

Optimización Generación de Movimiento de Caminata en Línea con Desplazamientos Automáticos Proyecto Final

24 de mayo de 2021

Ruth Ivonne Escobedo
ruth.escobedo@cimat.mx
Erika Rivadeneira Pérez
erika.rivadeneira@cimat.mx
Matemáticas Aplicadas - CIMAT

1. Resumen

El objetivo de este proyecto es comparar diferentes métodos de optimización para un modelo de control predictivo capaz de generar movimientos estables de marcha a partir del control del *zero momento point* ZMP de un robot bípedo, el cual se modela como un carro que se mueve sobre un plano horizontal.

2. Introducción

La motivación de este proyecto se basa en el campo de estudio de la marcha de los robots humanoides, donde el uno de los temas de mayor importancia es la generación de patrones de marcha de los mismos.

Este tipo de estudios requiere de conocimientos de control y de la dinámica del robot, ubicación del centro de masas CoM, la inercia entre articulaciones, entre otros conceptos mecánicos, los cuales se basan en construir patrones basados en ZMP para el control de la marcha.

Aunque, en ciertas ocasiones, el conocimiento dinámico es limitado, se ha tenido la necesidad de optar por una metodología de control de retroalimentación y la idea del péndulo invertido.

Este nuevo esquema de usar las técnicas de retroalimentación, permiten controlar la marcha de robot colocando de manera automática la posición del pie en cada paso. Lo que permite generar trayectorias seguras sin importar la existencia de perturbaciones, todo esto controlando la velocidad de referencia de la caminata.

Dado el grado dinámico de la caminata de un robot, de acuerdo con la relación entre el CoM, ZMP y los polígonos convexos de apoyo, la caminata se puede dividir en caminata estática, caminata cuasi dinámica y caminata dinámica. En la caminata estática, la proyección del CoM del robot en el suelo nunca supera el rango de los polígonos de apoyo [1]. En la caminata dinámica, la proyección de CoM puede superar el polígono de apoyo en algún punto, lo que requiere un control en línea de CoM para generar un patrón de caminata estable [2].

El desplazamiento del centro de masa (CoM) depende de las fuerzas de contacto entre la base del robot y el suelo, con la limitación de que la base solo puede empujar el suelo, esta es la dificultad para generar un movimiento de caminata estable ya que esto restringe los movimientos que puede realizar un sistema para caminar. Por tanto, el problema de interés radica en generar una marcha en línea, adaptando los movimientos continuamente a la dinámica actual del sistema.

3. Revisión del Estado del Arte

El modelo de control predictivo original (MPC) usando por S. Kajita [7] para generar movimientos de caminata basándose en el movimiento del CoM del robot muestran una mayor accesibilidad para sistemas rápidos que requieren de cálculos cortos, además, se muestra que los esquemas lineales de control predictivo (LMPC) permiten bucles de control más rápidos.

Para obtener el esquema LMPC de la caminata de un robot, se restringe la caminata a un plano horizontal constante, por lo que el CoM también se mueve en un plano horizontal a una altura h del suelo, lo que permite representar la posición de esté en el espacio como un punto (x, y) . De tal manera que si se tiene perturbaciones en x y/o y en cierto intervalo de tiempo se puede calcular la dinámica en tiempo discreto.

Usando la dinámica de forma recursiva podemos encontrar relaciones entre las perturbaciones, su posición y velocidad y el posición del ZMP en intervalos de tiempo más largos.

El objetivo del esquema MPC [7] es equilibrar sobre el plano de predicción de la minimización de la perturbación y del CoM con el seguimiento de una posición teniendo un aproximación de la posición del ZMP correspondiente al movimiento se puede desprestigiar los efectos de inercia relacionados por rotaciones del robot referencia ZMP, el cual se localiza en un punto medio del polígono representativo del robot con el fin de controlar mejor la respuesta ante perturbaciones.

Todo este proceso de minimizar las perturbaciones se realiza por medio de la programación cuadrática (QP), el cual es un proceso de solución de problemas de optimización que involucran funciones cuadráticas. Lo que busca QP es optimizar una función multivariable sujeta a restricciones lineales.

$$\min_{\ddot{X}_k, \ddot{Y}_k} \frac{\alpha}{2} \|\ddot{X}_k\|^2 + \frac{\gamma}{2} \|Z_{k+1}^x - Z_{k+1}^{x-ref}\|^2 + \frac{\alpha}{2} \|\ddot{Y}_k\|^2 + \frac{\gamma}{2} \|Z_{k+1}^y - Z_{k+1}^{y-ref}\|^2 \quad (1)$$

Por otro lado, Wieber en [9], menciona que la posición del ZMP mediante QP puede ser reemplazada haciendo cumplir la restricción de que el ZMP se mantenga dentro del polígono de apoyo en todo tiempo. Y en [10] da un análisis del MPC, el cual muestra que minimizar el movimiento del CoM del robot con las restricciones en el ZMP genera una estabilidad en la marcha en línea recta. Lo cual se puede emplear para una marcha con escalones, los cuales deben ser colocados con anticipación.

Diedam [11], propuso una adaptación en línea de las posiciones de los pies usando el mismo esquema predictivo, con la modificación de agregar nuevas variables correspondientes al número de pasos que tiene lugar la predicción. Estas variables fueron adaptadas en el proceso de minimización, controlando la posición del pie frente a un escalón.

Estas observaciones provocan que sea posible generar una marcha regulando la velocidad del CoM a un valor medio deseado, logrando que la nueva posición del pie sea automática.

Aún así, esta método de marcha provoca un balanceo lateral el cual se puede minimizar considerando un QP de la forma:

$$\min_{u_k} \frac{\beta}{2} \|\dot{X}_{k+1} - \dot{X}_{k+1}^{ref}\|^2 + \frac{\beta}{2} \|\dot{Y}_{k+1} - \dot{Y}_{k+1}^{ref}\|^2$$

En pocas palabras, lo que hace el QP es conociendo la posición de un pie (el cual esta fijo) determinará la posición del segundo pie para lograr el paso.

Finalmente, en [2] se menciona que la fase de soporte simple (un pie en el suelo) involucra restricciones que se pueden expresar como un conjunto de desigualdades lineales sobre la posición del ZMP. Pero en el caso del doble soporte, estas desigualdades son cuadráticas, las cuales son mejor evitar desde el punto de vista computacional.

Una de las observaciones más importante, dada por Wieber, es que durante los tiempos de transición durante la marcha, las limitaciones de soporte simple y doble se aplican, donde las restricciones del soporte simple son suficientes por el número de restricciones que conllevan.

Por lo que se busca crear un periodo de tamaño T que cumple las restricciones de la posición del ZMP, el cual debe ser estrictamente igual a la longitud del tiempo de soporte doble, evitando de esta manera que el tiempo de muestre caiga en el soporte doble, de esa manera solo se consideran las restricciones del soporte simple.

4. Desarrollo

Cuando se representa un modelo de caminata robótica controlando la trayectoria de centro de masas (CoM) de un robot, facilita el cálculo del sistema del ZMP.

Para el ZMP, se define una variable u_x como la derivada en el tiempo de la aceleración horizontal del CoM.

Entonces la ecuación que define al ZMP es un sistema dinámico respecto a x está dado por

$$\frac{d}{dt} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_x$$

y

$$p_x = \begin{bmatrix} 1 & 0 & -z_c/g \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix}.$$

De igual forma se tienen las ecuaciones para u_y

$$\frac{d}{dt} \begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_y$$

y

$$p_y = \begin{bmatrix} 1 & 0 & -z_c/g \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \end{bmatrix}.$$

Donde para generar una caminata continua es necesario trabajar el ZMP por pequeños intervalos de segmentos de la trayectoria.

Para ello se asumirá que la trayectoria de la caminata del robot se generará en un plano horizontal, entonces el CoM se trabaja en un plano horizontal con una distancia h del suelo, esto se puede representar usando solo dos variables (x, y) .

Se consideran entonces trayectorias para el CoM que pueden sufrir de perturbaciones a lo largo de intervalos de tiempo T que corresponden a tiempos discretos (t_k) , las cuales se escriben como

$$\begin{aligned} \hat{x}_{k+1} &= A\hat{x}_k + B\ddot{x}(t_k) \\ \hat{y}_{k+1} &= A\hat{y}_k + B\ddot{y}(t_k) \end{aligned} \tag{2}$$

con

$$\hat{x}_k = \begin{pmatrix} x(t_k) \\ \dot{x}(t_k) \\ \bar{x}(t_k) \end{pmatrix}, \quad \hat{y}_k = \begin{pmatrix} y(t_k) \\ \dot{y}(t_k) \\ \bar{y}(t_k) \end{pmatrix}$$

y

$$A = \begin{pmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{pmatrix}, B = \begin{pmatrix} T^3/6 \\ T^2/2 \\ T \end{pmatrix}.$$

Además, se debe considerar una aproximación de la posición (z_x, z_y) del ZMP, despreciando los efectos inerciales del robot, esto es

$$\begin{aligned} z_k^x &= (1 \quad 0 - h/g)\hat{x}_k, \\ z_k^y &= (1 \quad 0 - h/g)\hat{y}_k, \end{aligned}$$

donde h es la altura constante del CoM sobre el suelo, y g la fuerza de gravedad.

Existe una relación entre las perturbaciones, la posición (X), velocidad (\dot{X}) y el ZMP (Z^x) dentro de intervalos de tiempo de longitud NT , de la forma

$$\begin{aligned} X_{k+1} &= \begin{pmatrix} x_{k+1} \\ \vdots \\ x_{k+N} \end{pmatrix} = P_{ps}\hat{x}_k + P_{pu}\ddot{X}_k, \\ \dot{X}_{k+1} &= \begin{pmatrix} \dot{x}_{k+1} \\ \vdots \\ \dot{x}_{k+N} \end{pmatrix} = P_{vs}\hat{x}_k + P_{vu}\ddot{X}_k, \\ Z_{k+1}^x &= \begin{pmatrix} z_{k+1}^x \\ \vdots \\ z_{k+N}^x \end{pmatrix} = P_{zs}\hat{x}_k + P_{zu}\ddot{X}_k. \end{aligned}$$

Las relaciones para el eje vertical son similares:

$$\begin{aligned} Y_{k+1} &= \begin{pmatrix} y_{k+1} \\ \vdots \\ y_{k+N} \end{pmatrix} = P_{ps}\hat{y}_k + P_{pu}\ddot{Y}_k, \\ \dot{Y}_{k+1} &= \begin{pmatrix} \dot{y}_{k+1} \\ \vdots \\ \dot{y}_{k+N} \end{pmatrix} = P_{vs}\hat{y}_k + P_{vu}\ddot{Y}_k, \\ Z_{k+1}^y &= \begin{pmatrix} z_{k+1}^y \\ \vdots \\ z_{k+N}^y \end{pmatrix} = P_{zs}\hat{y}_k + P_{zu}\ddot{Y}_k, \end{aligned}$$

con

$$\ddot{X}_k = \begin{pmatrix} \ddot{x}_k \\ \vdots \\ \ddot{x}_{k+N-1} \end{pmatrix} \quad \text{y} \quad \ddot{Y}_k = \begin{pmatrix} \ddot{y}_k \\ \vdots \\ \ddot{y}_{k+N-1} \end{pmatrix}.$$

Las matrices $P_{ps}, P_{vs}, P_{zs} \in \mathbb{R}^{N \times 3}$ and $P_{pu}, P_{vu}, P_{zu} \in \mathbb{R}^{N \times N}$ se obtienen de manera recursiva trabajando con (2).

Ahora, para la minimización de las perturbaciones (\ddot{X}, \ddot{Y}) , se trabaja con la función objetivo descrita en

(1), donde si definimos

$$\begin{aligned} u_k &= \begin{pmatrix} \ddot{X}_k \\ \ddot{Y}_k \end{pmatrix}, \\ Q &= \begin{pmatrix} Q' & 0 \\ 0 & Q' \end{pmatrix}, \\ Q' &= \alpha I + \gamma P_{zu}^T P_{zu} \end{aligned}$$

y

$$p_k = \begin{pmatrix} \gamma P_{zu}^T \left(P_{zs} \hat{x}_k - Z_{k+1}^{x-ref} \right) \\ \gamma P_{zu}^T \left(P_{zs} \hat{y}_k - Z_{k+1}^{y-ref} \right) \end{pmatrix}.$$

Podemos escribir (1) de manera canónica como

$$\min_{u_k} \frac{1}{2} u_k^T Q u_k + p_k^T u_k \quad (3)$$

5. Metodología

El objetivo es minimizar las perturbaciones u_k del problema cuadrático (3) para lo cual se consideraron cuatro métodos de optimización. Tres de los métodos se estudiaron en clase, estos son: BFGS, Descenso por gradiente y Barzilai-Borwein. Adicionalmente, se considera el método de búsqueda en línea no-monótona de *Zhang-Hager*, el cual es una combinación del método de Barzilai-Borwein con búsqueda en línea no-monótona.

5.1. Algoritmos

A continuación se presentan los algoritmos de los métodos implementados

5.1.1. Método BFGS

BFGS
Input: $u_{k0} H_0$
1: $k = 0$
2: while $\ \nabla f(u_k)\ \neq 0$
3: $d_k = -H_k \nabla f_k$
4: Calcular α_k usando un método de búsqueda en línea
5: $u_{k+1} = x_k + \alpha_k d_k$
6: Calcular $\nabla f_{k+1}, y_k, s_k, \rho_k$ actualizar H_{k+1}
7: $H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T$
8: $k = k + 1$
9: end while
Output: u_k^*

5.1.2. Gradiente descendiente

Algoritmo Gradiente descendiente
Input: u_{k0}
1: $k = 0, g_0 = \nabla f(x_0)$
2: while $\ g_k\ \neq 0$ do
3: $\alpha_k = \text{método}$
4: $u_{k+1} = u_k - \alpha_k g_k$
5: $g_{k+1} = \nabla f(u_{k+1})$
6: $k = k + 1$
7: end while
Output: u_k^*

5.1.3. Método Barzilai-Borwein

Gradiente Barzilai-Borwein
Input: u_{k0}
1: $\lambda_0 = 0, \mathbf{y}_0 = \mathbf{u}_0, k = 0$
2: while $\ \mathbf{g}_k\ > \tau_g$ do
3: $\alpha_k = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}$
4: $\mathbf{u}_{k+1} = \mathbf{u}_k - \alpha_k \mathbf{g}_k$
5: $k = k + 1$
6: end while
Output: u_k^*

5.1.4. Zhang-Hager

Búsqueda en línea no-monótona de Zhang-Hager
Input: $u_0, \epsilon \in (0, 1), c_1, \eta \in (0, 1)$ 1: Hacer $k = 0$
$C_0 = f(u_0)$
2: while $\ \nabla f(u_k)\ > \epsilon$ do
3: Calcular α_k que satisfaga la condición de Armijo: $f(u_k + \alpha d_k) \leq C_k + c_1 \alpha g_k^T d_k$ (suficientemente descendiente)
4: $x_{k+1} = x_k + \alpha_k d_k$
5: $Q_{k+1} = \eta Q_k + 1$ $C_{k+1} = (\eta Q_k C_k + f(u_{k+1}))/Q_{k+1}$
6: $k = k + 1$
7: end while
Output: u_k^*

6. Resultados

Para la creación de la función objetivo (3) se consideraron los valores y matrices iniciales $N = 16$, $T = 0,1$, $m = 2$, $\ddot{x} = [0, 0, \dots, 0, 0]_N^T$ y $\ddot{y} = [1, 1, \dots, 1, 1]_N^T$. Además, sabiendo que el gradiente de la función objetivo es

$$\nabla f(u_k) = Q u_k + p_k$$

se obtuvieron los siguiente resultados (ver tabla (1))

	BFGS	Descenso por Gradiente	Barzilai	Zhang
Iteraciones	2	7094	10	9
Tiempo (s)	0.00146	0.64	0.013	0.004
$\ g(u_k^*)\ $	0.4217	0.0998	0.1146	0.0068

Tabla 1: Iteraciones, tiempo de compilación y error de los métodos BFGS, descenso por gradiente, Barzilai-Borwein y Zhang-Hager

Para obtener estos resultados se consideró una tolerancia de 1×10^{-4} , un número máximo de iteraciones de 27000 y adicionalmente se tomó a $\|g(u_k) - g(u_{k-1})\| < \text{tolerancia}$ como criterio extra de paro. Finalmente, en las gráficas (1) y (2) se muestran el número de iteraciones vs. la norma del gradiente de la función objetivo para cada método.

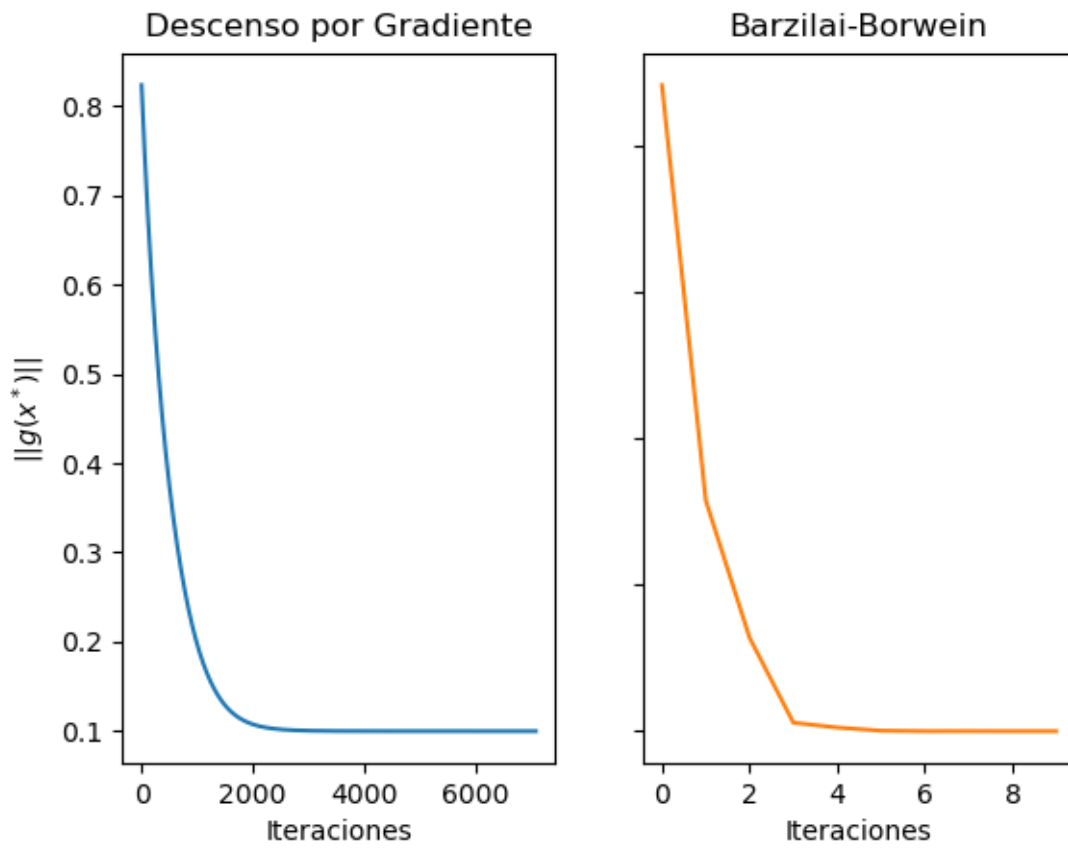


Figura 1: Iteraciones vs. $\|g(u_k^*)\|$ de Descenso por gradiente y Barzilai-Borwein

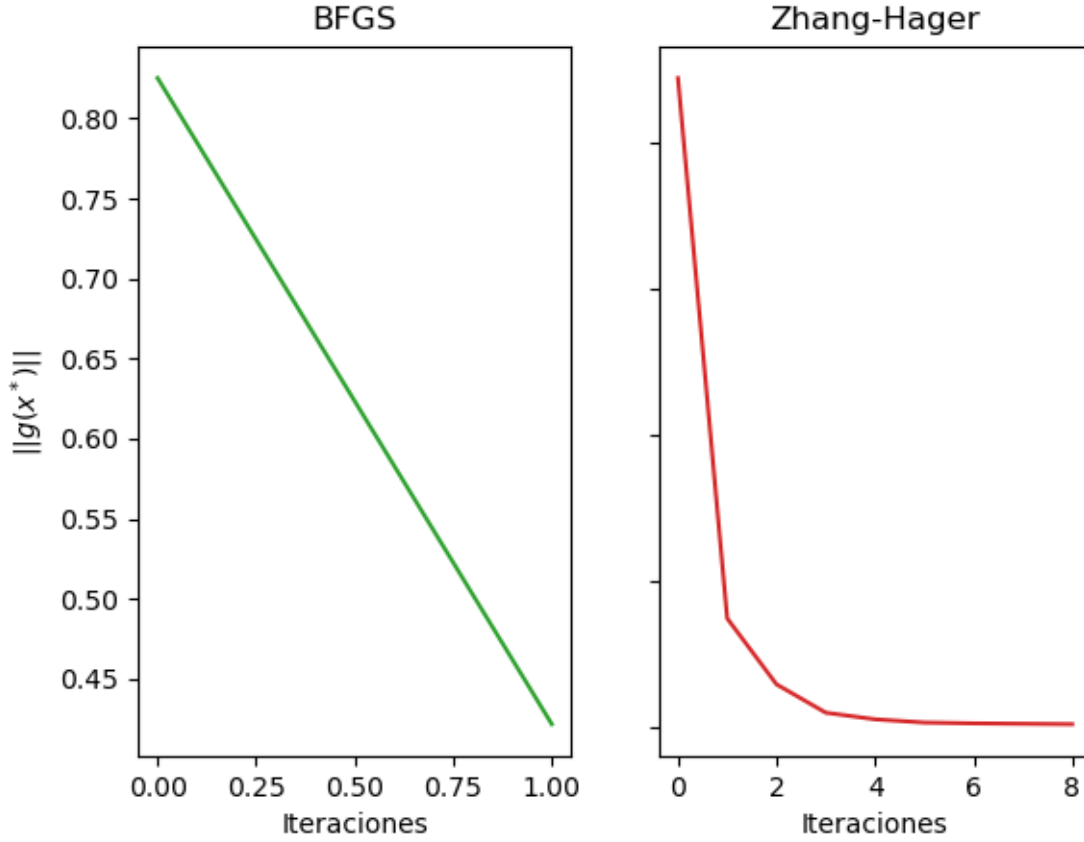


Figura 2: Iteraciones vs. $\|g(u_k^*)\|$ de BFGS y Zhang-Hager

7. Conclusiones y Discusión

Se puede observar que el método BFGS obtuvo un menor número de iteraciones y de tiempo de cómputo en relación al resto de métodos lo cual, pero su error en cuanto a la norma del gradiente, $\|g(u_k^*)\| = 0,4217$, es el mayor de todos los métodos (véase tabla (1)). El método que obtuvo mejores resultados fue el de Zhang-Hager ya que en pocas iteraciones y tiempo de cómputo obtuvo el mínimo u_k^* con menor norma del gradiente (0.0068).

Todos los métodos convergieron a un mínimo de la función objetivo (3). Entonces, dados Q , u_k y p_k del nuevo paso de la caminata se ha conseguido minimizar las perturbaciones (\ddot{x}, \ddot{y}) , lo cual es uno de los objetivos principales del proyecto. Adicionalmente, se intentó sin éxito la construcción del modelo de desplazamiento automático para el cómputo de u_k , Q y p_k de un nuevo paso, lo cual se puede considerar como un objetivo a desarrollar a futuro en conjunto con el agregar restricciones del centro de presión y de desplazamientos de un robot bípedo.

Referencias

- [1] Sugihara T, Yamamoto T. Foot-guided agile control of a biped robot through ZMP manipulation. in proc. IEEE/RSJ International Conference on Intelligent Robots Systems, Canada, Vancouver; 2017. p. 4546–51.
- [2] Geilinger M, Poranne R, Desai R, Thomaszewski B, Coros S. Skaterbots: optimization-based design and motion synthesis for robotic creatures with legs and wheels. ACM Trans Graph. 2018;37(4):1–12.

- [3] P.-B. Wieber. On the stability of walking systems. In Proceedings of the International Workshop on Humanoid and Human Friendly Robotics, 2002.
- [4] P.-B. Wieber. Holonomy and nonholonomy in the dynamics of articulated motion. In Proceedings of the Ruperto Carola Symposium on Fast Motion in Biomechanics and Robotics, 2005.
- [5] P.-B. Wieber and C. Chevallereau. Online adaptation of reference trajectories for the control of walking systems. *Robotics and Autonomous Systems*, 54(7), 2006.
- [6] 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. In Proceedings of the IEEE International Conference on Robotics Automation, 2003.
- [7] S. Kanzaki, K. Nishiwaki, K. Okada, and M. Inaba. Bracing against impact in a humanoid using disturbance preview control. In Proceedings of the Annual Conference on Robotics Society of Japan, 2004.
- [8] K. Nishiwaki and S. Kagami. High frequency walking pattern generation based on preview control of zmp. In Proceedings of the IEEE International Conference on Robotics Automation, 2006.
- [9] P.-B. Wieber. Trajectory free linear model predictive control for stable walking in the presence of strong perturbations. In Proceedings of the International Conference on Humanoid Robotics, 2006.
- [10] P.-B. Wieber. Viability and predictive control for safe locomotion. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots Systems, 2008.
- [11] H. Diedam, D. Dimitrov, P.-B. Wieber, K. Mombaur, and M. Diehl. Online walking gait generation with adaptive foot positioning through linear model predictive control. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots Systems, 2008.