Caminar), la parada de emergencia, y los comandos de avance y retroceso.

*III-B4. State Machine:* La máquina de estados se utiliza para definir, en base a lo que se interpreta del control remoto, los movimientos que se le comandarán al cuadrúpedo

#### IV. RESULTADOS EXPERIMENTALES

Antes de construir el robot completo se evaluaron los componentes por separado como se informó en el artículo previo [11]. Además de lo especificado, se evaluó el desempeño de la pierna mientras salta. Esto cumple varios objetivos:

- Confirmar que la pierna es capaz de cumplir con el requerimiento de diseño
- Evaluar el desempeño del sistema en una situación dinámica y con picos de corriente altos
- Evaluar el desempeño de la amortiguación por control de torques durante la caída (Este control todavía no se ha implementado con ROS2)

En la Fig. 7 se observa la altura máxima que se obtuvo sin exceso de corriente. La altura máxima superó ampliamente el requerimiento de diseño. Esto es debido a que los torques necesarios calculados e implementados para galopar a 1 m/s son mucho mayores a los requeridos para saltar 2 cm.

Una vez ensamblado el robot completo se ensayó el galope utilizando la marcha básica descrita en el artículo del Dyna-0 [13]. Esta consiste en dos fases, apoyo y balanceo. El apoyo es una función sinusoidal y el balanceo es una curva de Bezier

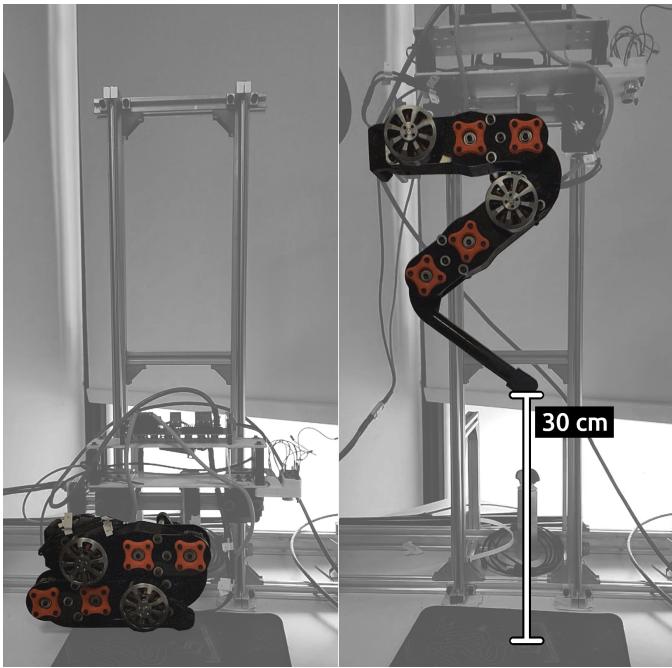


Figura 7. Evaluación de salto de pierna del Dyna-1. Se simulan los efectos del cuerpo agregando un cuarto de su peso al momento del ensayo.

de 12 puntos. En la Fig. 8 y la Fig. 9 pueden observarse la frecuencia de mensajes de control y mediciones. Estos motores brushless de uso típico en drones están diseñados para girar a grandes velocidades y enfriarse con el viento que desplazan. En este caso los estamos utilizando para entregar torques

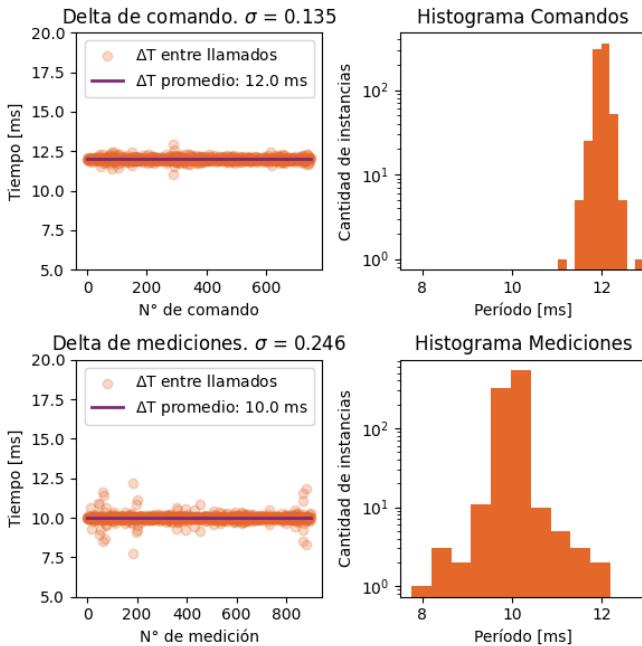


Figura 8. Períodos de señales de control y señales de medición. El control se busca hacer a 80Hz (período de 12.5 ms). En base a la estabilidad del período de control se estima que la definición de selección de período no permite decimales. Las frecuencia de mediciones es fijada por el nodo de Motores. En este caso la desviación es mayor pero nunca mayor que la mitad del período.

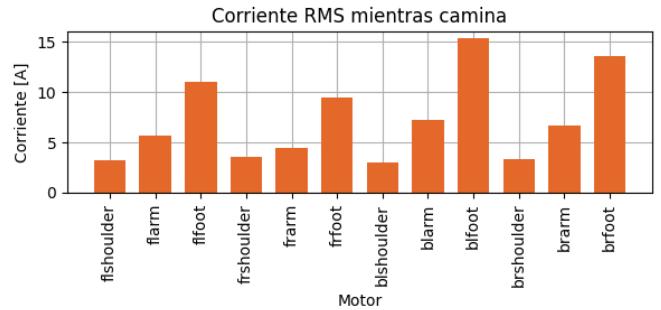


Figura 9. Corrientes RMS de cada motor mientras el Dyna-1 camina. La diferencia de corriente entre los pies de cada pierna puede ser atribuida al paso 'ciego' que se está empleando. Un futuro control de paso podría solucionar esto. Estas corrientes también son susceptibles al lazo de control interno de los controladores.

altos a bajas revoluciones. De todos modos, en los ensayos discutidos en [11] se observó que comienzan a calentarse a partir de los 20A continuos. Como se observa en la Fig. 9, la corriente rms de los motores mientras camina el cuadrúpedo no superan los 15A. En base a esto se puede extraer que el robot aún es capaz de llevar más peso ya sea de más sensores o de cargas. Adicionalmente, es posible que optimizar el galope del cuadrúpedo produzca una redistribución de corrientes para producir más torque con los motores de los hombros y menos con los motores de los codos (*arm* y *foot* respectivamente).

También se evaluó la corriente RMS máxima al cargar peso adicional. La Fig. 10 muestra la corriente RMS máxima registrada mientras camina el cuadrúpedo. Como puede observarse se puede cargar hasta 4 kg sin que la corriente RMS supere 20 A. Este es el límite al cual se observa que los motores comienzan a calentarse.

La Fig. 11 grafica las potencias de consumo registradas durante el ensayo de carga de peso. Estos resultados pueden verse afectados ante un paso distinto o más eficiente.

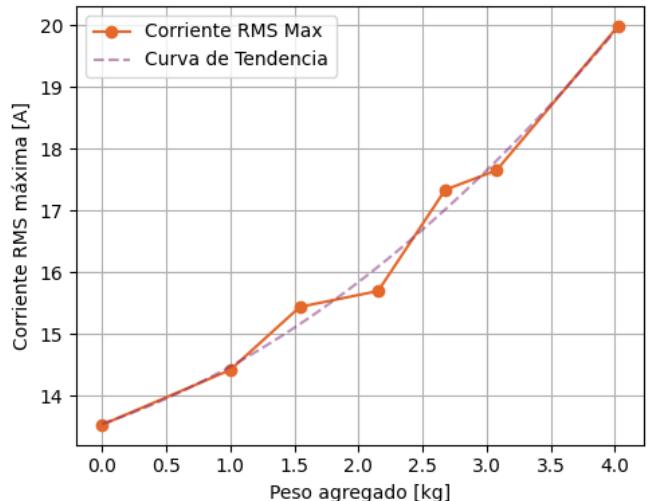


Figura 10. Corriente RMS de motores máxima (durante el galope) en función al peso agregado al cuadrúpedo. La línea de tendencia se calcula con un polinomio de orden 2. Se utilizó el mismo galope para todos los ensayos. Es posible que adaptando el paso para soportar mayor peso se reduzca la corriente máxima.

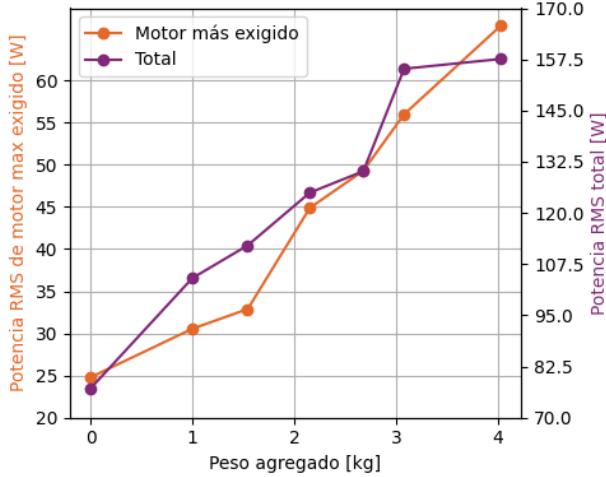


Figura 11. Potencia del motor más exigido (eje izquierdo) y potencia total (eje derecho) durante el ensayo de carga de peso. Como es de esperarse, a mayor carga de peso mayor la potencia consumida.

## V. SIMULACIÓN

Para poder entrenar y evaluar mecanismos de control sin tener que utilizar el robot físico se armó una simulación con Pybullet como se ve en la Fig. 12. PyBullet es una biblioteca de física y robótica para Python que permite la simulación de robots en un entorno 3D. Ofrece una API que facilita la interacción con el entorno simulado, permitiendo a los usuarios crear y manipular objetos, configurar el mundo simulado y recopilar datos de sensores y actuadores.

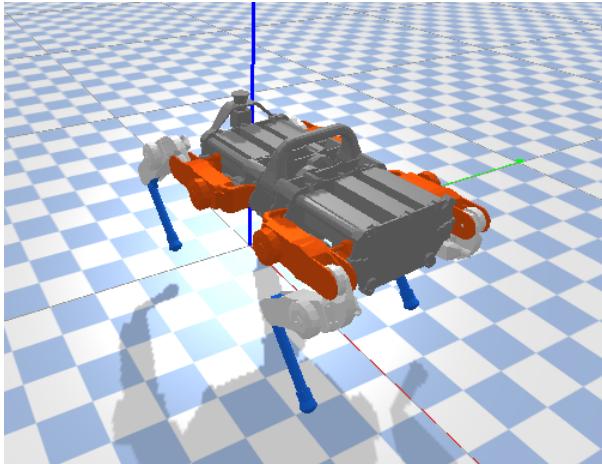


Figura 12. Simulación del cuadrúpedo en Pybullet. El motor físico utiliza las inercias y las masas informadas del robot para calcular las fuerzas y momentos sobre los motores. Las características físicas del Dyna-1 son calculadas en base a modelo de Fusion 360, asignando los materiales pertinentes.

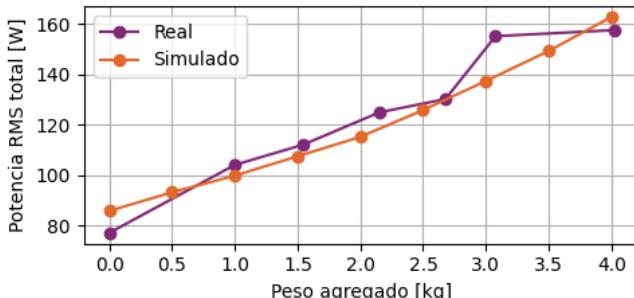


Figura 13. Comparación entre resultados experimentales y simulados de potencia total durante el galope.

La simulación tiene en cuenta los pesos y las inercias de todos los links del robot. Con este simulador se puede, por ejemplo, entrenar un algoritmo de aprendizaje reforzado con varios terrenos y variedades de condiciones de suelo; eso es: terreno, fricción, obstáculos y fuerzas externas. En la Fig. 13 se observar que la simulación sigue la tendencia de crecimiento de potencia en función del peso agregado.

## VI. TRABAJO FUTURO

En un futuro se hará una nueva ronda de calibración de los controladores internos de los motores para buscar un mejor desempeño de cada motor en particular. Con esto buscamos reducir la “flexibilidad” del cuadrúpedo ante fuerzas internas o su propia inercia.

Con esta plataforma robótica se podrán evaluar todo tipo de algoritmos de locomoción, ya sean heurísticos u entrenados. Se podrán montar además sensores como LIDAR, cámaras, o incluso sensores de contacto para las piernas. Si la capacidad de cómputo no es suficiente para manejar el algoritmo de control y todos los sensores adicionales se puede cambiar la computadora a bordo por una más potente como por ejemplo la Jetson Orin.

A futuro es posible que también se pase a simular el sistema en Isaac Sim [14] de Nvidia. Este simulador permite más fácilmente parallelizar ensayos y el aprendizaje automático, aprovechando así el poder de cómputo de las placas gráficas.

## REFERENCIAS

- [1] P. M. Wensing, A. Wang, S. Seok, D. Otten, J. Lang, and S. Kim, “Proprioceptive actuator design in the mit cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots,” *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 509–522, 2017.
- [2] P. Fankhauser and M. Hutter, “Anymal: A unique quadruped robot conquering harsh environments,” *Research Features*, vol. 126, pp. 54 – 57, 05 2018.
- [3] S. Vatau, V. Ciupe, C. Moldovan, and I. Maniu, “Mechanical design and system control of quadruped robot,” *MECHANIKA. Nr*, vol. 5, 01 2010.
- [4] T. Casiraghi, G. Torre, R. Bunge, and I. Mas, “dynal-quadruped,” Mar. 2024. [Online]. Available: <https://github.com/udesa-ai/dynal-quadruped>
- [5] G. Kenneally, A. De, and D. E. Koditschek, “Design principles for a family of direct-drive legged robots,” *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 900–907, 2016.
- [6] A. Spröwitz, A. Tuleu, M. Ajaoolejian, M. Vespiagnani, R. Moeckel, P. Eckert, M. D’Haene, J. Degrave, A. Nordmann, B. Schrauwen, J. J. Steil, and A. J. Ijspeert, “Oncilla robot: a versatile open-source quadruped research robot with compliant pantograph legs,” *CoRR*, vol. abs/1803.06259, 2018. [Online]. Available: <http://arxiv.org/abs/1803.06259>
- [7] Unitree. [Online]. Available: <https://shop.unitree.com/>
- [8] M. Wüthrich, F. Widmaier, F. Grimminger, J. Akpo, S. Joshi, V. Agrawal, B. Hammoud, M. Khadiv, M. Bogdanovic, V. Berenz, J. Viereck, M. Naveau, L. Righetti, B. Schölkopf, and S. Bauer, “Trifinger: An open-source robot for learning dexterity,” *CoRR*, vol. abs/2008.03596, 2020. [Online]. Available: <https://arxiv.org/abs/2008.03596>
- [9] O. Weigl. [Online]. Available: <https://odriverobotics.com/>
- [10] ———. [Online]. Available: <https://docs.odriverobotics.com/v/0.5.4/control.html>
- [11] T. Casiraghi, G. Torre, R. Bunge, and I. Mas, “Low-cost quadruped leg module: Design and initial validation,” *Jornada Argentina de Robótica (JAR)*, 03 2022.
- [12] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, “Robot operating system 2: Design, architecture, and uses in the wild,” *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022. [Online]. Available: <https://www.science.org/doi/abs/10.1126/scirobotics.abm6074>
- [13] G. Torre, A. Grillo, and R. Bunge, “Dyna-0: Cuadrúpedo para aprendizaje automático,” *Jornada Argentina de Robótica (JAR)*, 03 2022.
- [14] Nvidia. [Online]. Available: <https://developer.nvidia.com/isaac-sim>