

Universidad de Ingeniería Y Tecnología

Departamento de Ingeniería Mecatrónica



*Real Life Experience*

## Análisis de un robot cuadrúpedo

Integrantes

Palma Rodríguez, Diego Alonso V.

Solórzano Pinzás, Renzo Lucciano

Asesor: Ph.D. Oscar E. Ramos

Lima - Perú

2021-2

---

# 1. Introducción

En la actualidad, el análisis, diseño, generación de movimiento y desarrollo de robots móviles son áreas de investigación de gran interés en el campo de la robótica. Esto se debe a que los robots móviles tienen un amplio campo de aplicaciones en diversos ámbitos, donde los más prometedores son la exploración espacial, la aplicación militar, respuesta a emergencias, inspección, mantenimiento, logística, construcción, seguridad, entretenimiento y cuidado de ancianos [1, 2]. Entre todos los robots móviles, se tiene al robot cuadrúpedo que es un robot con 4 patas, donde cada una de estas suele tener desde 2 hasta 4 grados de libertad. Estos tipos de robot son considerados superiores a los robots con ruedas debido a su potencial para explorar diferentes tipos de terrenos [1]. Como ejemplos de este tipo de robots tenemos al robot de código abierto: Solo [3], el Cheetah (MIT) [4], AnyMal (ETH) [5], HyQ (IIT) [6], y uno de los robots cuadrúpedos más avanzados e inteligentes actualmente, el robot Spot [7].

A pesar del gran interés que ha generado las aplicaciones de este tipo de robots y las investigaciones que se realizanntorno a este, la literatura existente con respecto a los fundamentos físicos de este tipos de robots es muy escasa y no suele ser detallada. En ese sentido, en el presente informe busca brindar a detalle el análisis cinemático y dinámico de un robot cuadrúpedo con el objetivo de comprender los fundamentos del control cinemático y el control óptimo del robot.

El presente informe se divide tres secciones. La sección 2 describe de manera consisa un robot con base flotante, qué son las articulaciones actuadas y las coordenadas generalizadas que se toman en cuenta para este trabajo. En la sección 3 se encuentra el análisis cinemático donde se muestran las ecuaciones para cinemática directa, diferencial y cómo realizar el control cinemático del robot. La sección 4 se enfoca en el análisis dinámico del robot, donde se muestra el modelo dinámico para un robot con base flotante, las consideraciones a tomar en cuenta para la estabilidad en la caminata del robot y un ejemplo de un esquema de control propuesto basado en el centro de presión.

## 2. Robot con base flotante

Previo al análisis de un robot cuadrúpedo es importante identificar las características que juegan un rol importante en el movimiento del robot. Una de las características más relevantes que poseen este tipo de robots es su base flotante. Es decir, que su base, ubicada en el torso, no se encuentra restringida a una superficie empotrada, sino que se mueve libremente junto con el robot sin la necesidad de un actuador.

Este movimiento libre que posee la base flotante da lugar a un reto de control conocido como subactuación. Este indica que el movimiento no puede controlarse directamente, sino de forma indirecta a través del movimiento adecuado de las articulaciones, ya sea a nivel cinemático [8] o dinámico [9]. Por lo tanto, es mandatorio que la representación de un robot cuadrúpedo incluya el efecto de la base flotante en la cinemática y dinámica del robot, con el fin de que se genere el movimiento de todo el cuerpo del robot y, por ende, se aproveche la característica de este tipo de robots.

A continuación, se definen las articulaciones actuadas del robot, y la posición y orientación

---

de la base flotante para finalmente definir las coordenadas generalizadas que representan la posición y orientación del robot.

## 2.1. Articulaciones actuadas

La articulación de un robot permite la unión entre los eslabones rígidos del mismo. Una articulación se denomina actuada cuando está asociada a un actuador que genera su movimiento, ya sea mediante un motor o un mecanismo de transmisión de movimiento. Asimismo, las articulaciones de la mayoría de los robots cuadrúpedos permiten un movimiento rotacional o lineal, denominadas juntas de revolución y prismáticas, respectivamente [10].

A continuación, utilizaremos la notación matemática utilizada en [8] para realizar la descripción del robot. Entonces, consideremos un robot cuadrúpedo de  $n = r + p$  articulaciones actuadas, donde  $r$  es el número de articulaciones de revolución y  $p$  es el número de articulaciones prismáticas. En base a esta configuración, el robot posee un espacio articular de  $n$  dimensiones. Este espacio articular contiene todos los posibles valores que la variable articular,  $\mathbf{q}_a$  pueda tomar. Es decir,  $\mathbf{q}_a = [q_1 \ q_2 \ \cdots \ q_n]^T \in \mathbb{R}^n$ , donde cada valor articular  $q_i$  representa una articulación actuada con ciertos límites articulares.

## 2.2. Base flotante

En un robot cuadrúpedo, la base flotante, o simplemente base, está representada por el torso o la cintura del robot. La posición y orientación (postura) de la base se representa como sigue:

$$\mathbf{x}_b = \begin{bmatrix} \mathbf{p}_b \\ \boldsymbol{\varphi}_b \end{bmatrix} \in SE(3) \quad (1)$$

donde  $\mathbf{p}_b$  es la posición de la base,  $\boldsymbol{\varphi}_b$  es su orientación dada por una parametrización de un grupo especial ortogonal  $SO(3)$ , y  $SE(3)$  indica que  $\mathbf{x}_b$  pertenece al grupo espacial euclideo que dependiendo de la representación de la orientación puede tener 6 o 7 grados de libertad (gdl). En el presente informe, la posición se representará mediante coordenadas cartesianas  $\mathbf{p} \in \mathbb{R}^3$ ; y la orientación, mediante cuaterniones unitarios en el espacio Hamiltoniano  $\mathbf{Q}_b \in \mathbb{H}$ , por lo que se tendrá 7 gdl. Además, es importante mencionar que tanto  $\mathbf{p}_b$  y  $\boldsymbol{\varphi}_b$  están representados con respecto a un sistema de referencia inercial  $\{\mathcal{I}\}$ .

## 2.3. Coordenadas generalizadas

Consideremos que el robot está conformado por  $N$  cuerpos, entonces se necesitan  $7N$  coordenadas para especificar la posición y la orientación de todos los cuerpos en relación a un marco de referencia inercial  $\{\mathcal{I}\}$ . Sin embargo, las  $7N$  coordenadas pueden expresarse como funciones de un conjunto más pequeño de coordenadas  $\mathbf{q}$  que son todas independientes, representan el estado completo del robot y contienen tanto la base flotante como las articulaciones actuadas. Las coordenadas de este conjunto se conocen como *coordenadas generalizadas* ( $\mathbf{q}$ ).

[11]. Estas coordenadas  $\mathbf{q}$  y el vector de velocidad  $\dot{\mathbf{q}}$  se expresan como se muestra en (2).

$$\mathbf{q} = \begin{bmatrix} \mathbf{x}_b \\ \mathbf{q}_a \end{bmatrix} \quad \text{y} \quad \dot{\mathbf{q}} = \begin{bmatrix} \dot{\mathbf{x}}_b \\ \dot{\mathbf{q}}_a \end{bmatrix} \quad (2)$$

donde cada término de  $\mathbf{q}$  se definió en las secciones 2.1 y 2.2. Se debe notar que  $\dot{\mathbf{q}}$  resulta de derivar  $\mathbf{q}$ , y por ende  $\dot{\mathbf{x}}_b$  depende de la representación utilizada en  $\mathbf{x}_b$ . En ese sentido, se suele utilizar la definición del *twist* de la base,  $\xi_b = [\mathbf{v}_b \ \boldsymbol{\omega}_b]^T$ , para representar las coordenadas de velocidad (3), puesto que es independiente de la parametrización euclídeana dada por  $\mathbf{x}_b$  [11]. La relación entre  $\xi_b$  y  $\dot{\mathbf{x}}_b$  se determina en la sección 3.2.

$$\dot{\mathbf{q}} = \begin{bmatrix} \xi_b \\ \dot{\mathbf{q}}_a \end{bmatrix} \quad (3)$$

### 3. Análisis Cinemático

En esta sección analizaremos la cinemática directa y la cinemática diferencial de un robot cuadrúpedo. En base a ello, se muestran algunos esquemas de control cinemático de la posición, orientación o ambos a la vez. Como se mencionó anteriormente, en el análisis de un robot cuadrúpedo se debe tener en cuenta el efecto del movimiento libre de su base, por lo que primero se analizará al robot como si su base estuviera fija, y luego se añadirá el efecto de su base flotante.

#### 3.1. Cinemática Directa

El objetivo de la cinemática directa es calcular la posición y orientación del efector final, con respecto a su base fija, como función de las articulaciones actuadas [12]. Supongamos que el marco de referencia de su base se denomina como  $\{\mathcal{B}\}$ , entonces la expresión matemática de la cinemática directa se puede mostrar como:

$${}^B\mathbf{x}_i = \mathbf{f}_i(\mathbf{q}_a) \in SE(3) \quad (4)$$

donde  $i = 1, 2, 3, 4$  corresponde a cada una de las patas del robot, y  $\mathbf{f}_i$  es una función que transforma los valores articulares en la posición  ${}^B\mathbf{p}_i$  y orientación  ${}^B\boldsymbol{\varphi}_i$  del extremo de la pata del robot (efector final). Los métodos más comunes para calcular dicha función es mediante un análisis geométrico de la estructura del robot [12] y el método de Denavit-Hartenberg (DH) [13]. Una vez definido la posición y orientación del efector final, es necesario realizar la transformación de la base a un sistema de referencia inercial e incluir el efecto de la base flotante del robot.

La posición de la  $i$ -ésima pata con respecto al sistema inercial ( ${}^I\mathbf{p}_i$ ), se puede expresar en coordenadas cartesianas como:

$${}^I\mathbf{p}_i = {}^I\mathbf{R}_B(\boldsymbol{\varphi}_b) {}^B\mathbf{p}_i + \mathbf{p}_b \quad (5)$$

donde  ${}^I\mathbf{R}_B(\boldsymbol{\varphi}_b)$  es la matriz de rotación que representa la orientación del sistema de referencia  $\{\mathcal{B}\}$  con respecto al sistema de referencia inercial  $\{\mathcal{I}\}$ , y depende de la orientación de la base

$\varphi_b$ . Por otro lado, la orientación del  $i$ -th pata con respecto al sistema inercial ( ${}^I\mathbf{Q}_i$ ), se puede expresar mediante cuaterniones en el espacio Hamiltoniano como:

$${}^I\varphi_i = \varphi_b {}^B\mathbf{Q}_i \quad (6)$$

donde  $\varphi_b$  se definió en la sección 2.2, y  ${}^B\mathbf{Q}_i$  es el cuaternión que describe la orientación del  $i$ -ésima pata con respecto a la base del robot. Note que la multiplicación en (6) es una multiplicación estándar entre cuaterniones. Por lo tanto, (5) y (6) muestran la cinemática directa de la posición y orientación del robot cuadrúpedo para cualquiera de sus patas.

### 3.1.1. Cinemática directa del Robot Solo

A continuación, se muestra la implementación de la cinemática directa en el robot Solo [3]. Este robot cuenta con una base flotante y 3 *gdl* en cada una de sus patas. Para determinar su cinemática directa se utiliza un análisis geométrico que encuentra la transformación homogénea que relaciona el sistema de referencia del extremo de la pata del robot con el sistema de referencia inercial. En la figura 1a se muestran los sistemas de referencia del robot y en la figura 1b se muestran los parámetros necesarios para obtener las transformaciones homogéneas. Asimismo, la Tabla 1 muestra los valores de los parámetros mostrados en 1b.

Tabla 1: Valores de los parámetros del robot Solo

	$L$	$m$	$n$	$d_2$	$d_3$	$a_2$	$a_3$
Medidas [cm]	19.46	8.75	1.4	3.745	0.8	16	16

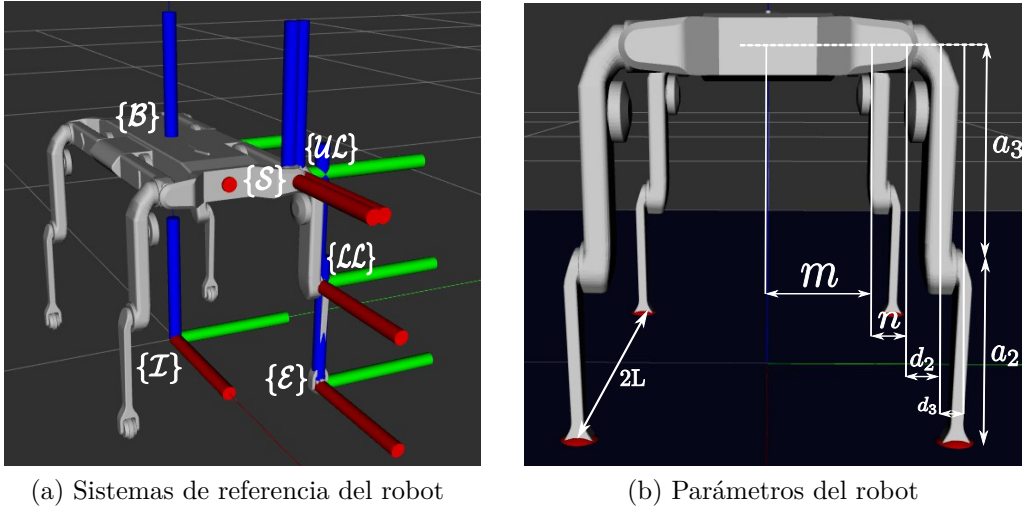


Figura 1: Robot cuadrúpedo Solo

A partir de lo anterior, es posible obtener (4) mediante la relación de las transformaciones homogéneas. A partir de (7) se implementa (5) y (6) realizando las conversiones necesarias.

$${}^B T_E = {}^B T_S {}^S T_{UL} {}^{UL} T_{LL} {}^{LL} T_E \quad (7)$$

Una vez establecidas las ecuaciones, se indican las posiciones articulares de las articulaciones actuadas, y la posición y orientación de la base flotante. En la figura 2a se observa al robot con su base en la posición  $\mathbf{x}_b = [0 \ 0 \ 0.3 \ 1 \ 0 \ 0 \ 0]$  y sus articulaciones  $\mathbf{q}_a = [0 \ -0.5 \ 1 \ 0 \ -0.5 \ 1 \ 0 \ 0.5 \ -1 \ 0 \ 0.5 \ -1]$ , donde las 3 primeras articulaciones corresponden a la pata delantera derecha, las tres siguientes a la pata delantera izquierda, y así sucesivamente con la pata trasera derecha e izquierda. Asimismo, en la figura 2b se tiene  $\mathbf{x}_b = [0.5 \ 0.55 \ 0.3 \ 1 \ 0 \ 0 \ 0]$  y sus articulaciones  $\mathbf{q}_a = [0 \ -0.5 \ 1 \ 0 \ -1 \ 1.5 \ 0 \ 0.5 \ -1 \ 0 \ 0.5 \ -1]$ . Los marcadores rojos indican la posición obtenida mediante la cinemática directa implementadas por (5) y (6). Cabe mencionar que se utiliza el software de Rviz y ROS (*Robot Operating System*) para realizar la visualización del robot y de los marcadores rojos, los cuales indican la posición deseada de las patas del robot.

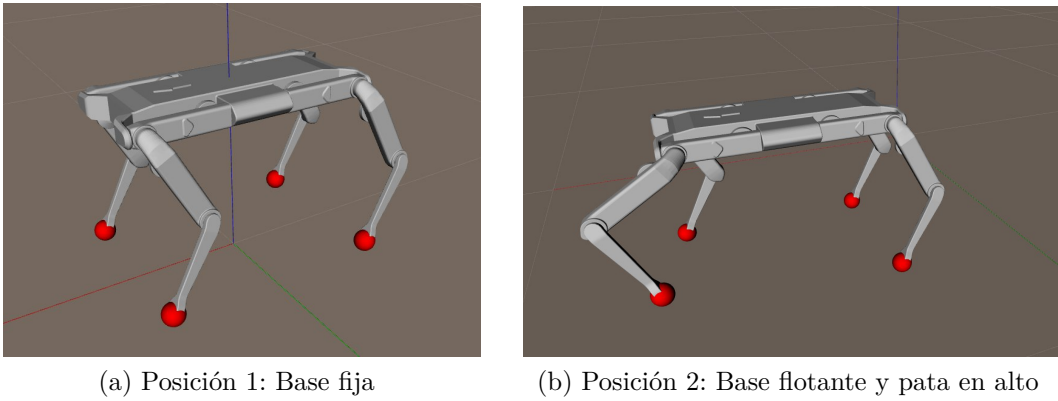


Figura 2: Cinemática directa del robot Solo

### 3.2. Cinemática Diferencial

La Cinemática Diferencial nos brinda la relación entre la velocidad de las coordenadas generalizadas del efector final (*twist*) y la velocidad de las articulaciones actuadas. Este mapeo se describe mediante una matriz, denominada Jacobiano geométrico, que depende de la configuración del robot [12]. Esta definición aplica para robots manipuladores seriales, pero se puede extender el Jacobiano de este tipo de robots a un robot cuadrúpedo añadiendo los términos que representan el efecto de la base flotante. Esto es de gran importancia para el control cinemático, pues nos permite aprovechar la libertad espacial que le brinda la base flotante al robot.

Si la base fuera fija, tanto la posición y velocidad de cada eslabón se representarían bajo el sistema de referencia de su base  $\{\mathcal{B}\}$ , pero dado que ahora esta base es libre, estas se deben expresar bajo el sistema de referencia inercial  $\{\mathcal{I}\}$ . Entonces, consideremos el extremo de la  $i$ -ésima pata del robot, cuya posición con respecto a  $\{\mathcal{I}\}$  está dada por  $\mathbf{p}_i$  y su velocidad lineal como  $\mathbf{v}_i$ , donde ambas  $\in \mathbb{R}^3$ . Usando las relaciones fundamentales de la cinemática, y asumiendo que ningún otro eslabón del robot tiene movimiento, la velocidad lineal de dicho extremo  $i$  está dada por:

$$\mathbf{v}_i = \mathbf{v}_b + \boldsymbol{\omega}_b \times (\mathbf{p}_b - \mathbf{p}_i) = \mathbf{v}_b + \mathbf{S}_p \boldsymbol{\omega}_b \quad (8)$$

donde  $\mathbf{S}_p$  representa la matriz antisimétrica asociada al vector  $\mathbf{p}_b - \mathbf{p}_i$ , de tal modo que:  $\boldsymbol{\omega}_b \times (\mathbf{p}_b - \mathbf{p}_i) = \mathbf{S}_p \boldsymbol{\omega}_b$ , donde  $\boldsymbol{\omega}_b$  es la velocidad angular de la base flotante.

Dado que se asume que ningún otro eslabón del robot tiene movimiento, entonces la velocidad angular de la base se transmite al extremo de la  $i$ -ésima pata del robot. Es decir:

$$\boldsymbol{\omega}_i = \boldsymbol{\omega}_b \quad (9)$$

Cabe mencionar, que si bien (8) y (9) hacen referencia a los extremos de cada pata del robot, estas también aplican para cualquier eslabón de la pata del robot. Así, las ecuaciones pueden expresarse de forma general como  $\mathbf{v}_{j,i} = \mathbf{v}_{j,b} + \mathbf{S}(\boldsymbol{\omega}_{j,b})(\mathbf{p}_{j,b} - \mathbf{p}_{j,i})$  y  $\boldsymbol{\omega}_{j,i} = \boldsymbol{\omega}_{j,b}$ , donde  $j$  indica el  $j$ -ésimo eslabón e  $i$  indica la  $i$ -ésima pata del robot. Sin embargo, dado que nuestro interés está en el efector final, se utilizará la notación de dichas ecuaciones.

Reordenando las ecuaciones (8) y (9), el *twist* con respecto al sistema de referencia  $\{\mathcal{I}\}$  ( $\boldsymbol{\xi}_i$ ) y el *twist* de la base flotante se relacionan como sigue ( $\boldsymbol{\xi}_b$ ):

$$\boldsymbol{\xi}_i = \begin{bmatrix} I & \mathbf{S}_p \\ \mathbf{0} & I \end{bmatrix} \boldsymbol{\xi}_b \quad (10)$$

donde  $I, \mathbf{0} \in \mathbb{R}^3$  son la matriz identidad y una matriz de ceros, respectivamente.

Con el objetivo de determinar la cinemática diferencial del robot, es necesario encontrar la relación entre  $\boldsymbol{\xi}_b$  y  $\dot{\mathbf{x}}_b$ , de tal modo que luego sea posible relacionar  $\boldsymbol{\xi}_i$  con  $\dot{\mathbf{x}}_b$  mediante (10). Dado que utilizamos cuaterniones para expresar la orientación del robot, al derivar (1) con respecto al tiempo, tenemos que:

$$\dot{\mathbf{x}}_b = \begin{bmatrix} \dot{\mathbf{p}}_b \\ \dot{\mathbf{Q}}_b \end{bmatrix} \quad (11)$$

Luego, podemos establecer la relación entre  $\boldsymbol{\xi}_b$  y  $\dot{\mathbf{x}}_b$  como sigue:

$$\boldsymbol{\xi}_b = \begin{bmatrix} \mathbf{v}_b \\ \boldsymbol{\omega}_b \end{bmatrix} = \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & T(\mathbf{Q}_b) \end{bmatrix} \dot{\mathbf{x}}_b \quad (12)$$

donde la relación entre  $\mathbf{v}_b$  y  $\mathbf{p}_b$  es la matriz identidad siempre y cuando se utilicen coordenadas cartesianas para representar la posición. Además,  $T(\mathbf{Q}_b) \in \mathbb{R}^{3 \times 3}$  se muestra en (13) y relaciona la velocidad angular con la derivada del cuaternión de tal modo que  $\boldsymbol{\omega}_b = T(\mathbf{Q}_b)\dot{\mathbf{Q}}_b$  [14], donde  $\dot{\mathbf{Q}}_b = (\dot{w}, \dot{\epsilon}_x, \dot{\epsilon}_y, \dot{\epsilon}_z)$ .

$$T(\mathbf{Q}) = 2 \begin{bmatrix} -\boldsymbol{\epsilon} & wI + \hat{\boldsymbol{\epsilon}} \end{bmatrix} = 2 \begin{bmatrix} -\epsilon_x & w & -\epsilon_z & \epsilon_y \\ -\epsilon_y & \epsilon_z & w & -\epsilon_x \\ -\epsilon_z & -\epsilon_y & \epsilon_x & w \end{bmatrix} \quad (13)$$

Ahora, reemplazando (12) en (10), es posible relacionar la derivada de la posición y orientación del robot y el *twist* del efector final del robot como sigue:

$$\boldsymbol{\xi}_i = J_b \dot{\mathbf{x}}_b \quad (14)$$

donde

$$J_b = \begin{bmatrix} I & \mathbf{S}_p T(\mathbf{Q}_b) \\ \mathbf{0} & T(\mathbf{Q}_b) \end{bmatrix} \quad (15)$$

Dado que este incluye el efecto de la base flotante,  $J_b$  se denomina como el Jacobiano geométrico de la base flotante [15, 8].

Finalmente, para determinar la relación entre el *twist* del efector final de la  $i$ -ésima pata del robot con la velocidad de las coordenadas generalizadas, descrita en (2), está dada por:

$$\xi_i = \begin{bmatrix} I & \mathbf{S}_p T(\mathbf{Q}_b) & {}^I \mathbf{R}_b(\mathbf{Q}_b)^B J_v \\ \mathbf{0} & T(\mathbf{Q}_b) & {}^I \mathbf{R}_b(\mathbf{Q}_b)^B J_\omega \end{bmatrix} \begin{bmatrix} \dot{\mathbf{x}}_b \\ \mathbf{q}_a \end{bmatrix} = J_G(\mathbf{q}) \dot{\mathbf{q}} \quad (16)$$

donde  $J_G$  es el Jacobiano geométrico completo del robot,  ${}^I \mathbf{R}_B(\mathbf{Q}_b)$  es definido como en (5). Note que (16) muestra la cinemática diferencial del robot. Además,  ${}^B J_v$  y  ${}^B J_\omega$  corresponden a los Jacobianos geométricos de los eslabones asociados solamente con las articulaciones actuadas y representados con respecto a la base del robot ( $\{\mathcal{B}\}$ ). Asimismo, se sabe que para las articulaciones de revolución del robot cuadrúpedo se tiene que  $J_{v_j} = \mathbf{z}_{j-1} \times \mathbf{p}_{j-1,N}$  y  $J_{w_j} = \mathbf{z}_{j-1}$  de tal modo que  ${}^B J_v = [J_{v_1} \ J_{v_2} \ \cdots \ J_{v_j}]$  y  ${}^B J_\omega = [J_{w_1} \ J_{w_2} \ \cdots \ J_{w_j}]$ .

### 3.3. Control Cinemático

El movimiento deseado de un robot, con configuración articular  $\mathbf{q}$  y  $n$  grados de libertad, puede ser descrito especificando un conjunto de tareas, definidos como *funciones de tarea* [16]. Se emplea este enfoque de *funciones de tarea*, pues es utilizado ampliamente en robots redundantes, como lo son los robots cuadrúpedos [17, 8, 18]. Para definir correctamente una tarea para el control cinemático, se especifican tres elementos: el Jacobiano de tarea, la función de la tarea y la referencia de la tarea. Luego de ello, se muestra el esquema de control cinemático para la posición de robot. A continuación, se definen como ejemplos las tareas cinemáticas de posición y orientación, y postura. Finalmente, se muestra el control cinemático del robot Solo aplicado a la caminata del mismo.

#### 3.3.1. Jacobiano de la tarea

Las funciones de tarea se representan generalmente en términos de un *Jacobiano de tarea*, también conocido como Jacobiano Analítico ( $J_A$ ), y son usadas para proporcionar una referencia de velocidad para las articulaciones del robot. A continuación, se determina la expresión que calcula  $J_A$ , en función de  $J_G$ .

En (12) se obtuvo la relación entre  $\xi$  y  $\dot{\mathbf{x}}$ , pero esta se puede reordenar de tal modo que se obtenga una expresión más general. Esta viene dada por  $\dot{\mathbf{x}} = E(\mathbf{x})\xi$ , o de forma más específica:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{p}}(\mathbf{q}) \\ \dot{\boldsymbol{\varphi}}(\mathbf{q}) \end{bmatrix} = \begin{bmatrix} E_p(\mathbf{p}) & 0 \\ 0 & E_o(\boldsymbol{\varphi}) \end{bmatrix} \xi \quad (17)$$

donde  $E_p(\mathbf{p})$  es la matriz identidad y  $E_o(\boldsymbol{\varphi})$  es la pseudo-inversa de  $T(\mathbf{Q}_b)$ , la cual se denota como  $T(\mathbf{Q}_b)^\#$  y se calcula como sigue:

$$T(\mathbf{Q})^\# = \begin{bmatrix} -\boldsymbol{\epsilon}^T \\ wI - \boldsymbol{\epsilon} \end{bmatrix} \quad (18)$$



Además de ello, de la definición del Jacobiano Analítico [12], sabemos que  $\dot{\mathbf{x}} = J_A \dot{\mathbf{q}}$ ; y del análisis de la cinemática diferencial, sabemos que  $\dot{\boldsymbol{\xi}} = J_G(\mathbf{q}) \dot{\mathbf{q}}$ . Reemplazando en (17), se obtiene que  $J_A(\mathbf{q}) = E(\mathbf{x}) J_G(\mathbf{q})$ , o de forma más específica:

$$\begin{bmatrix} J_p(\mathbf{q}) \\ J_o(\mathbf{q}) \end{bmatrix} = \begin{bmatrix} E_p(\mathbf{p}) & 0 \\ 0 & E_o(\boldsymbol{\varphi}) \end{bmatrix} \begin{bmatrix} J_v(\mathbf{q}) \\ J_\omega(\mathbf{q}) \end{bmatrix} \quad (19)$$

donde  $J_p$  y  $J_o$  corresponden a los términos de posición y orientación del Jacobiano Analítico; mientras que  $J_v$  y  $J_\omega$ , a los términos de velocidad lineal y angular del Jacobiano Geométrico, respectivamente.

La ecuación (19) nos indica que el Jacobiano de tarea puede ser calculado directamente de (16) utilizando una representación adecuada de la orientación ( $E_o(\boldsymbol{\varphi})$ ), a pesar de que la definición de  $J_A$  dependa directamente de la posición y orientación del robot.

### 3.3.2. La función de la tarea

Sea una tarea  $i$  denotada por  $e_i$ , y el Jacobiano de dicha tarea como  $J_i = \frac{\partial e_i}{\partial \mathbf{q}}$ , de modo que:

$$\dot{e}_i = J_i \dot{\mathbf{q}} \quad (20)$$

donde  $\dot{e}_i$  es la velocidad de la tarea, y  $\dot{\mathbf{q}}$  se define como (2).

Representemos como  $\mathbf{s}$  la configuración actual en un espacio de tarea dado, y la configuración deseada como  $\mathbf{s}^*$ . Entonces, la tarea  $e_i$  se define como la diferencia entre ambas configuraciones:

$$\mathbf{e}_i = \mathbf{s} \ominus \mathbf{s}^* \quad (21)$$

donde  $\ominus$  es el operador de diferencia en el espacio de la tarea. Esta definición lo convierte en un problema de regulación, es decir, de lograr que el efector final del robot alcance un punto deseado [12].

### 3.3.3. Referencia de la tarea

La referencia de la tarea está dada por:

$$\dot{\mathbf{e}}^* = -\lambda \mathbf{e}, \quad \lambda > 0 \quad (22)$$

donde  $\lambda$  es la ganancia cinemática de la tarea y garantiza la estabilidad asintótica del error a 0. Esto debido a su comportamiento exponencial decreciente.

### 3.3.4. Esquemas de Control y de la función de tarea

El esquema de control que se utilizará para realizar el control de posición del robot es el solucionador de cinemática inversa ponderada (SCIP) definido en [17]. Este esquema recibe como entradas las tareas cinemáticas ( $t$ ) que definen el movimiento deseado y devuelve la velocidad articular generalizada que mejor satisface el siguiente problema de optimización:

$$\min_{\dot{\mathbf{q}}_a} \left\{ \sum_{i=1}^t w_i \|\dot{e}_i - J_i \dot{\mathbf{q}}_a\|^2 \right\} \quad (23)$$

Esto indica que las tareas con mayor peso se satisfacen en mayor medida, pero puede ocurrir que existan tareas conflictivas que no se satisfagan por completo, pero que exista un equilibrio entre ellas.

En base a [17], se grafica el esquema de control en el cual se incluye el esquema de la función de tarea (ver Figura 3). Como se mencionó, el SCIP recibe como entradas las tareas cinemáticas, las cuales pueden ser varias, sin embargo, en el esquema se muestra solo uno por simplicidad.

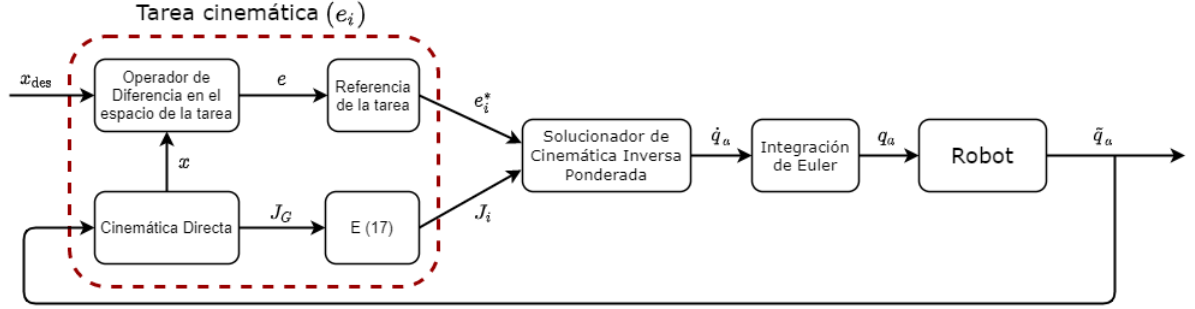


Figura 3: Esquema de control cinemático y de la función de tarea

Como se muestra en la Figura 3, la salida del controlador cinemático son las velocidades articulares. Para ser capaces de controlar al robot, mediante la posición articular, es necesario realizar algún método de integración, de los cuáles se escoge el método de Euler por ser el más sencillo de implementar y porque brinda buenos resultados mientras el tiempo de integración ( $\Delta T$ ) sea pequeño. Por lo general, va desde 1 ms a 10 ms [8]. En ese sentido, el esquema de integración es:

$$\mathbf{q}_a[k+1] = \mathbf{q}_a[k] + \Delta T \dot{\mathbf{q}}_a[k] \quad (24)$$

donde  $\mathbf{q}_a[k]$  y  $\mathbf{q}_a[k+1]$  representa el valor de  $\mathbf{q}_a$  en el tiempo  $k$  y un instante después, respectivamente. Luego, la salida de (24) va directamente a las articulaciones del robot.

Finalmente, se retroalimenta el sistema midiendo la posición y orientación de la base flotante del robot y las posiciones articular del robot. En la práctica, la posición y orientación de la base flotante se obtiene integrando las mediciones obtenidas por una Unidad de Medición inercial (IMU), y las posiciones articulares mediante encoders magnéticos.

Sin embargo, si es que las mediciones del IMU no se encuentran disponibles, es posible realizar un esquema de integración similar al de (24) para obtener la posición de la base flotante. Para ello, reemplazamos  $\mathbf{q}_a$  por  $\mathbf{x}_b$  y  $\dot{\mathbf{q}}_a$  por  $\dot{\mathbf{x}}_b$ . Con respecto a la orientación de la base flotante, el método de integración en (24) no es suficiente cuando se trabaja con cuaterniones, puesto que no se puede asegurar que se obtengan cuaterniones unitarios. De acuerdo con [8], una solución usar la representación eje/ángulo de tal modo que:

$$\Delta Q = (\cos(\Delta\theta\Delta T/2), \mathbf{u}\sin(\Delta\theta\Delta T/2)) \quad (25)$$

donde  $\Delta T$  es el tiempo de control,  $\Delta\theta$  representa la magnitud del ángulo de rotación ( $\|\omega\|$ ),  $\mathbf{u} = \frac{\omega}{\|\omega\|}$ , de modo que  $\omega_b = T(Q)\dot{Q}_b$  [14]. A partir de (25), se puede obtener el cuaternión

de la orientación como la multiplicación de dos cuaterniones:

$$\mathbf{Q}[k+1] = \Delta\mathbf{Q}[k]\mathbf{Q}[k] \quad (26)$$

donde  $\mathbf{Q}[k]$  y  $\mathbf{Q}[k+1]$  representa el valor de  $\mathbf{Q}$  en el tiempo  $k$  y un instante después, respectivamente.

### 3.3.5. Ejemplos de funciones de tareas

A continuación, se muestran algunos ejemplos de tareas cinemáticas, como el control de posición, orientación, posición y orientación, y postura. En cada una de estas tareas se definirá la función de tarea, se determinará su jacobiano y se utilizará la referencia de tarea establecida en 3.3.3.

#### A. Tarea de posición

El objetivo de esta tarea es mover un punto operacional del robot, en nuestro caso el extremo de la pata del robot, a una posición deseada. Dado que solo se requiere llegar a la posición deseada, no es necesario considerar la orientación. Entonces, definimos  $\mathbf{p}(\mathbf{q})$  como la posición actual del extremo de una de las patas del robot, y además definimos  $\mathbf{p}_d$  como la posición deseada. Al igual que en los análisis anteriores, se utilizan las coordenadas cartesianas para representar  $\mathbf{p}(\mathbf{q})$  y  $\mathbf{p}_d$  en  $\mathbb{R}^3$ . Por lo tanto, el error de posición viene dado por:

$$\mathbf{e} = \mathbf{p}_d - \mathbf{p}(\mathbf{q}) \quad (27)$$

Donde  $\mathbf{p}_d$  se calcula como en (5). Además, dado que solo se hace uso de la posición, el jacobiano de la tarea viene dado por el término correspondiente a la velocidad lineal, es decir:

$$\mathbf{J} = \begin{bmatrix} \mathbf{I} & \mathbf{S}_p \mathbf{T}(\mathbf{Q}_b) & {}^I \mathbf{R}_b(\mathbf{Q}_b)^B \mathbf{J}_v \end{bmatrix} \quad (28)$$

Finalmente, la tarea de posición queda completamente definida por (27), (28) y (22).

#### B. Tarea de orientación

El objetivo de esta tarea es rotar un punto operacional del robot, en nuestro caso el extremo de la pata del robot, de tal modo que alcance la orientación deseada. Dado que solo se requiere llegar a la orientación deseada, no es necesario considerar la posición. Al igual que en los análisis previos, se utilizan los cuaterniones unitarios para representar la orientación del robot. Entonces, definimos  $\mathbf{Q} = (w, \boldsymbol{\epsilon})$  como la orientación actual del extremo de una de las patas del robot, y además definimos  $\mathbf{Q}_d = (w_d, \boldsymbol{\epsilon}_d)$  como la orientación deseada.

El error de la orientación de  $\mathbf{Q}$  a  $\mathbf{Q}_d$ , en el espacio  $\mathbb{H}$ , se define como:  $\mathbf{Q}_e = \mathbf{Q}_d \mathbf{Q}^{-1}$ , donde  $\mathbf{Q}_e$  es el cuaternión que representa el error en la orientación y será igual a  $(1, \mathbf{0})$  cuando  $\mathbf{Q}$  y  $\mathbf{Q}_d$  sean iguales [14]. Entonces, dado que la identidad representa un error de 0, la tarea de la orientación  $\mathbf{e}$  puede definirse en términos de  $\mathbf{Q}_e = (w_e, \boldsymbol{\epsilon}_e)$  como sigue:

$$\mathbf{e} = \begin{bmatrix} w_e - 1 \\ \boldsymbol{\epsilon}_e \end{bmatrix} \quad (29)$$

Por otro lado, el Jacobiano  $J(\mathbf{q})$  para esta tarea describe la relación entre la tasa de cambio del cuaternión y la velocidad articular, es decir es igual a  $J_o(\mathbf{q})$  en 19. De forma más específica:

$$J(\mathbf{q}) = T(\mathbf{Q})^\# J_\omega(\mathbf{q}) \quad (30)$$

donde  $T(\mathbf{Q})^\#$  se define en 18 y  $J_\omega(\mathbf{q})$  como la segunda fila de (19).

Finalmente, la tarea de posición queda completamente definida por (29), (30) y (22).

### C. Tarea de posición y orientación

Esta tarea consiste en controlar tanto la posición como la orientación (a la vez) de un punto operacional dado, en nuestro caso el extremo de una pata del robot. En ese sentido, la tarea de la posición y orientación  $\mathbf{e}$  puede ser representada como:

$$\mathbf{e} = P^* \ominus P \quad (31)$$

donde  $P \in SE(3)$  es una representación de la posición y orientación del robot,  $P^*$  es la posición y orientación deseada y  $\ominus$  es el operador de diferencia. Dado que utilizamos coordenadas coortesianas para representar la posición, y cuaterniones, para la orientación, el error es definido como en (27) y (29). Asimismo, el Jacobiano de la tarea se determinó en (19). Por lo tanto, la tarea de posición y orientación  $\mathbf{e}$  queda completamente definida por (31), (19) y (22).

### D. Tarea de postura

El término postura, en el espacio articular, denota una configuración articular específica de todos o un subconjunto de las articulaciones presentes en el robot. Sea  $\mathbf{q}_s = \{q_i, q_j, \dots, q_k\}$ , donde  $0 \leq k \leq n - 1$ , un subconjunto arbitrario de las articulaciones del robot, y sea  $\mathbf{q}_d$  la postura deseada del robot. Entonces la tarea de postura queda definida como sigue:

$$\mathbf{e} = \tilde{\mathbf{q}}_d - \tilde{\mathbf{q}} \quad (32)$$

donde  $\mathbf{e}$  es el vector de la tarea de postura, cuyo tamaño depende de la cantidad de articulaciones que serán controladas.

Con respecto al Jacobiano de la tarea ( $J(\mathbf{q})$ ), dado que esta tarea se define en el espacio articular, entonces el jacobiano asociado tiene la siguiente forma:

$$J(\mathbf{q}) = [0_b \quad J_q] \quad (33)$$

donde  $0_b$  es una matriz de ceros correspondiente a la base flotante (parte subactuada), cuyo tamaño es de  $1 \times 7$  debido a la representación de la misma, y  $J_q$  es un vector de unos correspondiente a las articulaciones actuadas.

Por lo tanto, la tarea de *postura*  $\mathbf{e}$  queda completamente definida por (32), (33) y (22).

### 3.3.6. Control Cinemático del robot Solo

La implementación del control cinemático consiste básicamente en crear una serie de funciones que implementen el jacobiano de la tarea, la función de la tarea, y la referencia de las

tareas mostradas en 3.3.5 para cada una de las patas del robot Solo. En esta sección, se implementará la tarea de posición tanto para las articulaciones actuadas como la base flotante, de tal forma que se pueda visualizar la caminata del robot y el control del movimiento de una pata.

### A. Caminata del Robot

En la figura 4 se puede observar una serie de figuras que muestran la caminata del robot. Para lograr ello, se inicializó la posición del robot como en 2a y se procedió a realizar primero el movimiento de la pata delantera izquierda junto al movimiento de la base (figura 4a). Luego, se realizó el movimiento de la pata trasera izquierda (figura 4b). Paso seguido, se realiza el movimiento similar de las patas derechas del robot (figuras 4c y 4d). Y el movimiento se repite de forma secuencial (figuras 4e y 4f). También se puede observar, en las figuras 4g, 4h y 4i, el movimiento luego de que el robot da unos pasos. Cabe mencionar que se utilizó un valor de  $\lambda = 4.0$  en (22) y un valor de  $w = 1.0$  en (23) para cada una de las tareas.

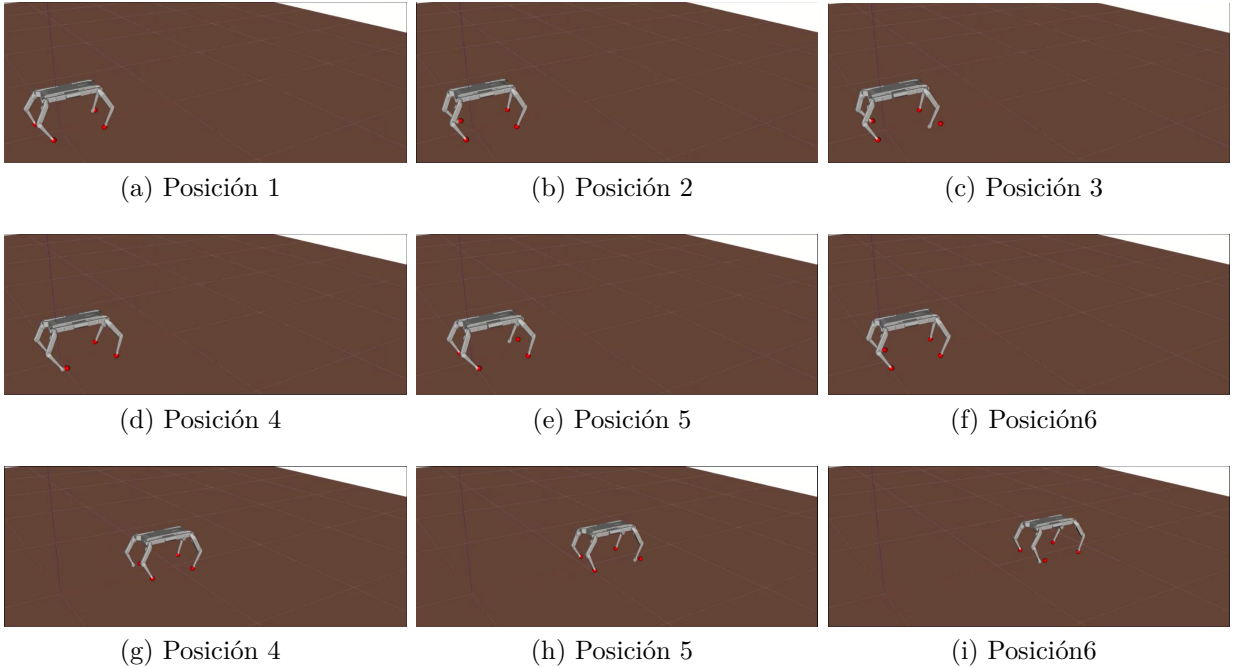


Figura 4: Control cinemático de la caminata del robot

### B. Movimiento de una pata

En la figura 5 se puede observar una serie de figuras que muestran cómo se levanta la pata derecha del robot hasta cierta posición deseada. En este caso, la base no está fija, sino que puede desplazarse.

En la figura 5a se muestra la pata derecha levantándose del suelo y continúa su trayectoria como se muestra en las figuras 5b y 5c, para finalmente llegar a su posición deseada (figura 5d). Se debe notar que se ha realizado un desplazamiento de la base

---

desde su posición inicial (figura 5a) hasta su posición final (figura 5d). Cabe mencionar que para cada una de las tareas se utilizaron los mismos valores de  $\lambda$  y  $w$  que en la caminata del robot.

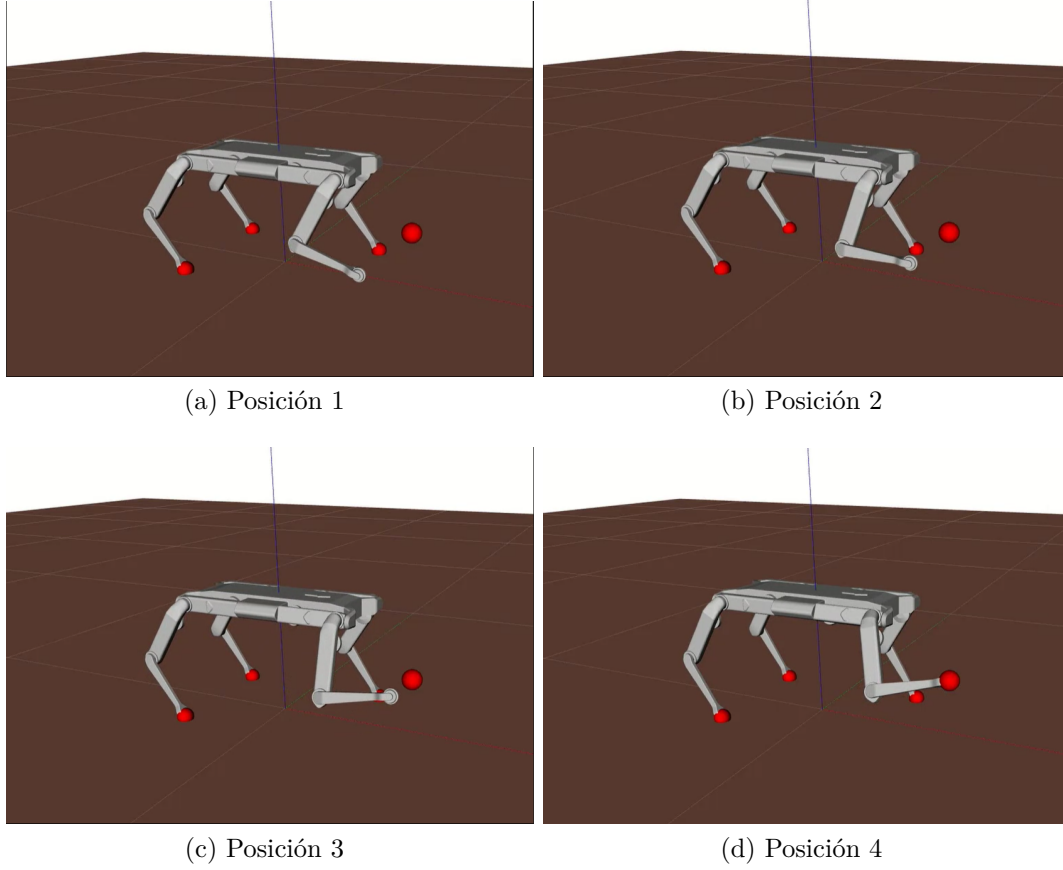


Figura 5: Control cinemático de la caminata del robot

El análisis cinemático de las articulaciones actuadas y la base flotante nos ha permitido tener un primer acercamiento al movimiento de los robots cuadrúpedos, desde la cinemática directa de sus patas hasta el control cinemático del robot. En el contexto del análisis cinemático, el movimiento se puede mejorar generando la trayectoria de cada pata o incluso de cada articulación, pero esto puede llegar a ser un proceso muy tedioso y sensible a externalidades. Además, si bien se logra controlar la caminata del robot y el movimiento de la pata de forma manual, esta no se visualiza de forma totalmente natural, puesto que no se simulan las fuerzas externas ni se modela el contacto físico entre las patas del robot y el suelo. Para solucionar ello es necesario realizar un análisis de la dinámica del robot, la cual se presenta en la siguiente sección.

---

## 4. Análisis Dinámico

En esta sección se analizará el modelo dinámico de un robot con base flotante, la cual toma en cuenta las fuerzas del entorno sobre el robot para realizar el movimiento del robot. Además se muestra cómo se mantiene la estabilidad de un robot cuadrúpedo durante la caminata y un esquema de control utilizando un método de lazos cerrados en cascada y el modelo dinámico del robot.

### 4.1. Modelo dinámico de un robot con base flotante

La dinámica de los robots con base flotante se considera más compleja que una de un robot con base fija ya que se debe tener en cuenta, aparte de los torques que genera cada motor, las fuerzas del entorno hacia el robot para poder controlar la posición y orientación de la base flotante de manera correcta. Asimismo como se mencionó en la sección 2, la base del robot se controla indirectamente a través del movimiento articular de los motores y las reacciones que el entorno genera sobre el robot. Esto se representa de la misma forma que en la ecuación 2

$$\mathbf{q} = \begin{bmatrix} \mathbf{x}_b \\ \mathbf{q}_a \end{bmatrix} \in \mathbb{R}^{d+7} \quad (34)$$

donde  $\mathbf{x}_b \in \mathbb{R}^7$  representa a la posición y orientación de la base flotante y  $\mathbf{q}_a \in \mathbb{R}^d$  representa las  $d$  articulaciones actuadas en el robot que se encuentran en las extremidades y controlan el movimiento del sistema. Como los términos de  $\mathbf{x}_b$  no están relacionados directamente a un actuador, en la bibliografía [19] se suele usar la matriz de selección  $\mathbf{S}$  junto con el vector de torques  $\boldsymbol{\tau}$  del robot:

$$\mathbf{S}\boldsymbol{\tau} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \boldsymbol{\tau} = \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\tau}_a \end{bmatrix} \quad (35)$$

Donde  $\boldsymbol{\tau}_a$  son los torques generados por los actuadores en las articulaciones. Teniendo en cuenta lo dicho anteriormente y las reacciones de las superficies que estan en contacto con el robot, el modelo dinámico de un robot con base flotante se representa como:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) = \mathbf{S}\boldsymbol{\tau} + \sum_i \mathbf{J}_i^T \boldsymbol{\lambda}_i \quad (36)$$

donde  $\mathbf{M}$ ,  $\mathbf{C}$ ,  $\mathbf{G}$ ,  $\mathbf{J}$  y  $\boldsymbol{\lambda}$  son la matriz de inercia, fuerzas de Coriolis y centrífugas, fuerzas gravitacionales, Jacobiano geométrico del efector final y fuerzas externas aplicadas, respectivamente.

### 4.2. Dinámica inversa con el enfoque de *funciones de la tarea*

Las funciones de la tarea presentadas en la sección 3.3 se pueden extender al control por dinámica inversa. El control por dinámica inversa se realiza considerando los torques

generalizados y la configuración articular del robot, la cual se describe por su segunda derivada  $\ddot{q}$ . Por esta razón, se debe encontrar una relación entre la aceleración de la tarea  $\ddot{e}$  y la aceleración articular  $\ddot{q}$  si se quiere utilizar el enfoque de funciones de la tarea. Esta relación se obtiene derivando la ecuación (20).

$$\ddot{e}_i = \dot{J}_i \dot{q} + J_i \ddot{q} \quad (37)$$

Se puede reemplazar  $J_i \ddot{q}$  en la ecuación del modelo dinámico del robot para obtener la expresión de dinámica inversa relacionando el espacio de la tarea con el control. Para el modelo dinámico se tomará en cuenta el usado en [20] el cual considera  $\mathbf{b} = \mathbf{C} + \mathbf{G}$  de nuestro modelo y no considera el efecto de las fuerzas externas. La expresión obtenida es la siguiente:

$$\ddot{e} + (J\mathbf{M}^{-1}\mathbf{b} - \dot{J}\dot{q}) = J\mathbf{M}^{-1}\tau \quad (38)$$

Esta ecuación tiene la misma estructura de la formulación genérica de la función de la tarea  $r + \delta = Gu$  el cual se encuentra mayor explicado en [20]. Es este caso  $\delta = J\mathbf{M}^{-1}\mathbf{b} - \dot{J}\dot{q}$ , la entrada de control es el vector de torques  $u = \tau$ , el mapa  $G = J\mathbf{M}^{-1}$ , la cual es una función del Jacobiano.

Finalmente el problema se reduce en encontrar el torque de control deseado  $\tau^*$  que va a generar la referencia de  $\ddot{e}^*$  usando las aceleraciones articulares necesarias. La ecuación genérica para la ley de control es:

$$u^* = G^\#(r^* + \delta) + P_G u_2 \quad (39)$$

Reemplazando esta ecuación con los valores obtenidos anteriormente se convierte en:

$$\tau^* = (J\mathbf{M}^{-1})^\#(\ddot{e}^* + J\mathbf{M}^{-1}\mathbf{b} - \dot{J}\dot{q}) + P_{J\mathbf{M}^{-1}} \bar{\tau} \quad (40)$$

donde  $P_{J\mathbf{M}^{-1}}$  es el proyector en el espacio nulo de  $J\mathbf{M}^{-1}$  y  $\bar{\tau}$  es un vector arbitrario que puede ser usado para controlar más tareas de baja prioridad.

### 4.3. Estabilidad de caminata

En robot con más de dos extremidades se puede garantizar la estabilidad de la caminata del robot, sin tomar en cuenta los efectos inerciales del cuerpo y las piernas, siempre y cuando el centro de masa (CoM) del robot se encuentre proyectado en el polígono de soporte del robot. Este polígono se define por las piernas del robot que se mantienen en contacto con el piso mientras las otras piernas se mueven para realizar el movimiento.

Dentro del polígono de soporte se puede definir un margen de estabilidad el cual puede ser: a) la distancia mínima desde la proyección del centro de masa a uno de los bordes del polígono; b) la distancia mínima desde la proyección del centro de masa a los bordes del polígono en la proyección del movimiento del robot. También llamado margen de estabilidad longitudinal. Esto se explica mejor en la figura 7.



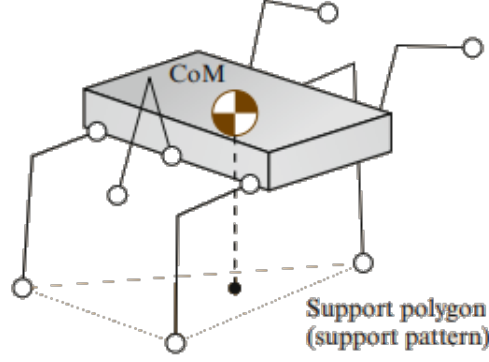


Figura 6: Polígono de soporte de un robot con múltiples patas [21]

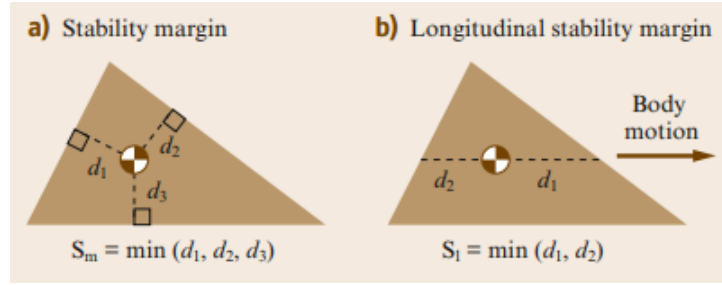


Figura 7: Márgenes de estabilidad de un robot cuadrúpedo [21]

#### 4.4. Ejemplo de esquema de control

En la figura 8 la generación de trayectoria del CoM se encuentra en el bloque de Motion planner, y es así como genera aceleración, velocidad y posición deseados para las articulaciones del robot. La aceleración se envía a un bloque de dinámica inversa, el cual también recibe la posición y velocidad actuales del robot, y las fuerzas de contacto con el suelo; para generar un valor de torque  $\tau_{ff}$ . Por otro lado la velocidad y posición deseada de las articulaciones se restan con los valores actuales del robot para dar paso al control de posición PID y generar otro valor de torque  $\tau_{fb}$ . Estos dos torques se suman para generar el valor de torque deseado  $\tau_d$  que se envía al control de torque de bajo nivel.

Dentro del esquema de control de torque de bajo nivel se encuentra un bloque de control PID de torque que tiene como entrada el error del torque deseado y el medido. También hay un bloque feed-forward compensador de eficiencia, que tiene como salida corriente  $i_{ff}$  y se rige por la siguiente ecuación:

$$i_{ff}(\tau_d) = \frac{1}{NK_\tau} \frac{\tau_d}{\mu_\alpha \mu_k} \quad (41)$$

donde  $N$  es el gear ratio;  $K_\tau$ , la constante de torque en Nm/A  $\mu_\alpha$ , el coeficiente de temperatura/cambio de velocidad; y  $\mu_k$ , el coeficiente de carga de torque. Los dos últimos se pueden encontrar en las fichas técnicas de los motores. Por último también hay un bloque de compensación de fricción de los motores, la cual tiene como entrada velocidad de giro del

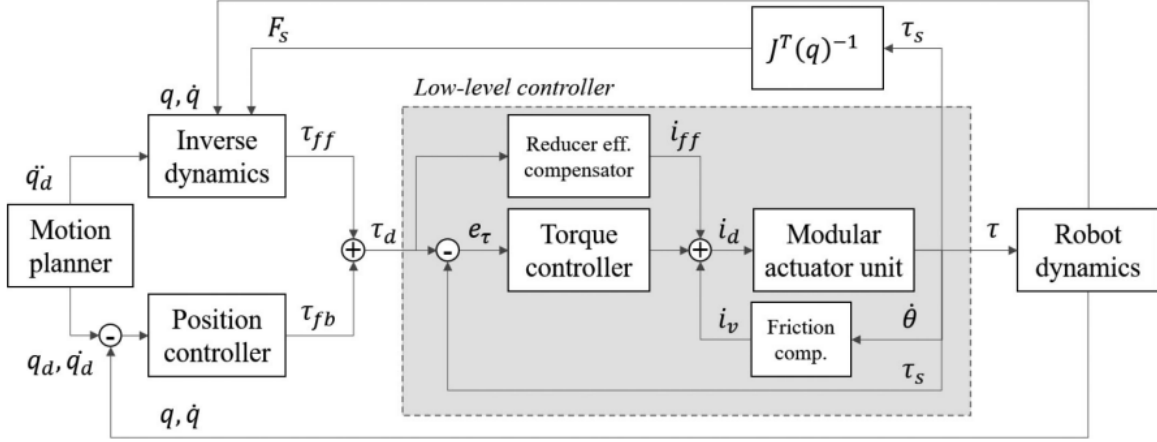


Figura 8: Esquema de control [22]

motor y como salida la corriente  $i_v$ , y se rige por la siguiente ecuación:

$$i_v(\dot{\theta}) = \frac{1}{NK_\tau} \left( f_c + \frac{f_s - f_c}{1 + (\dot{\theta}/\dot{\theta}_s)^2} + f_v \dot{\theta} \right) \quad (42)$$

donde  $f_c$  es la fricción de Coulomb;  $f_s$ , stiction (fricción estática?);  $f_v$ , fricción viscosa; y  $\dot{\theta}_s$  es la velocidad característica de la curva de Stribeck.

La suma de las corrientes:  $i_{ff}$ ,  $i_v$  y la salida del control PID de torque, producen la corriente deseada  $i_d$  la cual es la entrada real al motor.

## 4.5. Generación de trayectorias con ZMP

### 4.5.1. Cálculo del ZMP

En [22] proponen un planeador de trayectoria que utiliza el Zero-Moment Point (ZMP) para generar la referencia del centro de masa del robot. El ZMP se puede calcular utilizando dos métodos distintos. El primer método es conocido como el modelo cart-table [23] en donde la posición del ZMP se obtiene de la siguiente forma:

$$p_x = x_c - \frac{z_c}{g} \ddot{x}_c \quad (43)$$

Donde  $p_x$  es la posición del ZMP el eje  $x$ ;  $x_c$  y  $\ddot{x}_c$  la posición y aceleración del CoM en el eje  $x$ ;  $z_c$  la posición del CoM en el eje  $z$ ; y  $g$  es la aceleración de la gravedad. Para obtener la posición del ZMP en la dirección  $y$  se realiza de la misma forma. Este método tiene una desventaja: cuando la aceleración del CoM es cero, el ZMP corresponde a la proyección del CoM. Esta característica obliga al robot a estar planeando la posición del ZMP constantemente durante su recorrido y puede causar inestabilidad en terrenos irregulares.

El segundo método se denomina como ZMP directo basado en fuerza. Para el uso de este método se requiere el control de la fuerza ejercida sobre el piso. La expresión para el cálculo del ZMP es:

$$\mathbf{p}_{ZMP} = \begin{bmatrix} \mathbf{p}_{ZMP}^x \\ \mathbf{p}_{ZMP}^y \end{bmatrix} = \frac{\sum \mathbf{p}_i^{CoM} \times \mathbf{F}_i}{\sum \mathbf{F}_i} \quad (44)$$

donde  $\mathbf{p}_{ZMP}$ ,  $\mathbf{p}_i^{CoM}$  y  $\mathbf{F}_i$  son la posición del ZMP, la posición del contacto con el piso al centro de masa del robot y la fuerza de reacción del piso, respectivamente. Este punto se encuentra siempre en el polígono de soporte, explicado en la sección 4.3, debido a la siguiente restricción:

$$\mathbf{r}_l < \mathbf{p}_{ZMP} - \boldsymbol{\delta} < \mathbf{r}_h$$

donde  $\mathbf{r}_l$  y  $\mathbf{r}_h$  son límites del espacio del polígono de soporte, y  $\boldsymbol{\delta}$  es la compensación para el margen de estabilidad.

#### 4.5.2. Sistema de Control Preliminar (Preview Control)

Esta técnica de control utiliza información futura del robot para generar la trayectoria en el presente [24]. En la figura 9 se puede ver el esquema de retroalimentación donde un controlador recibe como entrada el error del ZMP de referencia  $p^{ref}$  y el ZMP real medido  $p$ . Este controlador genera la ley de control  $u$  la cual es la entrada al sistema representado como el modelo cart-table mencionado en la sección anterior. Este sistema tiene como salida la posición actual del ZMP, además de la posición, velocidad y aceleración del CoM (lo cual no se ve en la figura).

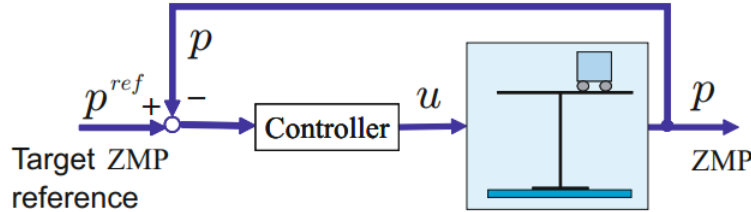


Figura 9: Esquema de Control del ZMP [24]

Para utilizar esta técnica se debe discretizar el sistema de la figura 9 usando un tiempo de muestreo  $\Delta t$  de la siguiente forma:

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{b}u_k \\ p_k = \mathbf{c}\mathbf{x}_k \end{cases} \quad (45)$$

donde

$$\mathbf{x}_k \equiv [x(k\Delta t) \quad \dot{x}(k\Delta t) \quad \ddot{x}(k\Delta t)]^T, \\ u_k \equiv u(k\Delta t),$$

---


$$p_k \equiv p(k\Delta t),$$

$$\mathbf{A} \equiv \begin{bmatrix} 1 & \Delta t & \Delta t^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{b} \equiv \begin{bmatrix} \Delta t^3/6 \\ \Delta t^2/2 \\ \Delta t \end{bmatrix}$$

$$\mathbf{c} \equiv [1 \quad 0 \quad -z_c/g]$$

Para que la salida  $p_k$  siga a la referencia de ZMP  $p_k^{ref}$  se busca minimizar la siguiente expresión de desempeño:

$$J = \sum_{j=1}^{\infty} \left\{ Q(p_j^{ref} - p_l)^2 + Ru_j^2 \right\} \quad (46)$$

donde Q y R son ganancias positivas. Este desempeño se puede minimizar con la siguiente entrada al sistema la cual utiliza  $N$  referencias futuras del ZMP:

$$u_k = -\mathbf{K}x_k + [f_1, f_2, \dots, f_N] \begin{bmatrix} p_{k+1}^{ref} \\ \vdots \\ p_{k+N}^{ref} \end{bmatrix} \quad (47)$$

donde

$$\mathbf{K} \equiv (R + \mathbf{b}^T \mathbf{P} \mathbf{b})^{-1} \mathbf{b}^T \mathbf{P} \mathbf{A} \quad (48)$$

$$f_i \equiv (R + \mathbf{b}^T \mathbf{P} \mathbf{b})^{-1} \mathbf{b}^T (\mathbf{A} - \mathbf{b} \mathbf{K})^{T*(i-1)} \mathbf{c}^T Q \quad (49)$$

La matriz  $\mathbf{P}$  es la ecuación de Riccati expresada de la siguiente manera:

$$\mathbf{P} = \mathbf{A}^T \mathbf{P} \mathbf{A} + \mathbf{c}^T Q \mathbf{c} - \mathbf{A}^T \mathbf{P} \mathbf{b} (R + \mathbf{b}^T \mathbf{P} \mathbf{b})^{-1} \mathbf{b}^T \mathbf{P} \mathbf{A} \quad (50)$$

$\mathbf{K}$  es una ganancia y la matriz de  $f_i$  los pesos que tendran cada una de las referencias futuras (mayor explicación sobre estos términos se encuentran en [24]). De esta expresión se puede deducir que la ley de control incluye un término de retroalimentación y otro que utiliza las muestras futuras de referencias con sus respectivos pesos.

Posteriormente en [24], [23] se propone otra una mejora para esta técnica de control para resolver un mal seguimiento del ZMP con offser que se presentaba en la ecuación 47 en largas caminatas. La expresión de la ley de control propuesta es:

$$\mathbf{u}(k) = -K_s \sum_{i=0}^k (p_j^{ref} - p_j) - K_x \mathbf{x}_k + \sum_{j=1}^N g_j p_{k+j}^{ref} \quad (51)$$

El procedimiento para llegar a esta expresión y definir los términos  $K_s$ ,  $K_x$  y  $g_j$  se encuentran en la bibliografía indicada. El nuevo esquema de control propuesto para este control

preliminar mejorado se encuentra en la figura 10. Aquí se puede observar que ahora el controlador requiere de más entradas que antes. Se requiere una entrada de  $N$  referencias futuras del ZMP, el error de la referencia actual y la posición medida del ZMP, y la posición del CoM; para generar la ley de control  $u$ .

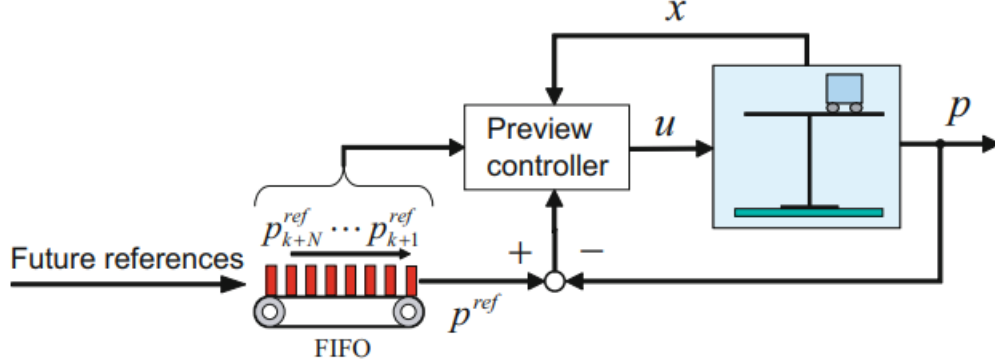


Figura 10: Esquema de Control de ZMP mejorado [24]

Por último, en [25] proponen otra forma de obtener la ley de control con la ecuación (44) y se expresa de la siguiente forma:

$$\mathbf{u}(k) = -k_j \sum_{j=0}^k \mathbf{e}(j) - \sum_{r=1}^N G(r) \mathbf{p}_{ZMP,d}(k+r) - \mathbf{f}_c \quad (52)$$

donde  $k_j$  y  $G(r)$  son ganancias,  $\mathbf{e}(j) = \mathbf{p}_{ZMP}(j) - \mathbf{p}_{ZMP,d}(j)$  es el error del ZMP de referencia y el actual y  $\mathbf{f}_c$  es un término para disminuir la distancia entre el ZMP y la trayectoria de referencia del CoM.

#### 4.5.3. Simulación del preview control

Con el objetivo de probar el método presentado se utiliza el software de Matlab para simular el control de posición del ZMP y la generación de trayectoria del CoM. Se define el sistema discretizado de la ecuación (45) y se consideran los siguientes valores para el sistema  $\Delta t = 0.005s$ ,  $z_c = 0.3m$  (distancia del suelo a la base del robot Solo cuando se encuentra como en 2a) y  $g = 9.81m/s$ . El Toolbox de control de sistemas que proporciona Matlab brinda una función llamada `dlqr()` la cual calcula las matrices  $\mathbf{K}$  y  $\mathbf{P}$  de las ecuaciones (48) y (50), respectivamente. Los argumentos de esta función son las matrices  $\mathbf{A}$  y  $\mathbf{b}$  del sistema, así como las ganancias  $Q^*$  y  $R = 1e^{-6}$ . Cabe resaltar que en [24] consideran a la ganancia  $Q$  como  $Q^* = \mathbf{c}^T Q \mathbf{c}$  (donde  $Q = 1$ ), así que esa matriz es la que se ingresa a la función. Una vez obtenidas las matrices  $\mathbf{K}$  y  $\mathbf{P}$  es posible generar los pesos  $f_i$  de la ecuación (49). Finalmente se genera la ley de control de la ecuación (47) capaz de minimizar el error de la posición del ZMP con respecto a la referencia en el sistema.

Los resultados para 7 segundos de simulación se muestran en la figura 11. Para esta simulación se utilizaron trayectorias de referencias del ZMP diferentes para las direcciones  $x$  y  $y$ , tal y como en [24], con el fin de demostrar que el sistema es capaz de seguir distintas

trayectorias de manera exitosa. También se puede observar en las gráficas que la posición del CoM tiene un comportamiento de seguimiento adecuado y movimientos suaves. Debido al método de preview control, el CoM prevee el cambio en la referencia del ZMP y comienza a moverse antes de que la referencia cambie.

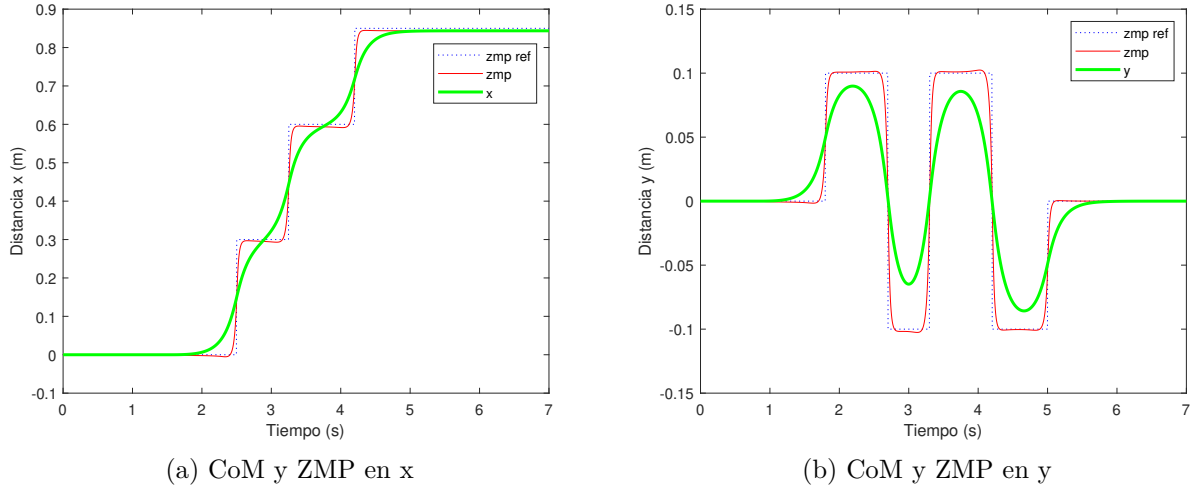


Figura 11: Trayectoria del CoM obtenida por preview control

## 5. Anexos

El código utilizado para generar las gráficas de la cinemática directa, el control cinemático y la generación del ZMP se puede encontrar en el siguiente repositorio: [https://github.com/DPR00/r1e\\_127](https://github.com/DPR00/r1e_127).

---

## Referencias

- [1] P. Biswal and P. K. Mohanty, “Development of quadruped walking robots: A review,” *Ain Shams Engineering Journal*, vol. 12, pp. 2017–2031, jun 2021.
- [2] C. Semini and P.-B. Wieber, *Legged Robots*, p. 1–8. Springer Berlin Heidelberg, 2020.
- [3] F. Grimminger, A. Meduri, M. Khadiv, J. Viereck, M. Wüthrich, M. Naveau, V. Berenz, S. Heim, F. Widmaier, J. Fiene, A. Badri-Spröwitz, and L. Righetti, “An open torque-controlled modular robot architecture for legged locomotion research,” *CoRR*, vol. abs/1910.00093, 2019.
- [4] D. J. Hyun, S. Seok, J. Lee, and S. Kim, “High speed trot-running: Implementation of a hierarchical controller using proprioceptive impedance control on the mit cheetah,” *The International Journal of Robotics Research*, vol. 33, pp. 1417–1445, ago 2014.
- [5] R. S. Lab, “Anymal.” Disponible en: <https://rsl.ethz.ch/robots-media/anymal.html>.
- [6] C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella, and D. G. Caldwell, “Design of hyq – a hydraulically and electrically actuated quadruped robot,” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 225, pp. 831–849, ago 2011.
- [7] BostonDynamics, “Spot: Automate sensing and inspection, capture limitless data, and explore without boundaries.” Disponible en: <https://www.bostondynamics.com/spot>.
- [8] O. E. Ramos, “Step-by-step kinematic modeling and control of a robot with a free-floating base,” in *2018 IEEE ANDESCON*, IEEE, ago 2018.
- [9] P.-B. Wieber, *Holonomy and Nonholonomy in the Dynamics of Articulated Motion*, pp. 411–425. Springer Berlin Heidelberg, 2006.
- [10] V. Scheinman, J. M. McCarthy, and J.-B. Song, *Mechanism and Actuation*, pp. 67–90. Springer International Publishing, 2016.
- [11] K. J. Waldron and J. Schmiedeler, *Kinematics*, pp. 11–36. Springer International Publishing, 2016.
- [12] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer London, 2009.
- [13] J. Denavit and R. S. Hartenberg, “A kinematic notation for lower-pair mechanisms based on matrices,” *Journal of Applied Mechanics*, vol. 22, pp. 215–221, jun 1955.
- [14] J. Chou, “Quaternion kinematic and dynamic differential equations,” *IEEE Transactions on Robotics and Automation*, vol. 8, no. 1, p. 53–64, 1992.

- 
- [15] K. Bouyarmane and A. Kheddar, “On the dynamics modeling of free-floating-base articulated mechanisms and applications to humanoid whole-body dynamics and control,” in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, IEEE, Nov 2012.
  - [16] C. Samson, E. Bernard, and M. L. Borgne, *Robot control: the task function approach*. Oxford University Press, 1991.
  - [17] O. E. Ramos, “A kinematic whole-body control system for highly redundant robots,” in *2016 IEEE ANDESCON*, IEEE, Oct 2016.
  - [18] L. Saab, O. E. Ramos, F. Keith, N. Mansard, P. Souères, and J. Fourquet, “Dynamic whole-body motion generation under rigid contacts and other unilateral constraints,” *IEEE Transactions on Robotics*, vol. 29, pp. 346–362, April 2013.
  - [19] N. Ratliff, “Controlling floating-based robots,” 2014.
  - [20] O. E. Ramos, *Generation of the whole-body motion for humanoid robots with the complete dynamics*. Universite Toulouse III Paul Sabatier, 2014.
  - [21] S. Kajita and B. Espiau, *Legged Robots*, p. 361–389. Springer Berlin Heidelberg, 2008.
  - [22] Y. H. Lee, Y. H. Lee, H. Lee, H. Kang, J. H. Lee, L. T. Phan, S. Jin, Y. B. Kim, D.-Y. Seok, and S. Y. e. a. Lee, “Development of a quadruped robot system with torque-controllable modular actuator unit,” *IEEE Transactions on Industrial Electronics*, vol. 68, no. 8, pp. 7263–7273, 2021.
  - [23] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, “Biped walking pattern generation by using preview control of zero-moment point,” *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, 2003.
  - [24] S. Kajita, H. Hirukawa, K. Harada, and K. Yokoi, *Introduction to Humanoid Robotics*. Springer Berlin Heidelberg, 2014.
  - [25] Y. H. Lee, J. C. Koo, H. R. Choi, Y. H. Lee, H. Lee, H. Kang, Y. B. Kim, J. H. Lee, L. T. Phan, S. Jin, and et al., “Whole-body motion and landing force control for quadrupedal stair climbing,” *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov 2019.