

Modelamiento de Legged Robots

Seykarin Mestre Muelas

File Panels Help

Interact Select Measure 2D Pose Estimate 2D Nav Goal Publish Point

Displays

- Global Options
- Global Status: Ok
- Grid
- RobotModel
- TF
- Optimized State...
- Current State
- Target
- GridMap

¿Por que modelar antes de simular?

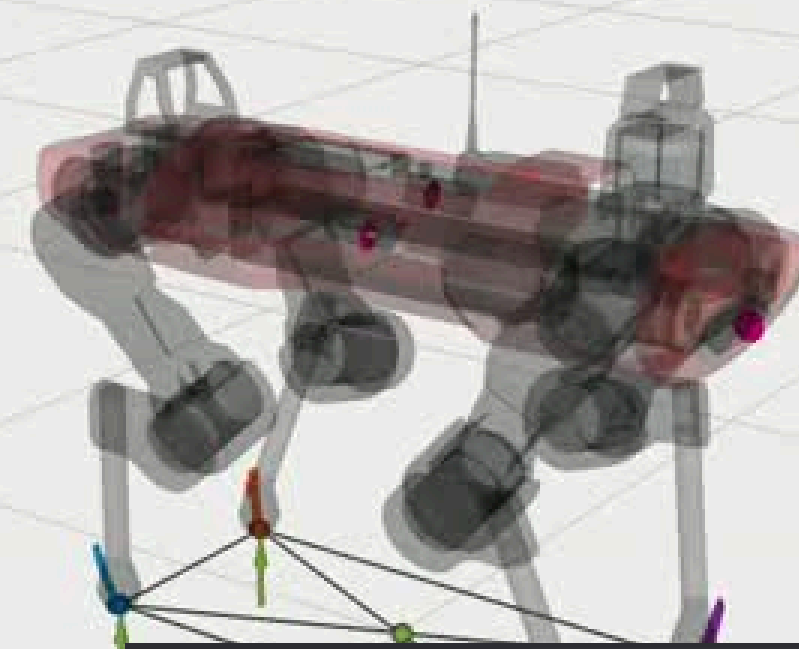
- Para trabajar con la simulación, es necesario conocer el modelo.
- Para trabajar con la documentación, es necesario entender el modelo.

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

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

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.



OCS2

Search docs

Overview

Installation

Getting Started

Where to start?

The optimal control formulation

How to setup an optimal control problem?

How to setup an MPC loop?

MPC Interface

MRT Interface

ROS and non-ROS versions

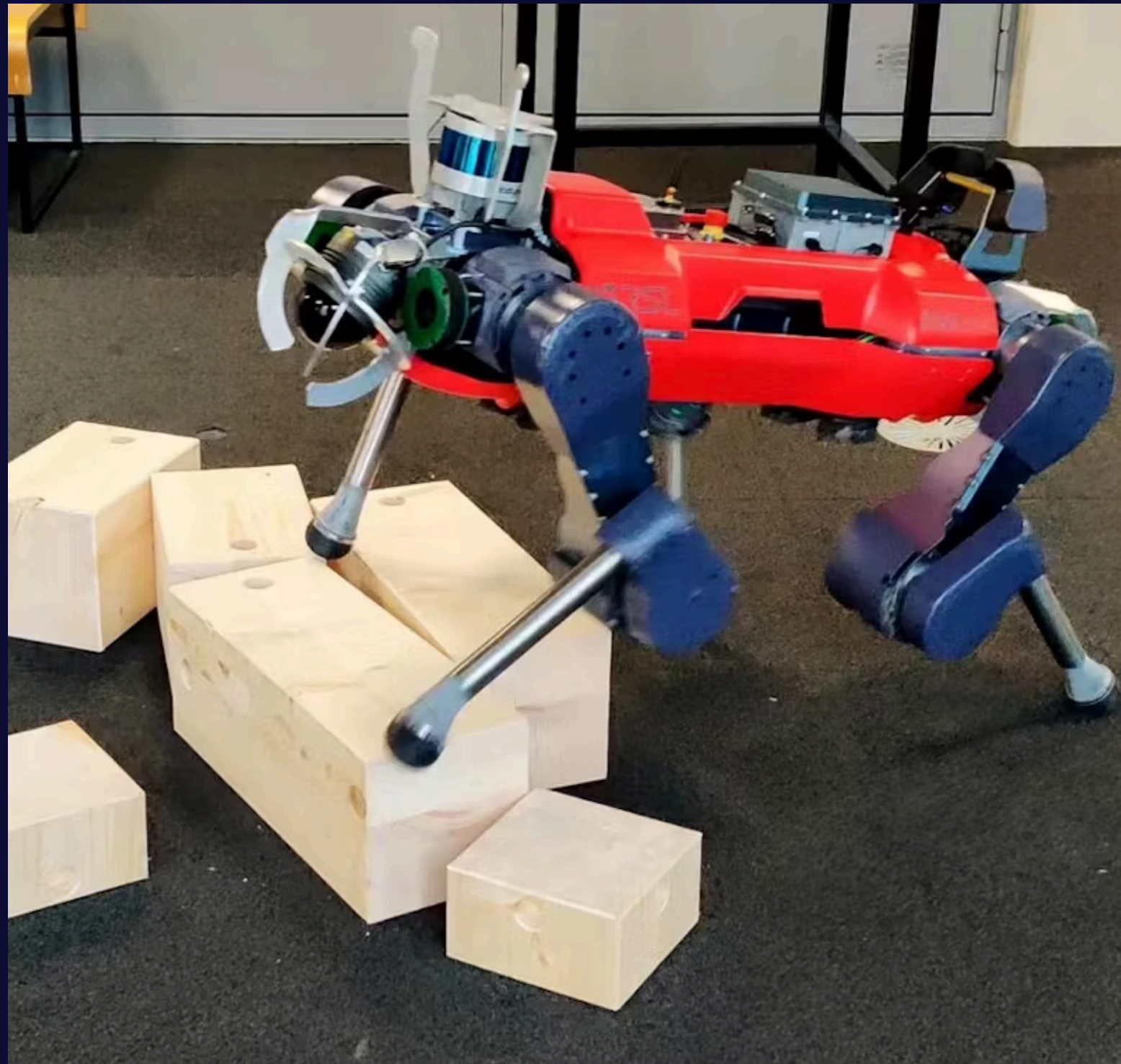
How to test your MPC output?

Optimal Control Modules

The optimal control formulation

OCS2 is tailored to solve optimal control problems for switched systems. A switched system consists of a finite number of dynamical subsystems subjected to discrete events which cause transition between these subsystems. Switched system models are encountered in many practical applications such as automobiles and locomotives with different gears, DC-DC converters, manufacturing processes, biological systems, and walking robotics. The optimal control for such a system can be formulated as:

$$\begin{cases} \min_{\mathbf{u}(\cdot)} \sum_i \phi_i(\mathbf{x}(t_{i+1})) + \int_{t_i}^{t_{i+1}} l_i(\mathbf{x}(t), \mathbf{u}(t), t) dt \\ \text{s.t. } \mathbf{x}(t_0) = \mathbf{x}_0 & \text{initial state} \\ \dot{\mathbf{x}}(t) = \mathbf{f}_i(\mathbf{x}(t), \mathbf{u}(t), t) & \text{system flow map} \\ \mathbf{x}(t_{i+1}^+) = \mathbf{j}(\mathbf{x}(t_{i+1})) & \text{system jump map} \\ \mathbf{g}_{1i}(\mathbf{x}(t), \mathbf{u}(t), t) = \mathbf{0} & \text{state-input equality constraints} \\ \mathbf{g}_{2i}(\mathbf{x}(t), t) = \mathbf{0} & \text{state-only equality constraints} \\ \mathbf{h}_i(\mathbf{x}(t), \mathbf{u}(t), t) \geq \mathbf{0} & \text{inequality constraints} \\ \text{for } t_i < t < t_{i+1} \text{ and } i \in \{0, 1, \dots, I-1\} \end{cases}$$



Importancia de este modelado

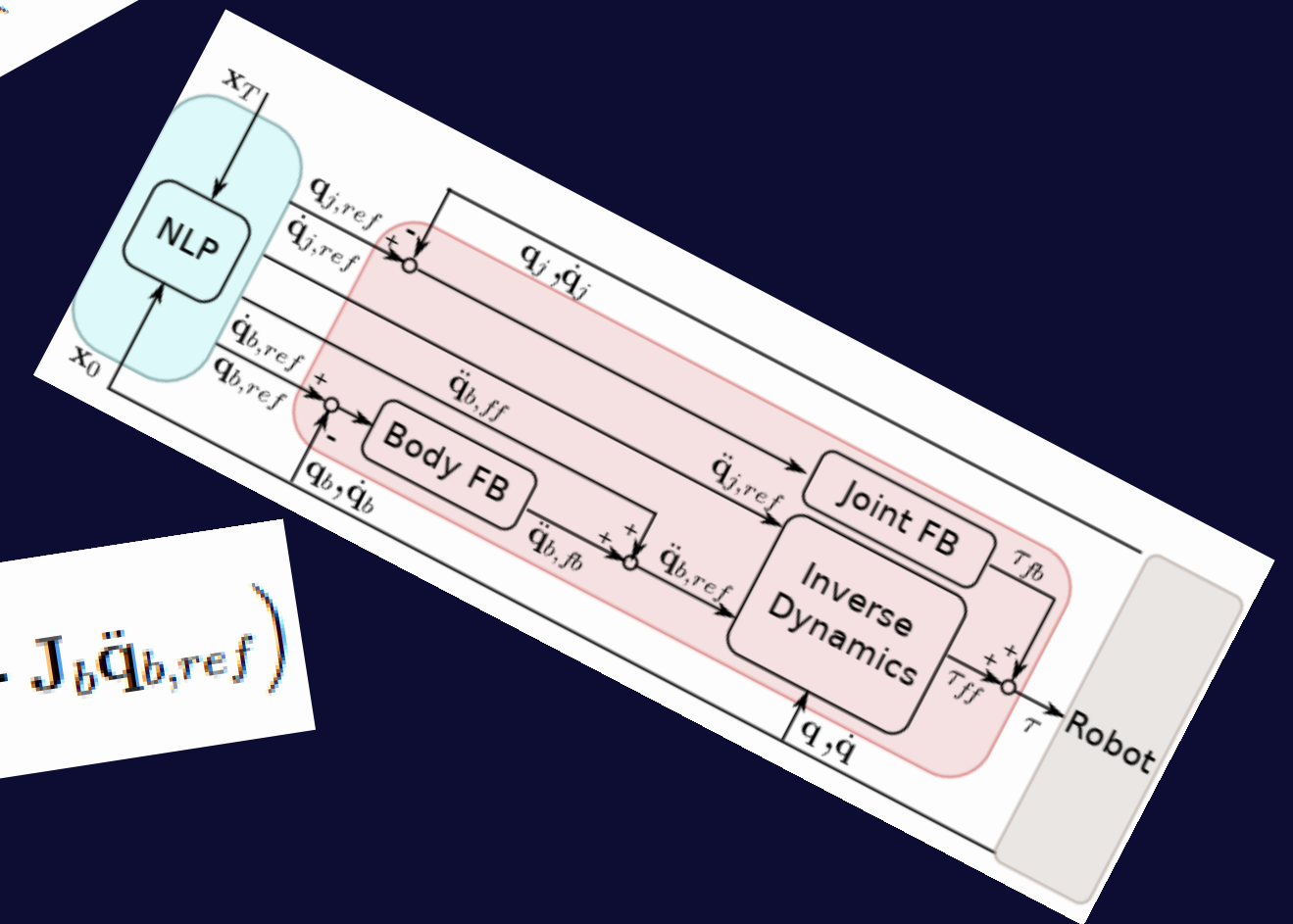
- Los robots de patas tienen ventajas en terrenos donde los sistemas con ruedas no son eficientes.
- Pueden desplazarse por superficies irregulares, subir escaleras y sortear obstáculos.
- Sin embargo, su modelado es más complejo debido a la dinámica no lineal y la necesidad de control preciso.

Problemática a tratar

- El modelado de estos robots puede ser difícil de entender para personas con poca experiencia en robótica.
- Es necesario un enfoque claro y accesible para facilitar su comprensión.

$$\mathbf{p}_c(t) = \sum_{f=1}^{n_f} \sum_{v=1}^{n_v} \lambda_v^f(t) (\mathbf{p}^f(t) + \mathbf{R}(\alpha^f(t)) \mathbf{v}_v)$$

$$\ddot{\mathbf{q}}_{j,ref} = \mathbf{J}_j^+ (\ddot{\mathbf{p}} - \dot{\mathbf{J}}\dot{\mathbf{q}} - \mathbf{J}_b\ddot{\mathbf{q}}_{b,ref})$$



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

$$\mathbf{w}_r = [\mathbf{a}_{1,0}, \dots, \mathbf{a}_{1,4}, \dots, \mathbf{a}_{n,0}, \dots, \mathbf{a}_{n,4}]$$

Solución

01

- Explicación detallada pero accesible el modelado de Legged robots, basado en la tesis Optimization-based motion planning for legged robots.

02

- Se desglosan conceptos clave y herramientas matemáticas utilizadas en la planificación del movimiento facilitando la comprensión y aplicación del modelado para quienes deseen desarrollar robots de patas.

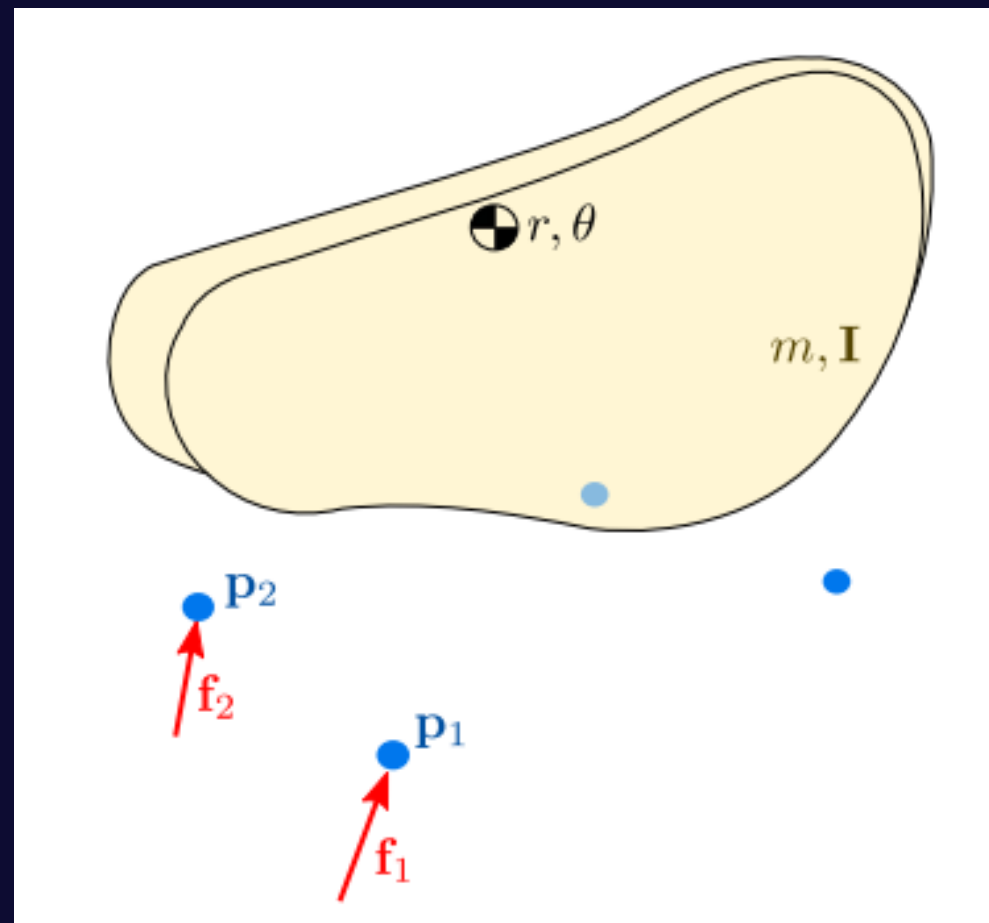
- ¿Qué significa?:

- ¿Por qué es útil?:

2.1 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: $p_{c,x} = 0.0 \text{ m}$.



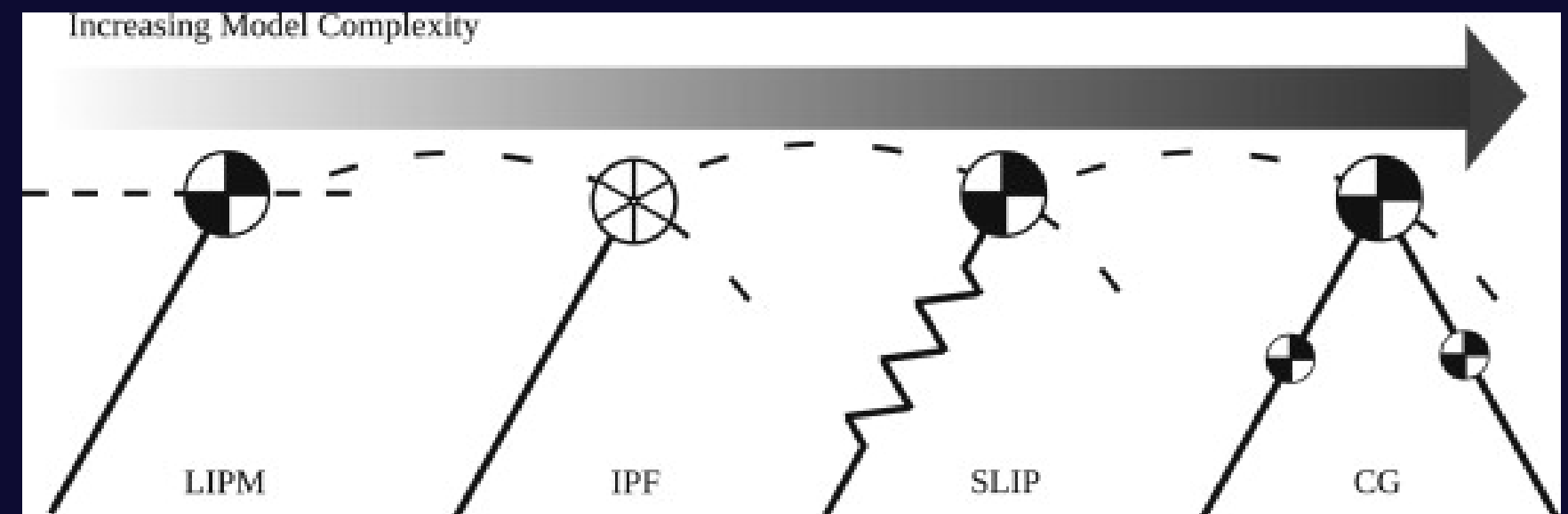
02

Dinámica de Cuerpo Rígido Simplificada (SRBD): Modelo que trata el robot como un solo cuerpo.

Modelos fundamentales

01

Modelo del Péndulo Invertido Lineal (LIPM): simplificada para analizar el equilibrio y la planificación de trayectorias.



Simultaneous foothold and body optimization

01

Se integra la planificación del movimiento del cuerpo y la determinación de los puntos de apoyo en una única formulación de optimización

02

Aunque la inclusión simultánea de ambas variables hace que el problema sea no lineal, se demuestra que su solución puede tomar solamente milisegundos.

ETH zürich

Online Walking Motion and Contact Optimization for Quadruped Robots

Alexander W. Winkler, Farbod Farshidian, Michael Neunert, Diego Pardo, Jonas Buchli

ICRA 2017, Singapore



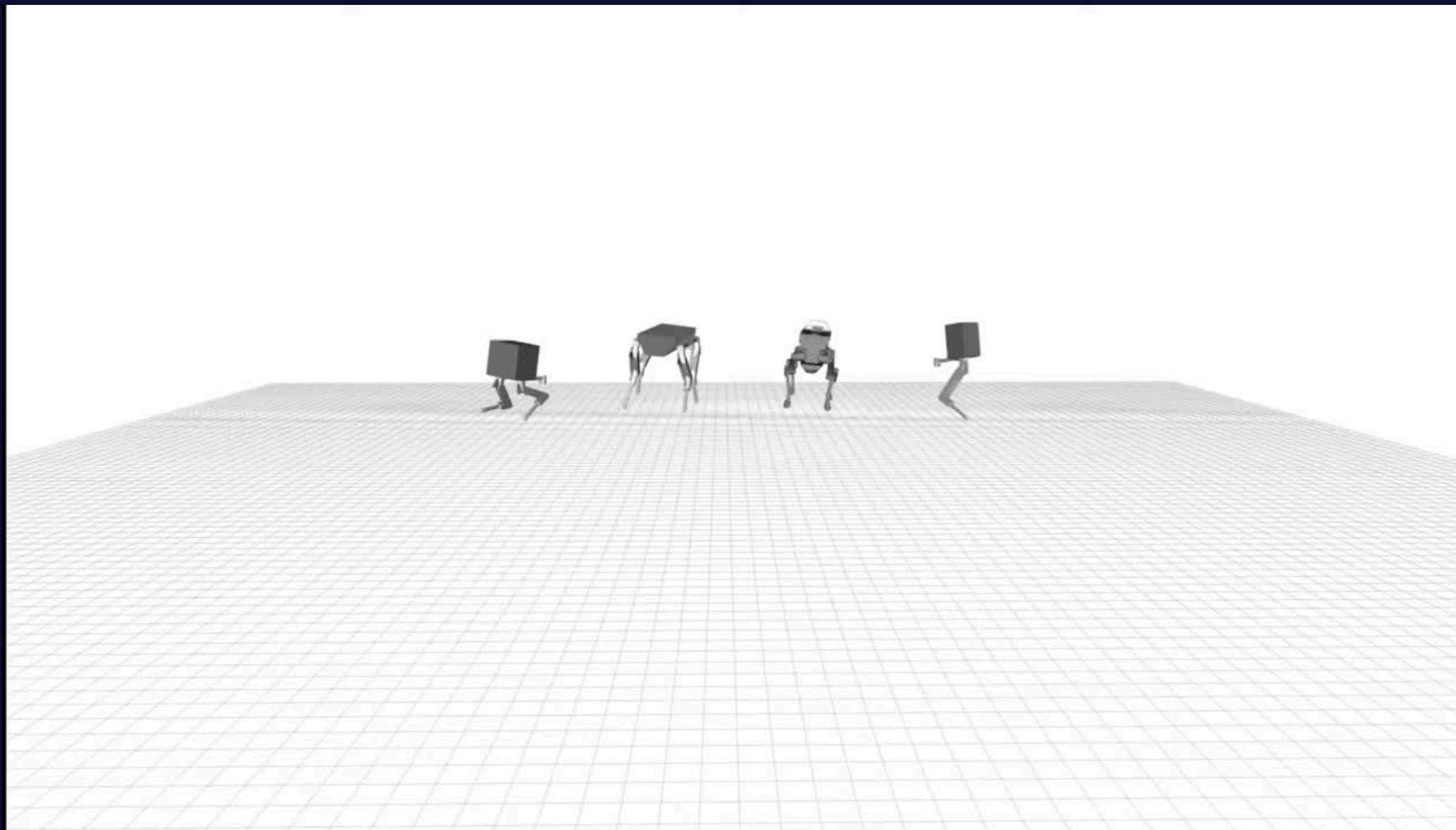
ADRL Agile and Dexterous Robotics Lab

Vertex-based ZMP constraints



- Se introduce un metodo que modela los contactos no solo como areas (polígonos) sino que también abarca puntos y líneas de contacto mediante una representación basada en vértices.
- Se pueden generar de forma eficiente movimientos como caminar, trotar, saltar, caminar lentamente,

Gait and trajectory optimization



- 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).
- En lugar de resumir toda la interacción entre el robot y el suelo en un único punto, se evalúan las fuerzas que actúan en cada pie de manera completa, considerando sus componentes verticales y laterales.