

Modelamiento de legged robots con ROS

Seykarin Mestre Muelas

February 24, 2025

1 Introducción

El modelado de robots de patas (legged robots) es un área de investigación clave en la robótica, ya que estos sistemas ofrecen ventajas significativas en terrenos irregulares y entornos complejos donde las ruedas pueden no ser eficientes. Sin embargo, el entendimiento de estos modelos puede ser desafiante para personas con bajos niveles de conocimiento en el área.

Este proyecto tiene como objetivo explicar de manera clara y accesible el modelado utilizado en la tesis doctoral *Optimization-based motion planning for legged robots*[1]. A través de este documento, se busca desglosar los conceptos fundamentales y las herramientas empleadas en el modelamiento de un legged robot para que cualquier persona interesada pueda comprender y aplicar el modelo.

2 Modelamiento matemático

El modelamiento matemático en robótica de locomoción con patas es esencial para describir y predecir el comportamiento dinámico del sistema. Existen tres modelos fundamentales que sirven como base para la formulación y control del movimiento de estos robots:

- **Dinámica de Cuerpos Rígidos (RBD, Rigid Body Dynamics):** Es el modelo más detallado, donde se consideran todos los grados de libertad y fuerzas actuantes en cada eslabón del robot. Permite una representación precisa pero a costa de una alta complejidad computacional.
- **Dinámica de Cuerpos Rígidos Simplificada (SRBD, Simplified Rigid Body Dynamics):** Reduce la complejidad del RBD al modelar el robot como un solo cuerpo rígido, generalmente representando el torso, y omitiendo dinámicas internas de las patas. Este modelo es útil para diseñar estrategias de control más eficientes.
- **Modelo del Péndulo Invertido Lineal (LIPM, Linear Inverted Pendulum Model):** Simplifica aún más la representación al asumir que la masa del robot está concentrada en un punto y que el movimiento se asemeja al de un péndulo invertido. Se usa ampliamente en el control de equilibrio y planificación de trayectorias debido a su formulación lineal.

Cada uno de estos modelos tiene aplicaciones específicas dependiendo del nivel de precisión requerido y de las restricciones computacionales. En esta sección, se explorarán sus fundamentos y cómo se utilizan en el desarrollo de estrategias de locomoción para robots con patas.

2.1 Dinámica de un cuerpo rígido (RBD)

En el modelo del cuerpo rígido se asume que:

A1: (los cuerpos no se deforman por las fuerzas.)

Con la suposición anterior, el sistema puede describirse con sus coordenadas generalizadas. Los estados del sistema quedan dependientes de la posición del tronco (q_b) y de los ángulos de cada una de las articulaciones (q_j).

$$q = [q_b^T \ q_j^T]^T \quad (1)$$

2.1.1 Ejemplo

Supongamos que estamos modelando un Legged Robot que tiene las siguientes características:

$$q = [q_b^T \ q_j^T]^T$$

Donde:

$$q_b^T = [x \ y \ z \ \phi \ \theta \ \psi]$$

$$q_j^T = [q_1^T \ q_2^T \ q_3^T \ q_4^T]$$

Siendo q_1^T el vector correspondiente a las articulaciones de la pata 1:

$$q_1^T = [\theta_{1,1} \ \theta_{1,2} \ \theta_{1,3}]$$

$$q_1^T = [0.2 \ -0.1 \ 0.5]$$

Donde:

- q_b^T representa el estado del tronco del robot:
 - x : posición en el eje x del centro de masa del tronco (en metros).
 - y : posición en el eje y del centro de masa del tronco (en metros).
 - z : posición en el eje z del centro de masa del tronco (en metros).
 - ϕ : orientación en roll del tronco (en radianes).
 - θ : orientación en pitch del tronco (en radianes).
 - ψ : orientación en yaw del tronco (en radianes).
- q_1^T representa los ángulos de las articulaciones de la pata 1:
 - $\theta_{1,1} = 0.2$: ángulo de la primera articulación de la pata 1 (en radianes).
 - $\theta_{1,2} = -0.1$: ángulo de la segunda articulación de la pata 1 (en radianes).

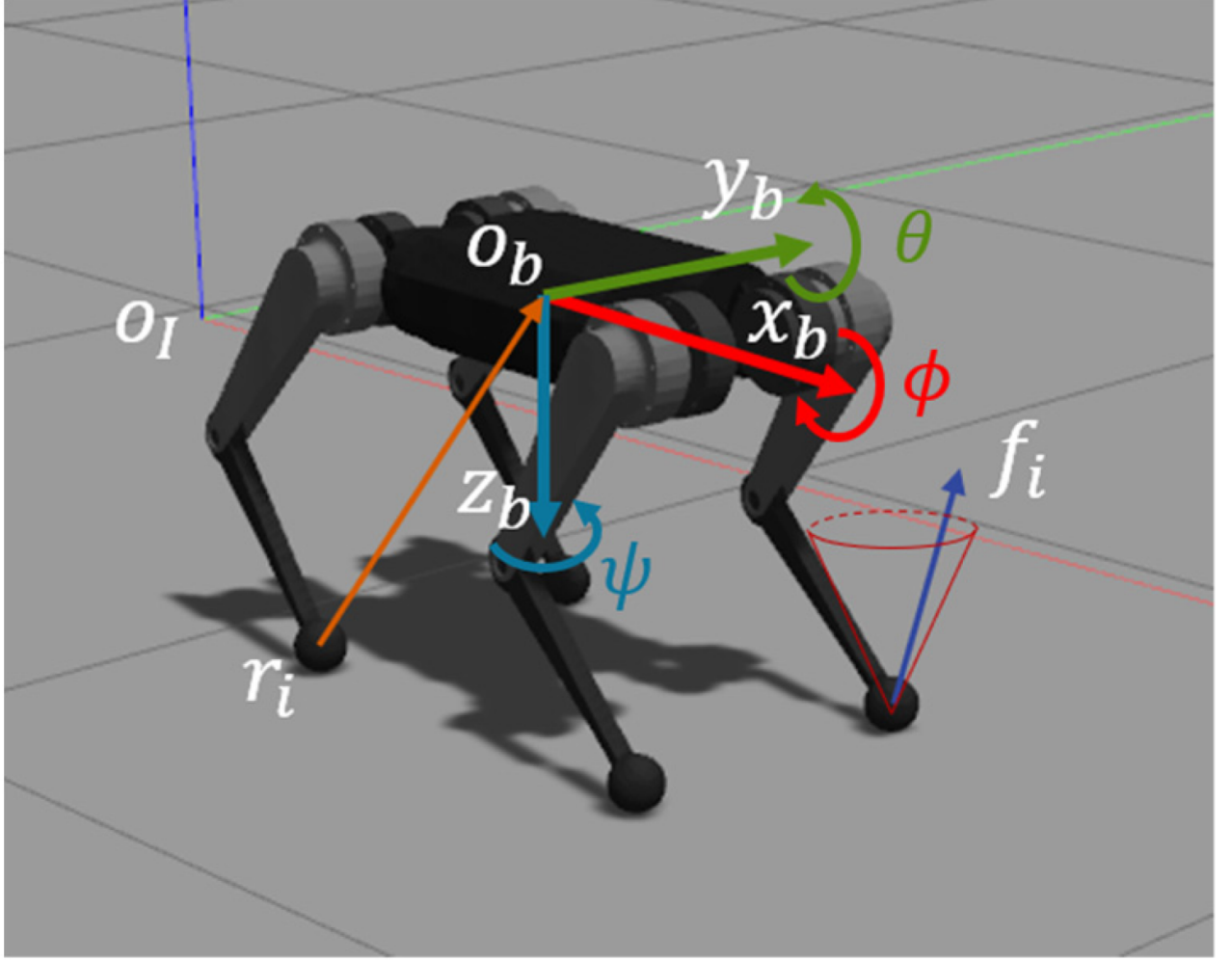


Figure 1: Ejes de movimiento de un legged robot.

- $\theta_{1,3} = 0.5$: ángulo de la tercera articulación de la pata 1 (en radianes).

Por la segunda ley de Newton, se establece una ecuación matricial de los torques y fuerzas externas que actúan sobre el sistema:

$$M(q)\ddot{q} + h(q, \dot{q}) = S^T \tau + J(q)^T f \quad (2)$$

Donde:

- $M \in \mathbb{R}^{(6+n) \times (6+n)}$ es la matriz de inercia generalizada, que describe cómo las masas y sus distribuciones afectan la aceleración \ddot{q} .
- $h \in \mathbb{R}^{6+n}$ representa las fuerzas no lineales, como:
 - **Fuerza centrífuga**: generada por el movimiento giratorio de las articulaciones.
 - **Fuerza de Coriolis**: describe cómo el movimiento de una parte del robot afecta dinámicamente a otras.
 - **Gravedad**: que actúa hacia el En suelo.

- $S = \begin{bmatrix} 0_{n \times 6} \\ I_{n \times n} \end{bmatrix}$ es una matriz de selección que aplica los torques de articulaciones τ únicamente a las coordenadas actuadas. En este caso las primeras n filas representan las cooredandas del tronco, pero este no se ve afectado por ningun torque directo.
- $J(q)$ es el jacobiano de las coordenadas generalizadas, que proyecta las fuerzas externas f en coordenadas específicas del sistema.

La ecuación 2 puede dividirse en dos:

1. El modelo del **tronco**:

$$M_u(q)\ddot{q} + h_u(q, \dot{q}) = J_u(q)^T f \quad (3)$$

Aquí, todos los términos llevan el subíndice u , indicando que son aplicables únicamente al tronco. En esta ecuación no aparece el término de torques (τ), ya que no se aplican directamente torques sobre el tronco, sino que este es controlado a través de las fuerzas externas generadas por las patas.

2. El modelo de las **articulaciones** :

$$M_a(q)\ddot{q} + h_a(q, \dot{q}) = \tau + J_a(q)^T f \quad (4)$$

Esta ecuación incluye el término τ , que describe los torques aplicados directamente sobre las articulaciones mediante actuadores, tales como motores u otros dispositivos internos del sistema. La matriz de posiciones S desaparece, ya que solamente era necesaria para clasificar los lugares en los que se ejercen los torques, ya que estos no se aplican en el tronco.

Dado que existen motores generadores de torques en las articulaciones, siempre y cuando mantengan el límite físico del hardware, esta ecuacion (4) está determinada y se vuelve trivial, por lo que las diferentes ecuaciones que produce son fácilmente solubles y se puede predecir el comportamiento del sistema de manera unica y precisa.

2.1.2 Dinámica centroidal

El tronco del robot se modela utilizando el cambio del momentum lineal y angular debido a las fuerzas externas aplicadas sobre el cuerpo y proyectadas en el centro de masa.

$$\frac{d}{dt}A\dot{q} = \begin{bmatrix} mg + \sum_{i=1}^{n_i} f_i \\ \sum_{i=1}^{n_i} f_i \times (r(q) - p(q)) \end{bmatrix} \quad (5)$$

$$A(q)\dot{q} + \dot{A}(q)\dot{q} = \begin{bmatrix} mg + \sum_{i=1}^{n_i} f_i \\ \sum_{i=1}^{n_i} f_i \times (r(q) - p(q)) \end{bmatrix} \quad (6)$$

Donde:

- $A(q) \in \mathbb{R}^{6 \times (6+n)}$: Matriz de Momento Centroidal (CMM), que mapea las velocidades generalizadas del robot a su momentum.

- mg : Fuerza gravitacional, donde m es la masa total del robot y g la aceleración gravitacional.
- \mathbf{f}_i : Fuerzas externas aplicadas en los puntos de contacto.
- $\mathbf{r}(q)$: Posición del CoM en las coordenadas generalizadas.
- $\mathbf{p}_i(q)$: Posiciones de los puntos de contacto de las fuerzas externas con el robot.

La ecuación de la izquierda representa el cambio de momentum. La ecuación superior del lado derecho describe las fuerzas lineales que intervienen en el momentum lineal, y la inferior describe los torques externos que intervienen en el momentum angular. Al restar estos vectores y multiplicarlos por las fuerzas externas mediante el producto cruz, se obtienen los torques generados.

2.1.3 Ejemplo Práctico

Imaginemos un robot cuadrúpedo con las siguientes características:

- Masa total del robot: $m = 50$ kg.
- Aceleración gravitacional: $g = 9.81$ m/s².
- El CoM se encuentra en $\mathbf{r}(q) = [0.0, 0.0, 0.5]$ m.
- Cada pata genera una fuerza de contacto \mathbf{f}_i y se encuentran en:
 - Pata 1: $\mathbf{p}_1 = [-0.2, 0.1, 0.0]$ m
 - Pata 2: $\mathbf{p}_2 = [0.2, 0.1, 0.0]$ m
 - Pata 3: $\mathbf{p}_3 = [-0.2, -0.1, 0.0]$ m
 - Pata 4: $\mathbf{p}_4 = [0.2, -0.1, 0.0]$ m

Supongamos que las fuerzas normales son:

$$\mathbf{f}_1 = \mathbf{f}_2 = \mathbf{f}_3 = \mathbf{f}_4 = [0, 0, 122.625] \text{ N}$$

Esto compensa el peso del robot ($mg = 50 \times 9.81 = 490.5$ N).

El momento angular de la pata 1 se calcula como:

$$\mathbf{f}_1 \times (\mathbf{r} - \mathbf{p}_1) = [0, 0, 122.625] \times ([0.0, 0.0, 0.5] - [-0.2, 0.1, 0.0])$$

Resolviendo:

$$\mathbf{f}_1 \times (\mathbf{r} - \mathbf{p}_1) = [12.2625, 24.525, 0] \text{ N} \cdot \text{m}$$

Realizando cálculos similares para las otras patas, obtenemos el momento total. Estos resultados se sustituyen en la ecuación dinámica para verificar que el modelo sea físicamente correcto.

3 Modelo de un cuerpo rígido dinamico único (SRBD)

Con el fin de simplificar las ecuaciones se considera que el robot completo puede aproximarse como un único cuerpo rígido concentrado en su Centro de Masa (CoM), así eliminando la dependencia de los angulos de las articulaciones en las ecuaciones. Por lo tanto, las siguientes dos supociciones son necesarias:

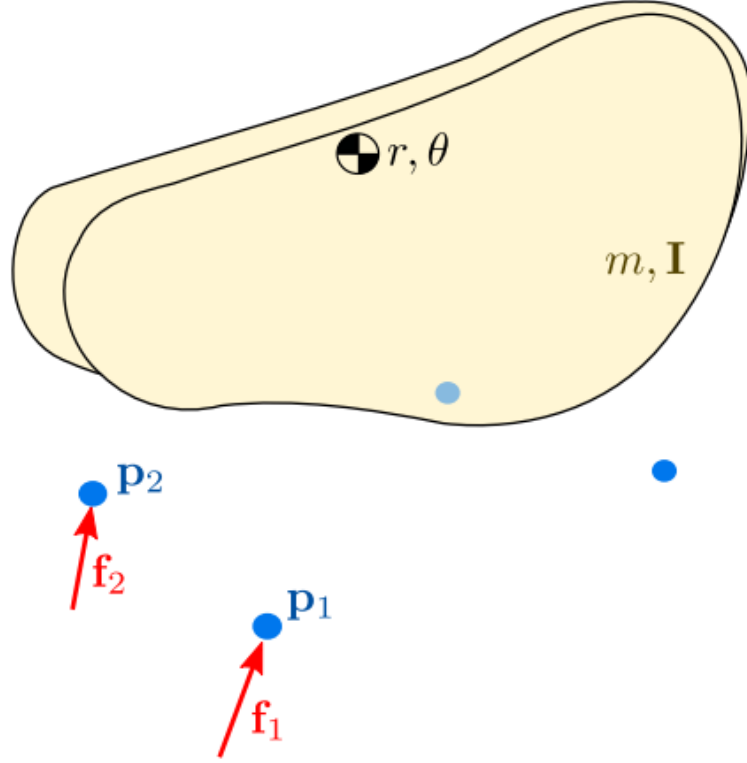


Figure 2: Representacion del modelo SRBD.

A2: El momentum producido por las velocidades articulares es despreciable.

A3 El momentum total permanece constante aunque las articulaciones se muevan.

Así se logra separar la ecuacion, obteniendo:

$$m\ddot{r} = mg + \sum_{i=1}^{n_i} f_i \quad (7)$$

$$I(\theta)\dot{w} + w \times I(\theta)w = \sum_{i=1}^{n_i} f_i \times (r - p_i) \quad (8)$$

Donde:

- $M(q)$: Matriz de inercia generalizada del sistema.
- r : Posicion del centro de masa (CoM)
- w : Velocidad angular total del robot.
- m : Masa total del robot.
- I : Inercia del tronco dependiente de su orientación.
- θ : Es la orientacion del tronco en los diferentes ejes coordenados.

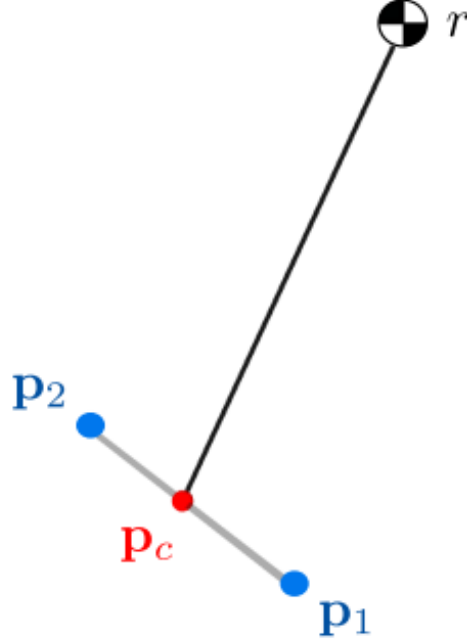


Figure 3: Representacion del LIPM.

4 Modelo de Péndulo Invertido Lineal (LIPM)

El Modelo de Péndulo Invertido Lineal (LIPM) es utilizado con el proposito eliminar no linealidad en las ecuaciones producida por el producto cruz. Se realizan 3 suposiciones adicionales:

- A4:** El centro de masa (r_z) no varía en el eje Z.
- A5;** La velocidad w y aceleración \dot{w} angular del tronco son iguales a cero.
- A6:** Los puntos de apoyo (base de las patas) permanecen a una altura constante (p_z)

La ecuación del movimiento es:

$$m\ddot{r}_x = \sum_{i=1}^{n_i} f_{i,x} = \frac{mg}{r_z - p_z} \left(r_x - \frac{\sum_{i=1}^{n_i} f_{i,z} p_{i,x}}{\sum_{i=1}^{n_i} f_{i,z}} \right) = \frac{mg}{h} (r_x - p_{c,x}) \quad (9)$$

$$\ddot{r}_x = \frac{g}{h} (r_x - p_{c,x})$$

Donde:

- $r_z - p_z = h$: Al restar el punto del centro de masa, en el eje z , con el punto de presión en las extremidades, se obtiene la altura del centro de masa.
- r_x : Posición del centro de masa (CoM) en el eje horizontal.
- h : Altura constante del centro de masa (CoM) sobre el suelo.
- $p_{c,x}$: Centro de Presión (CoP) representa el punto (en el eje x) dentro del área de contacto del pie del robot donde la fuerza resultante de reacción del suelo actúa para mantener la estabilidad.
- g : Es el valor de la gravedad.

Este modelo presenta desventajas gracias a las suposiciones que hace, ya que puede perder información importante sobre las fuerzas en el eje vertical que afectan a los pies.

4.1 Ejemplo

Supongamos que un robot está caminando y su centro de masa (CoM) tiene la siguiente posición inicial y condiciones:

- $r_x(0) = 0.2 \text{ m}$ (posición inicial del CoM en el eje x).
- $\dot{r}_x(0) = 0.1 \text{ m/s}$ (velocidad inicial del CoM).
- La altura del CoM sobre el suelo es constante: $h = r_z - p_z = 1.0 \text{ m}$.
- La gravedad es $g = 9.81 \text{ m/s}^2$.
- El Centro de Presión (CoP) en el eje x es constante: $p_{c,x} = 0.0 \text{ m}$.

La ecuación diferencial que describe el movimiento es:

$$\ddot{r}_x = \frac{g}{h}(r_x - p_{c,x}).$$

Con los datos:

$$\ddot{r}_x = \frac{9.81}{1.0}(r_x - 0.0) = 9.81r_x.$$

Esta ecuación diferencial se puede resolver suponiendo una solución de la forma:

$$r_x(t) = C_1 e^{\omega t} + C_2 e^{-\omega t},$$

$$\text{donde } \omega = \sqrt{\frac{g}{h}} = \sqrt{\frac{9.81}{1.0}} = 3.13$$

Las condiciones iniciales son:

- $r_x(0) = 0.2$,
- $\dot{r}_x(0) = 0.1$.

Derivando $r_x(t)$:

$$\dot{r}_x(t) = 3.13C_1 e^{3.13t} - 3.13C_2 e^{-3.13t}.$$

En $t = 0$:

$$\begin{aligned} r_x(0) &= C_1 + C_2 = 0.2, \\ \dot{r}_x(0) &= 3.13C_1 - 3.13C_2 = 0.1. \end{aligned}$$

Resolviendo este sistema:

$$\begin{aligned} C_1 + C_2 &= 0.2, \\ C_1 - C_2 &= \frac{0.1}{3.13} \approx 0.0319. \end{aligned}$$

Sumando y restando las ecuaciones:

$$\begin{aligned} 2C_1 &= 0.2 + 0.0319 = 0.2319 \implies C_1 = 0.11595, \\ 2C_2 &= 0.2 - 0.0319 = 0.1681 \implies C_2 = 0.08405. \end{aligned}$$

La solución general es:

$$r_x(t) = 0.11595e^{3.13t} + 0.08405e^{-3.13t}.$$

La solución muestra un término exponencial creciente ($e^{3.13t}$) y otro decreciente ($e^{-3.13t}$). Sin un control externo, el CoM diverge rápidamente debido al término $e^{3.13t}$. Esto es una representación matemática de la inestabilidad inherente del sistema, lo que resalta la necesidad de un mecanismo de control activo en los Legged Robots para mantener el equilibrio.

5 Tabla de modelos

Modelo	Estados x	Entrada u	Suposiciones
Cuerpo rigido dinamico (RBD)	q_b, q_j	τ, f	A1
Dinamica centroidal (CD)	q_b, q_j	f	A1
Cuerpo rigido dinamico unico (SRBD)	Γ, θ, p_i	f	A1,A2,A3
Pendulo invertido lineal (LIPM)	r_x, r_y	p_c	A1,A2,A3,A4-A6

Table 1: Descripción de los modelos dinámicos.

6 Restricciones físicas de la locomoción con patas

Existen fuertes restricciones sobre el movimiento de los legged robot y esto es lo que hace que la locomoción con patas sea tan difícil desde el punto de vista de la planificación y el control.

- A diferencia de otros sistemas como drones, los legged robots solo pueden aplicar fuerzas estando en contacto con el suelo. Sistemas como los automoviles tienen contacto constante con el suelo y las fuerzas verticales no son tema de interés al encontrarse en un equilibrio constante, características que los legged robots no poseen y complican el control de todos los procesos.

- Al desplazar una extremidad, esta se ve forzada a levantarse y pierde su capacidad de producir una fuerza que le permita controlar la posición del tronco.
- Las fuerzas tangenciales deben estar dentro de un cono de fricción. Para entender este punto, es necesario recordar que la fricción es directamente proporcional a la fuerza normal (opuesta a la fuerza contra la zona de contacto) y a un coeficiente de fricción propio de cada superficie.

$$friccion = -f_y * n \quad (11)$$

Siendo n el coeficiente de fricción asociado a la superficie. Por lo que la fuerza originadas al sumar la fuerza en el eje Y y en el eje X no debe ser mayor a la fuerza de fricción, esto para evitar que el robot resbale.

$$\sqrt{f_x^2 + f_y^2} \leq n * f_z \quad (12)$$

7 Planificación tradicional de la locomoción con patas

El método tradicional divide en dos etapas:

7.1 Selección de puntos de apoyo

Se eligen los lugares donde pisará el robot evitando obstáculos y se buscando superficies planas. Una vez conocidas donde y cuando las fuerzas externas pueden actuar, se puede establecer mecanismos para crear un movimiento corporal que pueda crear fuerzas que lleven al objetivo esperado.

7.2 Cálculo del movimiento corporal

Se usa un modelo como LIPM para generar el movimiento del cuerpo.

7.2.1 Limitaciones del LIPM

- Altura constante del centro de masa: El modelo asume que el centro de masa (CoM) se mantiene a una altura constante, lo que impide simular cambios verticales, saltos, terrenos irregulares o posición de las patas en diferentes a diferentes niveles, por lo que no se maneja en superficies inclinadas.
- Velocidad y aceleración angular nulas: Se supone que la base no gira ni acelera angularmente, lo que limita la capacidad del robot para reorientar su cuerpo durante movimientos dinámicos.
- Simplificación de las fuerzas de contacto: El LIPM agrupa las interacciones en un único punto, dejando de lado la complejidad de las fuerzas tridimensionales y las restricciones reales del contacto, como el cono de fricción.

8 Publicaciones relevantes

Cada artículo habla sobre un aspecto diferente de la planificación de movimientos para los Legged robots, y juntos forman enfoque más completo para la generación de trayectorias óptimas en terrenos complejos

8.1 *Simultaneous foothold and body optimization*

8.1.1 Motivación

En la planificación tradicional de movimiento, el proceso se divide en dos etapas. En la primera se determinan los puntos de apoyo (dónde pisar) mediante heurísticas (por ejemplo, límites en la longitud del paso) y luego se planifica el movimiento del cuerpo. Esta separación reduce la complejidad, pero obliga a usar reglas que pueden ser demasiado limitantes para rango de movimientos posibles.

El objetivo del paper es reducir la dependencia de estas heurísticas optimizando de forma conjunta tanto los movimientos corporales como la ubicación de los pies.

8.1.2 Contribución

1. Se integra la planificación del movimiento del cuerpo y la determinación de los puntos de apoyo en una única formulación de optimización.
 - ¿Qué significa?: Se trata de unificar en un solo proceso la generación de la trayectoria del cuerpo y la ubicación de los puntos de apoyo, en lugar de abordarlos por separado.
 - ¿Por qué es útil?: Permite encontrar soluciones de movimiento que tengan en cuenta la relación entre el movimiento corporal y el posicionamiento de los pies, lo que puede generar trayectorias más naturales y eficientes.
2. Aunque la inclusión simultánea de ambas variables hace que el problema sea no lineal, se demuestra que su solución puede tomar solamente mili segundos.

8.2 *Vertex-based ZMP constraints*

8.2.1 Motivación

Punto de momento cero (ZMP): es el punto donde la suma de la totalidad de las fuerzas horizontales, contabilizando la inercia y la gravedad sea igual a cero. El concepto asume que el área de contacto con el suelo es plana y que ejerce suficiente fricción para evitar que los pies se deslicen.

Dado el paper anterior, al optimizar simultáneamente el cuerpo y los puntos de apoyo, se ve limitado a modos de locomoción en los que al menos tres patas deben estar en contacto con el suelo. Sin embargo, en muchos trotes biológicos y en modos de locomoción más dinámicos, es común que sólo una o dos patas mantengan contacto en ciertos instantes.

Los métodos tradicionales basados en el LIPM y en el concepto de ZMP generalmente abstraen la fuerza de contacto en un solo punto, lo que dificulta modelar contactos puntuales o lineales.

8.2.2 Contribución

1. Se redefine la restricción del ZMP (Zero Moment Point) de modo que centro de presión (CoP) se trate directamente como una entrada del sistema. Esto permite modificar dinámicamente las áreas de soporte a través de la optimización de los puntos de apoyo.
 - ¿Qué significa?: En vez de utilizar el ZMP como un punto fijo donde se supone que se concentra la fuerza para mantener el equilibrio, se considera al centro de presión (CoP) como variable de control. Esto posibilita ajustar de forma directa la ubicación y la forma del área de soporte, en función de cómo se distribuyen las fuerzas en cada pie.
 - ¿Por qué es útil?: Permite que el robot se adapte a cambios en el terreno y a movimientos dinámicos, ya que la optimización ajusta continuamente el área de soporte.
 2. Se introduce un método que modela los contactos no solo como áreas (polígonos) sino que también abarca puntos y líneas de contacto mediante una representación basada en vértices.
 - ¿Qué significa?: En lugar de ordenar los puntos de contacto para formar un polígono de soporte, se identifican los vértices clave que definen cualquier tipo de contacto, ya sea puntual, lineal o en área. Esta representación abarca distintos tipos de contacto:
 - Puntos: Contactos puntuales donde el robot toca el suelo en ubicaciones específicas.
 - Líneas: Contactos que se extienden a lo largo de una dirección, como si el pie estuviera apoyado de forma alargada.
 - Áreas: Contactos extendidos que cubren una región más amplia del suelo.
 - ¿Por qué es útil?: Permite modelar una variedad más amplia de situaciones de contacto, incluyendo aquellas en las que el robot hace contacto con el suelo en formas no convencionales (por ejemplo, apoyos en líneas o puntos aislados). Al eliminar la necesidad de ordenar y reconstruir el polígono de soporte, se reduce la complejidad computacional en la formulación de restricciones.
 3. Gracias a esto, se pueden generar de forma eficiente movimientos como caminar, trotar, saltar, caminar lentamente, transiciones entre marchas, incluso recuperaciones ante empujes, e incluso modos bipedales y modos de cojear, todo en un mismo algoritmo.
- Ejemplo:** Imagina un robot cuadrúpedo que camina sobre un terreno irregular.
- Contacto puntual: En un paso, uno de sus pies toca el suelo en un punto muy pequeño, como si se apoyara en una piedra redonda. Con la representación basada en vértices, el algoritmo identifica ese único punto de contacto como un vértice.

- Contacto en línea: En otro paso, el mismo pie se desliza ligeramente a lo largo de una hendidura en el terreno, generando un contacto que se extiende en forma de línea. Aquí, el sistema extrae dos vértices (los extremos de la línea) que definen esta forma de contacto.
- Contacto de área: Finalmente, en una fase de apoyo prolongada, el pie se asienta completamente sobre una superficie plana, generando un contacto extendido. El algoritmo detecta varios vértices que delinean el contorno de esta área.

8.3 *Gait and trajectory optimization*

8.3.1 Motivación

Aunque los enfoques anteriores permiten una optimización conjunta, ambos se basan en el modelo LIPM. Este enfoque resulta insuficiente para movimientos complejos en los que se requieren cambios en la altura del centro de masa, reorientaciones corporales o movimientos que involucren fases de vuelo. La abstracción de las fuerzas de contacto en un único punto limita la capacidad de modelar adecuadamente la interacción real entre el robot y el terreno, especialmente en escenarios con terrenos irregulares.

8.3.2 Contribución

1. Se reemplaza el modelo LIPM (que asume una altura constante del centro de masa) por el modelo de un solo cuerpo rígido (SRBD) en 6 dimensiones. Esto significa que el robot se modela como una única pieza que se mueve y rota en el espacio, considerando tres dimensiones para la posición (x , y , z) y tres para la orientación (roll, pitch, yaw).
 - ¿Qué significa?: En lugar de limitarse a movimientos en un plano horizontal (como en el LIPM), se tiene en cuenta la totalidad de los movimientos posibles, incluyendo cambios de altura y rotaciones.
 - ¿Por qué es útil?: Permite generar trayectorias que reflejen de manera más fiel el comportamiento real del robot, ya que se consideran todos los grados de libertad, facilitando movimientos dinámicos como saltos, inclinaciones y cambios de postura, esenciales para operar en terrenos irregulares o en situaciones de alta dinámica. Gracias al mejor modelamiento, su control también puede ser más preciso.
2. En lugar de resumir toda la interacción entre el robot y el suelo en un único punto (el ZMP), se evalúan las fuerzas que actúan en cada pie de manera completa, considerando sus componentes verticales y laterales.
 - ¿Qué significa esto?: Cada pie tiene fuerzas en tres direcciones (hacia arriba/abajo, adelante/atrás y a los lados).
 - ¿Por qué es útil?: Permite aplicar directamente las leyes físicas (como la fricción y la unilateralidad, es decir, que los pies solo pueden empujar y no tirar) para asegurarse de que cada contacto se mantenga estable.

3. Parametrización basada en fases: Se divide el movimiento de cada pierna en diferentes etapas o fases (por ejemplo, levantar, mover en el aire y aterrizar).
 - ¿Qué significa esto?: En lugar de tener valores fijos para cuándo y cómo se mueve cada pie, el algoritmo determina de manera continua y automática estas fases.
 - ¿Por qué es útil?: Permite que el robot decida de forma flexible el orden, la duración y la secuencia de los pasos, adaptándose mejor a distintos terrenos y situaciones.
4. En vez de combinar variables continuas (como posiciones y velocidades) con variables discretas (como decisiones binarias sobre qué paso dar), se trabaja únicamente con variables continuas.
 - ¿Qué significa esto?: Todo el problema se plantea en un espacio donde las decisiones pueden cambiar de forma gradual, sin tener que forzar elecciones de “sí/no”.
 - ¿Por qué es útil?: Facilita la resolución del problema usando métodos de optimización no lineal, lo que suele ser más rápido y menos complejo que resolver problemas que mezclan variables continuas y discretas.

9 Métricas de capacidad de los algoritmos de planificación de movimiento

Esta sección establece un conjunto de métricas que permiten evaluar y comparar el rendimiento de diferentes algoritmos de planificación de movimiento

9.1 Componentes optimizados

Se analiza qué elementos del problema de locomoción se optimizan, tales como el movimiento del cuerpo, la ubicación de los puntos de apoyo y la secuencia de pasos.

9.2 Dificultad de la tarea y del terreno

Se evalúa la complejidad de las tareas que se han planteado (por ejemplo, superar terrenos irregulares o realizar maniobras dinámicas) y la capacidad del algoritmo para encontrar soluciones en estos escenarios.

9.3 Confiabilidad de las soluciones

Esta métrica clasifica la capacidad del algoritmo para generar, de forma consistente y sin intervención manual excesiva, soluciones viables ante diferentes condiciones. Se pueden definir niveles de confiabilidad como bajo, medio o alto, dependiendo de la robustez del algoritmo frente a variaciones en el objetivo, el terreno o las condiciones iniciales.

	Paper I	Paper 2	Paper 3
Modelo dinámico	LIPM (4)	LIPM (4)	SRBD (7) y (8)
Componentes optimizados			
Movimiento base	2D	2D	6D
Posiciones de apoyo	2D	2D	3D
Secuencia de pasos	×	×	✓
Tiempo de paso	×	×	✓
Fuerza de contacto	×	×	✓
Movimiento de la pierna oscilante	×	×	✓
Número de pasos	×	×	×
Dificultad de la tarea mostrada			
Contactos puntuales y lineales	×	✓	✓
Terreno no plano	×	×	✓
Terreno inclinado	×	×	✓
Fases de vuelo	×	×	✓
Contactos deslizantes	×	×	×
Impulsos	×	×	×
Número de efectores finales	4	4	1, 2, 4
Fiabilidad	alta	alta	media
Tiempo de cálculo ¹	3 ms	35 ms	100 ms
Verificación ²	100 E	50 E, 0 S	10 E, 20 S

Table 2: Capacidades del algoritmo desarrollado de planificación de movimiento.

9.4 Tiempo de computo

Se mide cuánto tiempo tarda el algoritmo en encontrar una solución.

9.5 Verificación de los planes de movimiento

Se considera la manera en que se comprueba la validez de los planes generados, ya sea mediante inspección visual, simulación en entornos de física o pruebas en robots reales. Este aspecto es crucial para garantizar que las soluciones sean físicamente correctas y ejecutables en un sistema real.

9.6 Tabla comparativa

9.7 Lista de contribuciones

C1: Optimización simultánea de las posiciones de apoyo (footholds) y del movimiento del centro de masa (CoM)

Paper I: Se desarrolló un algoritmo que optimiza al mismo tiempo dónde colocar las patas del robot (footholds) y cómo mover el cuerpo (CoM) para lograr un movimiento estable. Esto elimina la necesidad de un planificador de pasos separado y permite movimientos más dinámicos y adaptativos.

C2: Reformulación del problema de locomoción basado en el Punto de Momento Cero (ZMP) en un problema de optimización de trayectoria (TO)

Paper II: Se reformuló el enfoque tradicional del ZMP (usado para mantener el equilibrio) en un problema de optimización de trayectoria utilizando como entrada el CoP. Esto permite comparar más fácilmente con otros métodos de optimización y genera movimientos más variados.

C3: Representación basada en vértices de las restricciones de área de apoyo

Paper II: Se introdujo una nueva forma de representar las áreas de apoyo (donde el robot puede apoyarse) usando vértices. Esto permite manejar contactos puntuales, lineales y de área de manera uniforme, lo que facilita la generación de movimientos como caminar, trotar y recuperarse de empujones.

C4: Generación automática de movimientos con fases de vuelo usando el modelo de Dinámica de Cuerpo Rígido (SRBD)

Paper III: Se reemplazó el modelo simplificado LIPM por el modelo SRBD, que permite generar movimientos con fases de vuelo (como saltos) y manejar terrenos no planos. Esto es esencial para movimientos más dinámicos y complejos.

C5: Descubrimiento automático de la secuencia de pasos (gait) mediante una parametrización basada en fases

Paper III: Se introdujo una nueva forma de parametrizar el movimiento de las patas y las fuerzas de contacto basada en fases (contacto y vuelo). Esto permite que el algoritmo optimice la secuencia de pasos y los tiempos de cada fase sin necesidad de programación entera mixta (MIP), lo que acelera el proceso.

C6: Algoritmo principal que genera planes de movimiento para robots con patas

Paper III: El resultado final es un algoritmo que determina automáticamente la secuencia de pasos, los tiempos de cada paso, las posiciones de apoyo, las fuerzas de contacto, los movimientos de las patas en el aire y el movimiento del cuerpo en 6 dimensiones. Esto permite generar movimientos altamente dinámicos sobre terrenos no planos para robots con diferentes morfologías (monopodos, bípedos, cuadrúpedos).

10 Simultaneous foothold and body optimization

Este trabajo contribuye a la planificación de movimientos dinámicamente factibles para robots con patas utilizando restricciones de factibilidad dinámica. Una de las claves en esta formulación es el uso del modelo Cart-Table, una variación simplificada del Linear Inverted Pendulum Model (LIPM) que permite describir de manera eficiente la locomoción estable de robots bípedos y cuadrúpedos.

- ZMP (Zero Moment Point): El **Zero Moment Point (ZMP)** es el punto en el cual la suma de los momentos (torques) generados por las fuerzas de contacto del robot con el suelo se anula. Para que el robot sea estable, el ZMP debe permanecer dentro del polígono de soporte, es decir, dentro de la región delimitada por los puntos de contacto de las patas con el suelo.

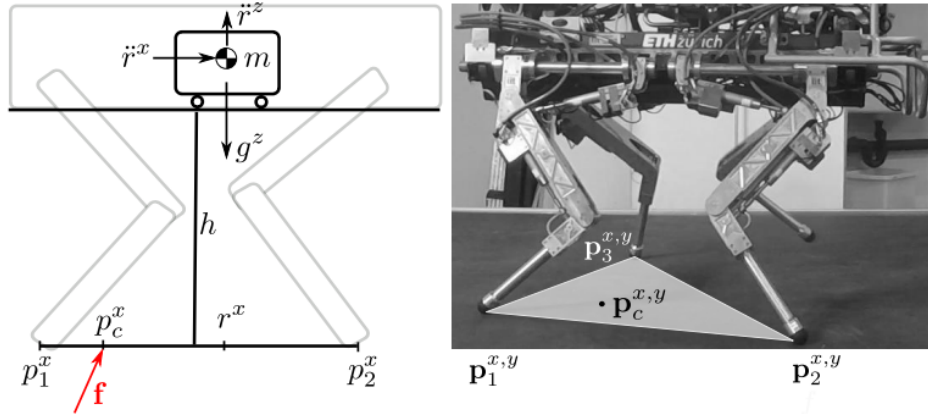


Figure 4: Modelo Cart-Table.

10.1 Modelo Cart-Table

El **modelo Cart-Table** es una simplificación del **péndulo invertido lineal (LIPM)** utilizado para modelar la locomoción de robots bípedos y cuadrúpedos. Representa la dinámica del **Centro de Masa (CoM)** asumiendo que está montado sobre una mesa ficticia (Table) con patas que actúan como un carrito (Cart), lo que permite que el CoM se mueva en el plano horizontal mientras permanece a una altura fija o variable.

10.1.1 Ecuación del Modelo Cart-Table

La ecuación fundamental del modelo Cart-Table es:

$$\ddot{r}_x = \frac{g_z + \ddot{r}_z}{r_z}(r_x - p_x) \quad (13)$$

$$\ddot{r}_y = \frac{g_z + \ddot{r}_z}{r_z}(r_y - p_y) \quad (14)$$

Donde:

- r_x, r_y, r_z representan las coordenadas del **Centro de Masa (CoM)**.
- p_x, p_y son las coordenadas del **Zero Moment Point (ZMP)**.
- g_z es la aceleración gravitacional en la dirección z .
- \ddot{r}_z es la aceleración vertical del CoM.

10.1.2 Ejemplo

Supongamos que un robot bípedo tiene las siguientes condiciones:

- **Centro de Masa (CoM):** $(r_x, r_y, r_z) = (0.5, 0.5, 1.0)$ m
- **Zero Moment Point (ZMP):** $(p_x, p_y) = (0.4, 0.5)$ m
- **Gravedad:** $g_z = 9.81$ m/s²

- **Aceleración vertical:** $\ddot{r}_z = 0.2 \text{ m/s}^2$

Calculamos la aceleración en x :

$$\ddot{r}_x = \frac{9.81 + 0.2}{1.0}(0.5 - 0.4) \quad (15)$$

$$\ddot{r}_x = 10.01 \times 0.1 = 1.001 \text{ m/s}^2 \quad (16)$$

Esto significa que el CoM tenderá a moverse hacia adelante con una aceleración de **1.001 m/s²** en la dirección de p_x , lo que estabiliza la marcha del robot.

10.2 Restricciones dinámicas

Garantizan que el movimiento generado sea físicamente estable según el modelo LIPM:

10.2.1 Base de Soporte

El polígono de soporte es la región dentro de la cual el **ZMP** debe permanecer para garantizar la estabilidad del robot. Se construye a partir de los puntos de contacto de las patas con el suelo.

Ejemplo: Si un robot cuadrúpedo tiene sus cuatro patas en contacto con el suelo en las posiciones:

$$P_1 = (0, 0), \quad P_2 = (1, 0), \quad P_3 = (1, 1), \quad P_4 = (0, 1) \quad (17)$$

El polígono de soporte es un cuadrado delimitado por estos cuatro puntos, y el ZMP debe permanecer dentro de esta área para evitar la inestabilidad.

10.2.2 Cálculo de Bordes del Polígono de Soporte

Cada borde del polígono se modela con una ecuación de línea basada en dos puntos consecutivos del polígono.

Fórmula: Para un borde definido por los puntos $P_1 = (p_{x1}, p_{y1})$ y $P_2 = (p_{x2}, p_{y2})$, la ecuación del borde es:

$$n_e = \frac{1}{d} \begin{bmatrix} p_{y1} - p_{y2} \\ p_{x2} - p_{x1} \\ p_{x1}p_{y2} - p_{x2}p_{y1} \end{bmatrix} \quad (18)$$

Donde d es la distancia euclidiana entre los dos puntos:

$$d = \sqrt{(p_{x2} - p_{x1})^2 + (p_{y2} - p_{y1})^2} \quad (19)$$

10.2.3 Restricción del ZMP

Para que el robot sea estable, el ZMP debe estar dentro del polígono de soporte. Su posición se calcula como:

$$p_{x,yc} = r_{x,y}(a) - \frac{h}{g_z} \ddot{r}_{x,y}(a) \in P(p_{x,y}) \quad (20)$$

Donde:

- $r_{x,y}(a)$ es la posición del **Centro de Masa (CoM)**.
- h es la altura del CoM sobre los puntos de apoyo.
- g_z es la aceleración gravitacional.

Ejemplo: Si el CoM está en (0.5, 0.5) a 0.5 m del suelo, con una aceleración de 1 m/s² en la dirección x , entonces:

$$p_x = 0.5 - \frac{0.5}{9.81}(1) \approx 0.449 \quad (21)$$

El ZMP en el eje x se ha movido ligeramente debido a la aceleración del CoM.

10.2.4 Generalización en 2D

En una dimensión, la restricción se simplifica a:

$$p_{x1} < z_x < p_{x2} \quad (22)$$

En dos dimensiones, se verifica que el ZMP no cruce ninguno de los bordes del polígono de soporte.

Ejemplo: Si tenemos los puntos (0, 0) y (1, 0), la restricción en 1D es simplemente:

$$0 < z_x < 1 \quad (23)$$

10.2.5 Margen de Seguridad m

Para mejorar la estabilidad, se impone un margen de seguridad m , asegurando que el ZMP se mantenga alejado de los bordes del polígono.

Fórmula:

$$\begin{bmatrix} p_{xc}[t_k] & p_{yc}[t_k] & 1 \end{bmatrix} n_e > m \quad (24)$$

Ejemplo: Si el margen es $m = 0.1$, el ZMP debe mantenerse al menos 10 cm dentro del polígono para evitar inestabilidad.

10.2.6 Optimización Simultánea

La restricción de factibilidad dinámica se optimiza junto con:

- La ubicación de los puntos de apoyo, que afecta la forma del polígono de soporte.
- La aceleración del CoM, que afecta la ubicación del ZMP.

Ejemplo: Si el robot planea moverse rápidamente hacia adelante, los puntos de apoyo pueden ser ajustados más adelante para garantizar que el ZMP no se salga del polígono.

10.3 Restricciones cinematicas

Las restricciones cinemáticas establecen condiciones que aseguran que el robot se mueva de manera coherente y alcanzable, respetando sus limitaciones estructurales y articulares. Se dividen en restricciones de posición, velocidad y aceleración.

$$0 < \|\mathbf{p}_i^{xy} - \mathbf{r}^{xy}[t_k] - \mathbf{p}_{\text{nom}}^{B,xy}\|_2 < r \quad (25)$$

Esta ecuación impone una restricción de posición sobre la pata i del robot en el plano xy , asegurando que se mantenga dentro de un rango permitido con respecto a su posición nominal.

Donde:

- $\mathbf{p}_i^{xy} \rightarrow$ Posición de la pata i en el plano xy .
- $\mathbf{r}^{xy}[t_k] \rightarrow$ Posición del **centro de masa** del robot en el instante de tiempo t_k .
- $\mathbf{p}_{\text{nom}}^{B,xy} \rightarrow$ Posición **nominal** de la pata en relación con el cuerpo del robot (Es el punto donde la pata debería estar en condiciones normales de operación).
- $\|\cdot\|_2 \rightarrow$ **Norma euclidiana** o distancia en 2D.
- $r \rightarrow$ Radio máximo permitido para la extensión de la pata.

10.3.1 Interpretación Física

- La expresión $\mathbf{r}^{xy}[t_k] + \mathbf{p}_{\text{nom}}^{B,xy}$ define la **posición ideal** de la pata en el mundo, considerando el movimiento del centro de masa del robot.
- La ecuación establece que la pata no puede estar demasiado cerca del cuerpo (no menor a 0) ni alejarse más allá de r .
- Esta restricción evita que el robot estire demasiado las patas o las tenga en posiciones físicamente inalcanzables.

10.3.2 Ejemplo Numérico

Supongamos que la posición nominal de la pata es:

$$\mathbf{p}_{\text{nom}}^{B,xy} = (0.3, 0.2) \quad (26)$$

y que el centro de masa del robot en un instante t_k es:

$$\mathbf{r}^{xy}[t_k] = (1.0, 1.2) \quad (27)$$

Si la posición real de la pata es:

$$\mathbf{p}_i^{xy} = (1.4, 1.5) \quad (28)$$

Podemos calcular la distancia de la pata respecto a su posición nominal:

$$d = \|(1.4, 1.5) - (1.0, 1.2) - (0.3, 0.2)\|_2 \quad (29)$$

$$d = \|(0.1, 0.1)\|_2 = \sqrt{(0.1)^2 + (0.1)^2} = \sqrt{0.02} \approx 0.14 \quad (30)$$

Si el radio permitido es $r = 0.5$, como $0 < 0.14 < 0.5$, la restricción se cumple y la pata está en una posición alcanzable.

Esta restricción es importante ya que:

- La ecuación impone una restricción cinemática sobre la posición de las patas para que se mantengan dentro de un rango de movimiento válido.
- Evita que las patas del robot se extiendan más allá de un radio r o colapsen dentro del cuerpo.
- Garantiza que el robot mantenga una configuración estable y físicamente posible.

10.4 Restricciones de Continuidad del Centro de Masa (CoM)

Las restricciones de continuidad del **Centro de Masa (CoM)** aseguran que su movimiento sea suave y sin cambios bruscos en velocidad o aceleración.

Se exige que la posición y sus derivadas sean continuas en el tiempo:

$$\mathbf{r}^{xy}(t) \text{ y sus derivadas deben ser continuas.} \quad (31)$$

Esto implica que:

- La **posición** $\mathbf{r}^{xy}(t)$ no debe tener saltos.
- La **velocidad** $\dot{\mathbf{r}}^{xy}(t)$ debe ser continua.
- La **aceleración** $\ddot{\mathbf{r}}^{xy}(t)$ tampoco debe presentar discontinuidades.

10.5 Restricciones de Objetivo (Goal Constraints)

Estas restricciones aseguran que el centro de masa del robot alcance una posición y velocidad objetivo al final del movimiento planificado.

10.5.1 Ecuaciones Generales

$$\mathbf{r}^{xy}(t_f) = \mathbf{r}_{\text{goal}}^{xy} \quad (32)$$

$$\dot{\mathbf{r}}^{xy}(t_f) = \dot{\mathbf{r}}_{\text{goal}}^{xy} \quad (33)$$

Donde:

- $\mathbf{r}^{xy}(t_f)$ es la posición final deseada del centro de masa.

- $\dot{\mathbf{r}}^{xy}(t_f)$ es la velocidad deseada al final del movimiento.

Gracias a esto:

- Las restricciones de continuidad del CoM evitan cambios bruscos en el movimiento del robot, garantizando estabilidad.
- Las restricciones de objetivo aseguran que el robot alcance posiciones y velocidades finales planificadas con precisión.
- Son esenciales en la planificación de movimientos para robots caminantes, evitando caídas y asegurando trayectorias suaves.

10.6 Asignación de CoM y puntos de apoyo a estados de cuerpo completo

Se reconstruye el estado del cuerpo a partir del movimiento planificado del Centro de Masa (CoM) y la ubicación de los puntos de apoyo

10.6.1 Estado de la Base(tronco) del Robot

El estado de la base se define considerando que:

- El origen del sistema de referencia de la base está en el **CoM del robot**.
- La altura de la base se mantiene **constante** en h .
- La orientación inicial del cuerpo es **cero**.

Bajo estas condiciones, el estado de la base se expresa como:

$$\mathbf{q}_{b,\text{ref}}(t) = [0 \ 0 \ 0 \ r^x(t) \ r^y(t) \ h]^T. \quad (34)$$

10.6.2 Ejemplo

Supongamos que el CoM del robot sigue la trayectoria:

$$\begin{aligned} r^x(t) &= 0.5t, \\ r^y(t) &= 0.2t, \\ h &= 0.3 \text{ m.} \end{aligned}$$

Entonces, en $t = 2$ s, el estado de la base será:

$$\mathbf{q}_{b,\text{ref}}(2) = [0 \ 0 \ 0 \ 1.0 \ 0.4 \ 0.3]^T.$$

10.6.3 Estado de las Articulaciones de las Patas de Apoyo

El estado de las articulaciones de las patas de apoyo se obtiene mediante el uso de **cinemática inversa**, tomando como referencia la posición del centro de masa (**CoM**) y los puntos de apoyo optimizados.

- Cinemática inversa: La cinemática inversa (IK) se calcula determinando los ángulos de las articulaciones necesarios para que un efector final (como una pata robótica) alcance una posición y orientación específicas. El método puede depender de la complejidad del sistema.

10.6.4 Cálculo de las Coordenadas Articulares

Dado el estado de la base del robot en un instante de tiempo t , denotado como:

$$\mathbf{q}_{b,ref}(t) = [0 \ 0 \ 0 \ r^x(t) \ r^y(t) \ h]^T \quad (35)$$

donde:

- $r^x(t)$ y $r^y(t)$ representan la posición optimizada del **CoM** en el plano horizontal,
- h es la altura constante de la base,
- Se asume que la orientación de la base es cero.

Utilizando este estado base ($\mathbf{q}_{b,ref}(t)$) y la posición deseada de los puntos de apoyo de las patas \mathbf{p}^{xy} , la configuración articular de las patas de apoyo se calcula mediante la **cinemática inversa**:

$$\mathbf{q}_{j,ref}(t) = IK(\mathbf{q}_{b,ref}(t), \mathbf{p}^{xy}). \quad (36)$$

10.6.5 Estado Completo del Robot

Finalmente, combinamos el estado de la base y el de las articulaciones para obtener el estado completo:

$$\mathbf{q}_{ref}(t) = [\mathbf{q}_{b,ref}(t) \ \mathbf{q}_{j,ref}(t)]^T. \quad (37)$$

Este proceso permite mapear la trayectoria del CoM y los puntos de apoyo al estado completo del robot, asegurando que sus patas se posicionen correctamente y su cuerpo siga una trayectoria estable.

10.7 Seguimiento del movimiento planificado

Se debe lograr que el robot ejecute correctamente la trayectoria que se planeó en los pasos anteriores. Ya calculamos dónde debería estar su centro de masa (CoM) y las posiciones de sus patas, pero ahora hay que asegurarse de que realmente se mueva de esa manera.

Para esto, se usa un controlador de retroalimentación que compara el estado actual del robot con el estado deseado y ajusta la fuerza aplicada en sus articulaciones.

Este cálculo se basa en el modelo de dinámica del cuerpo rígido del sistema:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}}_{\text{ref}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}^T \boldsymbol{\tau} + \mathbf{J}_c(\mathbf{q})^T \mathbf{f}, \quad (38)$$

Donde:

- $\mathbf{M}(\mathbf{q})$ es la matriz de inercia del sistema.
- $\ddot{\mathbf{q}}_{\text{ref}}$ es la aceleración deseada del robot.
- $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ representa las fuerzas centrífugas y de Coriolis.
- \mathbf{S} es la matriz de selección que evita la acción directa sobre la base flotante.
- $\boldsymbol{\tau}$ representa los torques en las articulaciones.
- $\mathbf{J}_c(\mathbf{q})^T \mathbf{f}$ es la contribución de las fuerzas de contacto \mathbf{f} .

Si el robot tiene patas o ruedas que tocan el suelo, estas generan fuerzas de contacto \mathbf{f} que afectan el movimiento. Sin embargo, estas fuerzas no se pueden calcular directamente porque dependen de la fricción, la presión y otros factores.

Esto hace que la ecuación sea un sistema subdeterminado (más incógnitas que ecuaciones), lo que significa que no podemos resolverla directamente.

Dado que las fuerzas de contacto \mathbf{f} no pueden determinarse directamente, se introduce un operador de proyección:

$$\mathbf{P} = \mathbf{I} - \mathbf{J}_c^+ \mathbf{J}_c, \quad (39)$$

donde \mathbf{J}_c^+ es la pseudoinversa de la matriz Jacobiana de contacto. Este operador permite eliminar \mathbf{f} de la ecuación y calcular los torques articulares (acciones de control) de manera directa:

$$\boldsymbol{\tau} = (\mathbf{P}\mathbf{S}^T)^+ \mathbf{P}(\mathbf{M}\ddot{\mathbf{q}}_{\text{ref}} + \mathbf{C}). \quad (40)$$

Dado que los modelos utilizados en simulación no son perfectos en comparación con el sistema real, es esencial incorporar retroalimentación en el lazo de control. Para mejorar la precisión del seguimiento, se agregan controladores PD:

- Un **controlador PD en el espacio operacional** para la posición y velocidad de la base.
- Un **controlador PD de baja ganancia** para ajustar la posición y velocidad de las articulaciones.

Este enfoque es fundamental en robots con base flotante como cuadrúpedos, donde el control de contacto es clave para la estabilidad y el seguimiento preciso de trayectorias.

11 *Simultaneous foothold and body optimization)*

combina los enfoques rápidos basados en el Punto de Momento Cero (ZMP) con la optimización de trayectorias para mejorar la planificación del movimiento en robots cuadrúpedos. La metodología optimiza simultáneamente el movimiento del cuerpo, los puntos de apoyo y el centro de presión. Se introduce una representación basada en vértices para modelar restricciones en el área de soporte, permitiendo tratar contactos de diferentes tipos (puntuales, lineales y superficiales) de manera uniforme.

Gracias a esta generalización, el enfoque puede generar diversos patrones de locomoción como caminar, trotar, saltar, y realizar transiciones entre estos estados, además de movimientos en robots bípedos y recuperación ante empujes. La formulación empleada representa de manera mínima las leyes físicas (fuerzas de contacto unilaterales) y las restricciones cinemáticas, logrando generar movimientos en menos de un segundo. Finalmente, se valida la factibilidad de los movimientos generados en un robot cuadrúpedo físico.

11.1 Modelo

- RF: Right front
- LF: Left front
- RH: Right hind
- LH: Left hind

El robot con patas se modela utilizando el Modelo del Péndulo Invertido Lineal (LIPM), donde su Centro de Masa (CoM) $\mathbf{r} = (r_x, c_y)$ se mantiene a una altura constante h . La posición de contacto con el suelo, también conocida como ZMP (Zero Moment Point) o CoP (Center of Pressure), se define como $\mathbf{p}_c = (p_{c,x}, p_{c,y})$.

La aceleración del CoM está determinada por la física de un péndulo en caída, según la ecuación:

$$\begin{bmatrix} \dot{r} \\ \ddot{r} \end{bmatrix} = F(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \dot{r} \\ (r - p_c)gh^{-1} \end{bmatrix} \quad (41)$$

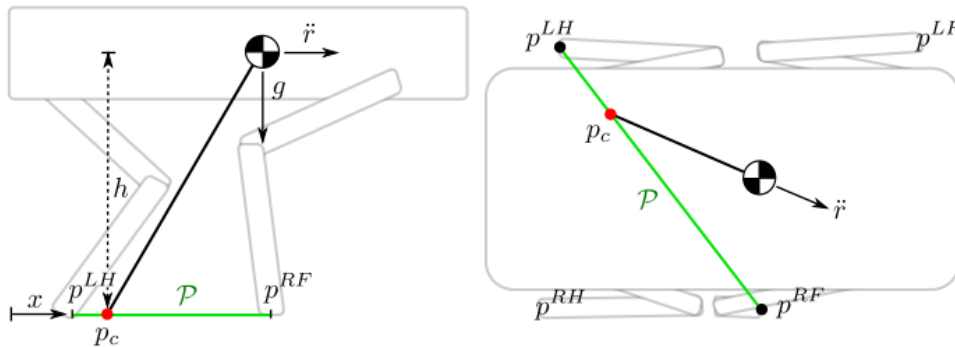


Figure 5: Modelo de un Legged Robot con LIPM.

- La dinámica del sistema se ve afectada por la posición del CoM (r), el CoP (p_c) y la gravedad (g).
- Este modelo permite describir un robot con patas, ya que el robot puede controlar los torques en las articulaciones y, por lo tanto, las fuerzas de contacto, lo que influye en la posición del CoP.
- Ejemplo en dirección x :
 - - Si el robot levanta la pata trasera, el sistema se comporta como un péndulo apoyado en la pata delantera con un punto de contacto $p_c = p^{RF} = (P_x, P_Y)^{RF}$.
 - - Si el robot levanta la pata delantera, el sistema se comporta como un péndulo apoyado en la pata trasera con $p_c = p^{LH}$, lo que genera una mayor aceleración hacia adelante.
- Al distribuir la carga entre ambas patas, el robot puede generar movimientos como si el péndulo estuviera anclado en un punto intermedio entre ambas patas.

11.1.1 Ecuación del modelo físico

$$\begin{bmatrix} \dot{r} \\ \ddot{r} \end{bmatrix} = F(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \dot{r} \\ (r - p_c)gh^{-1} \end{bmatrix} \quad (42)$$

Esta ecuación describe la dinámica del Centro de Masa (CoM) del robot basado en el modelo del péndulo invertido lineal (LIPM).

- $\dot{r} \rightarrow$ Es la velocidad del CoM.
- $\ddot{r} = (r - p_c)gh^{-1} \rightarrow$ Es la aceleración del CoM.
 - r : Posición del Centro de Masa (CoM).
 - p_c : Posición del Centro de Presión (CoP), que depende de la distribución de fuerzas en los pies.
 - g : Aceleración gravitatoria.
 - h : Altura constante del CoM sobre el suelo.
 - $(r - p_c)$: Distancia entre el CoM y el CoP, que determina la aceleración del robot.

Los estados y las entradas del sistemas estan dados por:

$$\mathbf{x}(t) = [\mathbf{r} \quad \dot{\mathbf{r}} \quad \mathbf{p}^1, \alpha^1, \dots, \mathbf{p}^n, \alpha^n]^T \quad (43)$$

$$\mathbf{u}(t) = \mathbf{p}_c \quad (44)$$

11.1.2 Interpretación Física

El término $(r - p_c)$ indica que si el CoM está alineado con el CoP, la aceleración es cero y el robot permanece estable.

- Si $r > p_c \Rightarrow$ El CoM está más adelante del CoP \Rightarrow el robot acelera hacia adelante.

- Si $r < p_c \Rightarrow$ El CoM está detrás del CoP \Rightarrow el robot acelera hacia atrás.

Esta ecuación se asemeja a la de un péndulo en caída, donde el movimiento del CoM depende de su inclinación con respecto al punto de apoyo (CoP).

11.1.3 Ejemplo

Supongamos que un robot bípedo tiene un CoM a una altura $h = 1$ m y que inicialmente su CoP está en $p_c = 0$.

Si en un instante dado el CoM está desplazado a $r = 0.2$ m respecto al CoP, la aceleración del CoM será:

$$\ddot{r} = (0.2 - 0) \cdot \frac{9.81}{1} = 1.962 \text{ m/s}^2 \quad (45)$$

Esto significa que el CoM comenzará a moverse hacia adelante con esta aceleración.

Si el robot quiere mantenerse en equilibrio, debe ajustar p_c moviendo los pies para mantener el CoM alineado con el CoP.

11.2 Movimiento del centro de masa

Se utilizarán n **splines polinomiocos** de cuarto grado para modelar la trayectoria del centro de masa. Esta aproximación permite optimizar el movimiento asegurando suavidad y continuidad en la trayectoria.

- Un spline es una función compuesta por varios polinomios definidos en diferentes intervalos, garantizando que la transición entre estos sea suave. En este caso, se utilizan polinomios de cuarto grado que permiten modelar con precisión la posición y velocidad del CoM sin generar discontinuidades en el movimiento. En términos generales, los splines se usan para representar trayectorias en sistemas mecánicos y robóticos donde es necesario mantener transiciones suaves entre segmentos.

11.2.1 Parametrización del CoM mediante splines

El movimiento del CoM se modela mediante una combinación de **polinomios cuárticos** que se definen en diferentes intervalos de tiempo $[t_k, t_{k+1}]$, asegurando continuidad en los puntos de conexión.

11.2.2 Expresión general del spline

$$\mathbf{x}(t) = \begin{bmatrix} r(t) \\ \dot{r}(t) \end{bmatrix} = \sum_{i=1}^4 \begin{bmatrix} (t - t_k)^i \\ i \end{bmatrix} a_{k,i} (t - t_k)^{i-1} + \begin{bmatrix} a_{k,0} \\ 0 \end{bmatrix} \quad (46)$$

donde:

- $r(t)$ representa la posición del CoM en un instante de tiempo t .

- $\dot{r}(t)$ es la velocidad del CoM.
- t : Es el tiempo continuo en el cual se evalúa la posición y velocidad del centro de masa. Representa el instante de tiempo en el que queremos calcular la trayectoria usando la función $\mathbf{x}(t)$.
- t_k : Es el tiempo en el que comienza el segmento k -ésimo del spline. Cada segmento del spline corresponde a un intervalo de tiempo específico, y t_k indica el inicio de ese intervalo.
- La expresión $(t - t_k)$ representa el tiempo transcurrido desde el inicio del segmento k . Como la trayectoria se modela con polinomios cuárticos (de grado 4), cada t_k marca el inicio de un nuevo polinomio que describe la posición y velocidad en ese segmento.
- Los términos $(t - t_k)^i$ forman un polinomio de cuarto grado con coeficientes $a_{k,i}$, que definen la forma de la curva en el intervalo correspondiente.
- $a_{k,0}$ es un término adicional que permite ajustar la posición inicial del polinomio.

Esta ecuación permite expresar la posición y velocidad del CoM dentro de cada segmento del spline.

11.2.3 Vector de parámetros del spline

$$\mathbf{w}_r = [a_{1,0}, \dots, a_{1,4}, \dots, a_{n,0}, \dots, a_{n,4}] \quad (47)$$

Este vector agrupa **todos los coeficientes** $a_{k,i}$ de los polinomios en una única estructura, permitiendo definir completamente la trayectoria del CoM y facilitando su optimización mediante técnicas de control.

11.2.4 Condiciones de continuidad entre segmentos del spline

$$\sum_{i=1}^4 \begin{bmatrix} T_k \\ i \end{bmatrix} a_{k,i} T_k^{i-1} + \begin{bmatrix} a_{k,0} \\ 0 \end{bmatrix} = \begin{bmatrix} a_{k+1,0} \\ a_{k+1,1} \end{bmatrix} \quad (48)$$

Para que la trayectoria del CoM sea **suave y sin saltos**, se deben cumplir condiciones de continuidad en los puntos de unión entre dos polinomios consecutivos (k y $k + 1$):

- (a) **Continuidad en la posición:** la posición final de un segmento debe coincidir con la posición inicial del siguiente.
- (b) **Continuidad en la velocidad:** la velocidad de transición entre polinomios debe ser la misma en ambos lados del punto de unión.

Donde:

- $T_k = t_{k+1} - t_k$ es el tiempo entre dos polinomios consecutivos.
- La ecuación asegura que los coeficientes del spline se ajusten para mantener una transición suave entre segmentos.

11.2.5 Restriccion dinamica

Para asegurar la consistencia entre el movimiento parametrizado y la dinámica del sistema, la integración de nuestra solución aproximada $\dot{r}(t)$ debe ser similar a la del sistema real. Esto se expresa como:

$$\int_{t_k}^{t_{k+1}} \ddot{r}(t) dt \approx \int_{t_k}^{t_{k+1}} \mathbf{F}_2(\mathbf{x}(t), \mathbf{u}(t)) dt. \quad (49)$$

La regla de Simpson nos dice que si $\ddot{r}(t)$ se elige como un polinomio de segundo orden (razón por la cual $r(t)$ se elige como un polinomio de cuarto orden) que coincide con la dinámica del sistema \mathbf{F}_2 al principio, en el centro y al final del intervalo, entonces el error en la aproximación es proporcional a $(t_{k+1} - t_k)^4$.

Por lo tanto, agregamos las siguientes restricciones para cada polinomio:

$$\ddot{r}[t] = \mathbf{F}_2(\mathbf{x}[t], \mathbf{u}[t]), \quad \forall t \in \{t_k, \frac{t_{k+1} - t_k}{2}, t_{k+1}\}. \quad (50)$$

Esto garantiza que la trayectoria generada siga las ecuaciones del sistema en estos puntos clave (inicio, punto medio, punto final), manteniendo el error de integración bajo.

Al mantener la duración de cada intervalo polinómico corta (alrededor de 50 ms), el error en la integración de Simpson se mantiene pequeño y la solución polinómica de cuarto orden $r(t)$ se aproxima bien a la solución real de la ecuación diferencia (42).

11.3 Movimiento de las patas

En la parametrización del movimiento de los pies durante la marcha se asume que cuando un pie está en contacto con el suelo, su posición \mathbf{p}_s^f y su orientación α_s^f permanecen constantes. Cuando el pie se mueve entre dos contactos, su trayectoria en el plano del suelo se modela mediante un polinomio cúbico:

$$\begin{bmatrix} \mathbf{p}^f(t) \\ \alpha^f(t) \end{bmatrix} = \sum_{i=0}^3 \begin{bmatrix} a_{s,i}^f \\ b_{s,i}^f \end{bmatrix} (t - t_s)^i \quad (51)$$

Donde:

- $(t - t_s)$ representa el tiempo transcurrido desde el inicio del movimiento del pie
- los coeficientes $a_{s,i}^f$ y $b_{s,i}^f$ se determinan en función de la duración del movimiento y las posiciones de contacto.
- La variable $\alpha^f(t)$ representa la **orientación** del pie durante la marcha. Esta indica el ángulo que el pie forma con respecto a un sistema de referencia fijo en el suelo. Es crucial para garantizar que el pie aterrice en la dirección correcta en cada paso, manteniendo estabilidad y control en la locomoción.

Los parámetros w_p representan las variables de decisión utilizadas para describir el movimiento de los pies en una caminata. Se definen como:

$$w_p = [w_p^1, \dots, w_p^{n_f}]$$

donde cada w_p^f para una pierna f está dado por:

$$w_p^f = [p_s^f, \alpha_s^f, \dots, p_{n_s}^f, \alpha_{n_s}^f]$$

11.3.1 Descripción de los elementos

- p_s^f : Posición inicial del pie f .
- α_s^f : Orientación inicial del pie f .
- $p_{n_s}^f$: Posición del pie después de n_s pasos.
- $\alpha_{n_s}^f$: Orientación del pie después de n_s pasos.

Los parámetros w_p provienen del problema de optimización de caminata del robot. Se generan dentro de un NLP (Nonlinear Programming Problem), que busca minimizar un costo (energía, estabilidad, etc.).

La solución óptima del NLP entrega la secuencia de posiciones y orientaciones de los pies que permiten una caminata eficiente y estable.

11.3.2 Ejemplo de optimización

Si un robot debe caminar minimizando su consumo de energía, el algoritmo de optimización ajusta los valores de w_p para encontrar la trayectoria de marcha más eficiente, asegurando que cada paso esté bien posicionado y orientado.

11.4 Movimiento del centro de presión

se parametriza el Centro de Presión (CoP) a partir de la carga soportada por cada esquina del pie del robot.

11.4.1 Parametrización del CoP

La carga sobre cada pie en el tiempo t se representa mediante un vector:

$$\lambda(t) = [\lambda^1(t), \dots, \lambda^{n_f}(t)]^T \quad (52)$$

donde n_f es el número de pies en contacto con el suelo. Cada pie tiene **varios vértices o esquinas** que distribuyen la carga, representados por un vector adicional:

$$\lambda^f(t) = [\lambda_1^f(t), \dots, \lambda_{n_v}^f(t)] \quad (53)$$

donde n_v es el número de vértices del pie f .

Ejemplo: Si el robot tiene un pie cuadrado con 4 esquinas, el peso que soporta ese pie se distribuye entre sus esquinas con cuatro valores $\lambda_1, \lambda_2, \lambda_3, \lambda_4$.

Si $|\lambda^f(t)|_1 = 0.9$, significa que el pie f soporta el **90% del peso del robot** en ese instante.

11.4.2 Cálculo del Centro de Presión (CoP)

El CoP $p_c(t)$ se obtiene como una combinación ponderada de las posiciones de los pies y sus vértices:

$$p_c(t) = \sum_{j=1}^{n_f} \sum_{i=1}^{n_v} \lambda_i^j (p^j + \mathbf{R}(\alpha^j(t))v_i) \quad (54)$$

donde:

- p^j es la posición del pie j .
- $\mathbf{R}(\alpha^j(t))$ es la **matriz de rotación** del pie.
 - * La matriz de rotación del pie, denotada como $\mathbf{R}(\alpha^j(t))$, es una matriz de rotación 2×2 que transforma coordenadas del sistema de referencia del pie a la orientación optimizada en el espacio global (este sistema de referencia es independiente del robot y permite expresar la posición y orientación de sus partes en un marco común).
 - * ¿Para qué se usa? Se emplea para representar la orientación del pie durante el movimiento del robot con patas, permitiendo que la posición de cada vértice del pie se exprese correctamente en el sistema de referencia global.
- v_i es la posición de la esquina i del pie en su propio sistema de referencia.

Ejemplo: Si un pie cuadrado tiene vértices en posiciones relativas $(-0.1, -0.1)$, $(-0.1, 0.1)$, $(0.1, -0.1)$, $(0.1, 0.1)$ metros y la distribución de la carga es $\lambda = [0.3, 0.2, 0.4, 0.1]$, el CoP se ubicará más cerca del vértice con **mayor peso asignado**.

Para un **robot con pies puntuales** (sin esquinas), la ecuación se simplifica a:

$$p_c = \sum_{f=1}^{n_f} \lambda^f p^f \quad (55)$$

donde el CoP es simplemente la combinación lineal (suma) de las posiciones de los pies.

11.4.3 Implementación Discretizada

Dado que el tiempo es continuo, $\lambda(t)$ se discretiza cada **20 ms**, generando una serie de valores constantes $\lambda_i = \lambda(t_i)$ a lo largo del movimiento.

Las variables de decisión adicionales utilizadas en **Programación No Lineal (NLP)** para optimizar el CoP son:

$$w_u = [\lambda_1, \dots, \lambda_{n_u}] \quad (56)$$

donde n_u es el número total de nodos de tiempo.

Esta parametrización del CoP permite un control preciso del equilibrio del robot al distribuir el peso dinámicamente en los pies y sus vértices. Este enfoque mejora la estabilidad del robot en superficies irregulares y situaciones de alta carga.

11.5 Generación de aceleraciones de referencia del cuerpo completo

Se describe cómo se generan las aceleraciones de referencia para el cuerpo completo de un robot con una base flotante de 6 grados de libertad.

Para reconstruir la pose de la base del robot con **6 grados de libertad (6-DoF)**, se considera que la posición del **centro de masa (CoM)** es r , con una altura constante h . La pose de referencia de la base en términos de aceleración es:

$$q_{b,\text{ref}}(t) = [0 \ 0 \ 0 \ r_x(t) \ r_y(t) \ h]^T \quad (57)$$

donde:

- $r_x(t)$ y $r_y(t)$ son las coordenadas del centro de masa en el plano horizontal.
- h es la altura fija de la base.
- Los primeros tres valores corresponden a una orientación inicial nula.

Para manejar incertidumbres en el control, se aplica un **controlador PD (Proporcional-Derivativo)** en el espacio operacional:

$$\ddot{q}_{b,\text{ref}} = \ddot{q}_{b,\text{ff}} + K_p(q_b - q_{b,\text{ref}}) + K_d(\dot{q}_b - \dot{q}_{b,\text{ref}}) \quad (58)$$

donde:

- $\ddot{q}_{b,\text{ref}}$ es la aceleración de referencia corregida de la base.
- $\ddot{q}_{b,\text{ff}}$ es la aceleración de alimentación directa calculada mediante optimización.
- q_b representa la pose de la base del robot, es decir, la posición y orientación de la base flotante en el espacio.
- K_p y K_d son las ganancias proporcional y derivativa, respectivamente.

11.5.1 Cálculo de aceleraciones articulares deseadas

La relación entre la velocidad de los pies $\dot{p}(t)$ y la velocidad del cuerpo completo es:

$$\ddot{p}(t) = J\ddot{q} + \dot{J}\dot{q} \quad (59)$$

donde:

- $\dot{p}(t)$ representa las aceleraciones de los n pies o puntos de contacto en el espacio.
- J es el Jacobiano del sistema que mapea Velocidades de cuerpo completo a velocidades lineales del pie.

$$J = [J_b \ J_j] \quad (60)$$

- * J_b es el Jacobiano asociado al tronco
- * J_j es el Jacobiano asociado a las extremidades.

Para obtener las **aceleraciones articulares de referencia**, se despeja \ddot{q}_j y se aplica la **pseudoinversa de Moore-Penrose** J_f^+ :

$$\ddot{q}_{j,\text{ref}} = J_j^+ \left(\ddot{p} - \dot{J}\dot{q} - J_b\ddot{q}_{b,\text{ref}} \right) \quad (61)$$

11.6 Dinámica inversa para calcular torques articulares

El controlador de dinámica inversa calcula los torques articulares τ necesarios para seguir la aceleración de referencia \ddot{q}_{ref} , basada en el modelo de dinámica del cuerpo rígido.

Dado que este modelo también involucra fuerzas de contacto desconocidas, estas se eliminan proyectando la ecuación en el espacio de torques articulares mediante la matriz $P = I - J_c^+ J_c$, donde J_c^T es la Jacobiana de contacto que asigna fuerzas de contacto cartesianas a torques articulares.

Así, los torques requeridos se calculan como:

$$\tau = (PS^T)^+ P(M\ddot{q}_{\text{ref}} + C)$$

donde:

- M es la matriz de inercia en el espacio articular.
- C representa los efectos de Coriolis.
- S es una matriz de selección que evita la activación de la base flotante.

Además, se recomienda agregar un controlador PD de baja ganancia para compensar errores en la dinámica y en el seguimiento de fuerzas.

References

- [1] Alexander W. Winkler. “Optimization-Based Motion Planning for Legged Robots”. Doctor of Sciences Thesis. Switzerland: ETH Zurich, 2018. URL: <https://www.research-collection.ethz.ch/handle/20.500.11850/272432>.
- [2] Alexander W. Winkler et al. “Fast Trajectory Optimization for Legged Robots using Vertex-based ZMP Constraints”. In: *Robotics and Automation Letters RA-L* (2017).
- [3] Alexander W. Winkler et al. “Gait and Trajectory Optimization for Legged Systems through Phase-based Endeffector Parameterization”. In: *Robotics and Automation Letters RA-L* (2018).
- [4] Alexander W. Winkler et al. “Online Walking Motion and Foothold Optimization for Quadruped Locomotion”. In: *IEEE International Conference on Robotics and Automation*. Vol. ICRA. 2017.