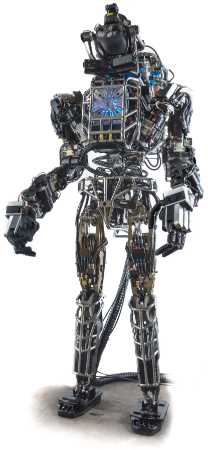
Underactuated Robots
Lecture 5: Legged Robot Models

Nicola Scianca

December 2024

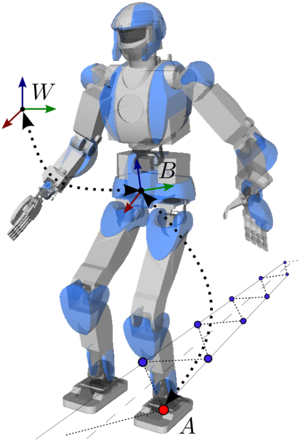
outline

- full dynamic model
- Newton-Euler equations
- centroidal dynamics
- single-rigid-body dynamics
- zero moment point
- linear inverted pendulum
- divergent component of motion
- 3D linear inverted pendulum



full dynamic model

- we know how to express the dynamics of a manipulator using **Lagrangian dynamics**
- we can do the same for a legged robot, but we must carefully choose the configuration variables
- the **joint configuration** is **not sufficient**! we must also express the position and orientation of the robot in the world



full dynamic model

- the **floating base** is a particular robot link (usually the torso), whose coordinates $(p_0, r_0) \in SE(3)$ represent the absolute position and orientation of the robot

- typically, this is represented as a **position vector**, and a **quaternion** for the orientation

$$p_0 = (x_0, y_0, z_0), \quad r_0 = a_0 + b_0\mathrm{i} + c_0\mathrm{j} + d_0\mathrm{k}$$

- the floating base linear and angular **velocities** are

$$\dot{p}_0 = (\dot{x}_0, \dot{y}_0, \dot{z}_0), \quad \omega_0 = (\omega_0^x, \omega_0^y, \omega_0^z)$$

full dynamic model

- the full dynamic model can be expressed in the familiar form

$$M(q)\ddot{q} + n(q, \dot{q}) = S\tau + \sum_i J_i^T f_i$$

- $M(q)$ is the mass matrix
- $n(q, \dot{q})$ collects the Coriolis/centrifugal and gravity terms
- S maps torques to coordinates. in particular, the lines corresponding to the floating base coordinates are zero because the base is not actuated
- $J_i^T f_i$ is the effect of the i-th contact force, i.e., ground reaction force on one foot

partial and simplified models

- the full dynamic model is very complex and nonlinear
- sometimes, it is better to opt for a simpler model, that captures only the essential aspects of the dynamics
- the simplified models can be derived from the full model, but it is much easier to start from scratch, by writing the **Newton-Euler equations**

Newton-Euler equations

- the **Newton-Euler equations** describe the dynamics of the robot as a whole, in terms of balance of forces and momenta
- **force balance**: the sum of all forces is equal to the acceleration of the **Center of Mass** (CoM) p_c

$$m\ddot{p}_c = mg + \sum_i f_i$$

- **moment balance**: the sum of the moment of each force is equal to the derivative of the **angular momentum** L_c

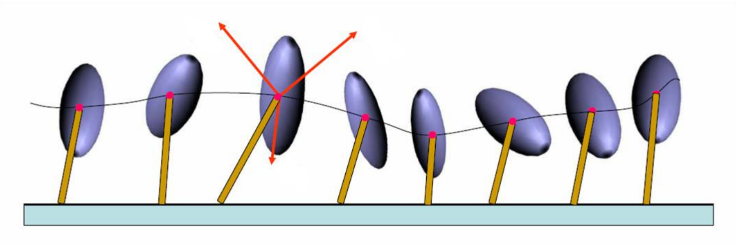
$$\dot{L}_c = \cancel{(p_c - p_c) \times g} + \sum_i (p_i - p_c) \times f_i$$

centroidal dynamics

- the Newton-Euler equations describe the **centroidal dynamics** of the robot

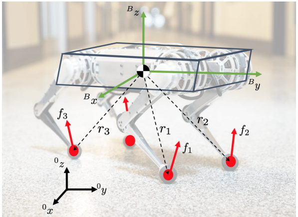
$$m\ddot{p}_c = mg + \sum_i f_i, \qquad \dot{L}_c = \sum_i (p_i - p_c) \times f_i$$

- they can be seen as the dynamics of a variable inertia **ellipsoid** around the CoM



single-rigid-body dynamics

- the angular momentum L_c is a complicated object because the internal configuration of a robot is changing
- one way of simplifying the model is to assume that the angular momentum comes from the motion of a **rigid body**
- the orientation of this rigid body could be mapped to the orientation of the torso, or the entire robot upper body



single-rigid-body dynamics

- velocity of a rigid body: **linear** velocity of its CoM and **angular** velocity (expressed in the **local frame**)

$$\dot{p}_c = (\dot{p}_c^x, \dot{p}_c^y, \dot{p}_c^z), \qquad \omega = (\omega^x, \omega^y, \omega^z),$$

- the variation of angular momentum of a rigid body is

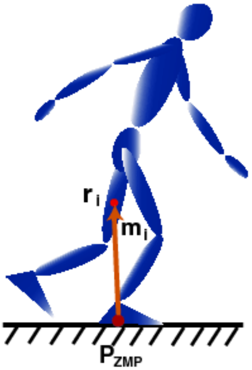
$$L_c = I\dot{\omega} + \omega \times (I\omega)$$

- then rotational part of the dynamics can be written as

$$I\dot{\omega} + \omega \times (I\omega) = \sum_i (p_i - p_c) \times f_i$$

zero-moment point

- the **zero-moment point** (ZMP) is an alternative way of encoding information on the contact forces
- it represents the point of application of the resultant **ground reaction force** (GRF)
- in statics, we can tell if a body is balanced by checking if the ground projection of the CoM is inside the **base of support**, in dynamics, we do something similar with the ZMP



zero-moment point

- by definition, the sum of the moments of contact forces with respect to the ZMP p_z is zero

$$\sum_i (p_i - p_z) \times f_i = 0$$

- let's assume that p_z, as well as all contact points p_i, are on **flat horizontal ground**

- also, let's define the ZMP to be on the ground, which means that $p_z^z = 0$

zero-moment point

- let's compute the vector product

$$\sum_i \begin{pmatrix} 0 & -(p_i^z - p_z^z) & p_i^y - p_z^y \\ p_i^z - p_z^z & 0 & -(p_i^x - p_z^x) \\ -(p_i^y - p_z^y) & p_i^x - p_z^x & 0 \end{pmatrix} \begin{pmatrix} f_i^x \\ f_i^y \\ f_i^z \end{pmatrix} = 0$$

$$\sum_i \begin{pmatrix} (p_i^y - p_z^y) f_i^z - (p_i^z - \cancel{p_z^z}) f_i^y \\ (p_i^z - \cancel{p_z^z}) f_i^x - (p_i^x - p_z^x) f_i^z \\ (p_i^x - p_z^x) f_i^y - (p_i^y - p_z^y) f_i^x \end{pmatrix} = 0$$

- the first two equations are

$$\sum_i (p_i^y - p_z^y) f_i^z - p_i^z f_i^y = 0$$

$$\sum_i (p_i^x - p_z^x) f_i^z - p_i^z f_i^x = 0$$

zero-moment point

- these two equations we can write them compactly as

$$p_z^{x,y} \sum_i f_i^z = \sum_i p_i^{x,y} f_i^z$$

$$p_z^{x,y} = \frac{\sum_i p_i^{x,y} f_i^z}{\sum_i f_i^z}$$

- this is the **position of the ZMP** on flat ground

zero-moment point

- if we denote the total vertical force as $f_z = \sum_i f_i^z$, we can write the position of the ZMP as

$$\boxed{p_z^{x,y} = \sum_i p_i^{x,y} \frac{f_i^z}{f_z}}$$

this is a **weighted sum** of the position of the contact points

- the coefficients f_i / f_z are positive because forces point up $\uparrow$

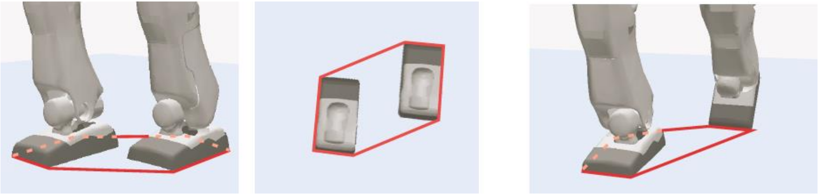
$$\frac{f_i^z}{f_z} \geq 0, \qquad \sum_i \frac{f_i^z}{f_z} = 1$$

- thus, the ZMP position is a **convex combination** of the contact points

zero-moment point

- because the contact forces are **unidirectional** (they only point up), the ZMP must be inside the **convex hull** of the contact surface

- this region is called the **support polygon**



ZMP dynamics

- let's go back to the **moment balance** equation

$$\dot{L}_c = \sum_i (p_i - p_c) \times f_i$$

- add and subtract this term to make the ZMP appear

$$\begin{aligned}\dot{L}_c &= \sum_i (p_i - p_c) \times f_i + \sum_i (p_c - p_z) \times f_i - \sum_i (p_c - p_z) \times f_i \\ &= \cancel{\sum_i (p_i - p_z) \times f_i} - (p_c - p_z) \times \sum_i f_i\end{aligned}$$

- the first term is zero because of the definition of ZMP, in the second term we recover the sum of contact forces

ZMP dynamics

- the sum of contact forces is given by the first of the Newton-Euler equations $m\ddot{p}_c = mg + \sum_i f_i$

$$\dot{L}_c = -m(p_c - p_z) \times (\ddot{p}_c - g)$$

- in particular, we are interested in the x and y components of this equation

$$\dot{L}_c^x = -m(p_c^y - p_z^y)(\ddot{p}_c^z - g^z) + m(p_c^z - \cancel{p_z^z})(\ddot{p}_c^y - \cancel{g^y})$$
$$\dot{L}_c^y = m(p_c^x - p_z^x)(\ddot{p}_c^z - g^z) - m(p_c^z - \cancel{p_z^z})(\ddot{p}_c^x - \cancel{g^x})$$

ZMP dynamics

- the dynamics of the CoM can be expressed in terms of the position of the ZMP (g^z is $-g$, because it is pointing down)

$$\ddot{p}_c^y = \frac{\ddot{p}_c^z + g}{p_c^z}(p_c^y - p_z^y) + \frac{\dot{L}_c^x}{m p_c^z}$$
$$\ddot{p}_c^x = \frac{\ddot{p}_c^z + g}{p_c^z}(p_c^x - p_z^x) - \frac{\dot{L}_c^y}{m p_c^z}$$

- this can be written compactly as

$$\ddot{p}_c^{x,y} = \frac{\ddot{p}_c^z + g}{p_c^z}(p_c^{x,y} - p_z^{x,y}) + R\frac{\dot{L}_c^{x,y}}{m p_c^z}, \qquad R = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

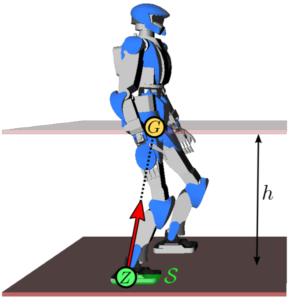
where R is a $\pi/2$ rotation matrix

linear inverted pendulum

- two simplifying assumptions:
  CoM height is constant $z_c = h$;
  internal **angular momentum** derivative is zero $\dot{L}_c = 0$

- the ZMP-CoM dynamics becomes

$$\ddot{p}_c^{x,y} = \overbrace{\cancel{\frac{\ddot{p}_c^z + g}{p_c^z}}}^{g/h}(p_c^{x,y} - p_z^{x,y}) + R\cancel{\frac{\dot{L}_c^{x,y}}{mp_c^z}}$$



linear inverted pendulum

- the **linear inverted pendulum** (LIP) dynamics is

$$\boxed{\ddot{p}_c^{x,y} = \eta^2 (p_c^{x,y} - p_z^{x,y})} \qquad \eta = \sqrt{\frac{g}{h}}$$

- significance: the ZMP **pushes away** the CoM

- it is an **unstable dynamics**: this makes sense because it represents the essence of the dynamics of balancing

linear inverted pendulum

- the LIP model is decoupled, so let's just drop the ^{x,y}, and work with a generic component p that could be either x or y

$$\ddot{p}_c = \eta^2 (p_c - p_z)$$

- to write this model in state-space form, we must identify its states, inputs and outputs: one possible choice is

$$x = \begin{pmatrix} p_c \\ \dot{p}_c \end{pmatrix}, \quad u = p_z, \quad y = p_c$$

- which gives us the following system

$$\frac{d}{dt} \begin{pmatrix} p_c \\ \dot{p}_c \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ \eta^2 & 0 \end{pmatrix} \begin{pmatrix} p_c \\ \dot{p}_c \end{pmatrix} + \begin{pmatrix} 0 \\ -\eta^2 \end{pmatrix} p_z$$

divergent component of motion

- the LIP has two eigenvalues: $\pm\eta$

- we can perform a **change of coordinates** to decouple the dynamics associated with these eigenvalues

$$\begin{pmatrix} p_s \\ p_u \end{pmatrix} = \begin{pmatrix} 1 & -1/\eta \\ 1 & 1/\eta \end{pmatrix} \begin{pmatrix} p_c \\ \dot{p}_c \end{pmatrix}$$

- we call p_s the one that will be associated with $-\eta$ (stable) and p_u the one that will be associated the $+\eta$ (unstable)

divergent component of motion

- to perform this change of coordinates, we first have to identify the inverse transformation

$$\begin{pmatrix} p_c \\ \dot{p}_c \end{pmatrix} = \begin{pmatrix} 1 & -1/\eta \\ 1 & 1/\eta \end{pmatrix}^{-1} \begin{pmatrix} p_s \\ p_u \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1/\eta & 1/\eta \\ -1 & 1 \end{pmatrix} \begin{pmatrix} p_s \\ p_u \end{pmatrix}$$

- we can substitute it inside the state-space dynamics

$$\frac{1}{2} \begin{pmatrix} 1/\eta & 1/\eta \\ -1 & 1 \end{pmatrix} \frac{d}{dt} \begin{pmatrix} p_s \\ p_u \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ \eta^2 & 0 \end{pmatrix} \frac{1}{2} \begin{pmatrix} 1/\eta & 1/\eta \\ -1 & 1 \end{pmatrix} \begin{pmatrix} p_s \\ p_u \end{pmatrix} + \begin{pmatrix} 0 \\ -\eta^2 \end{pmatrix} p_z$$

$$\frac{d}{dt} \begin{pmatrix} p_s \\ p_u \end{pmatrix} = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} p_s \\ p_u \end{pmatrix} + \begin{pmatrix} \eta \\ -\eta \end{pmatrix} p_z$$

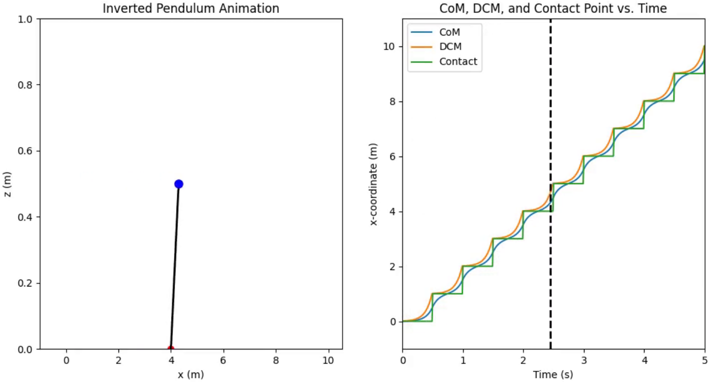
divergent component of motion

- the decoupled dynamics are

$$\dot{p}_s = (-p_s + p_z)$$
$$\dot{p}_u = (p_u - p_z)$$

- the stable dynamics p_s is **attracted** to the ZMP
- the unstable dynamics p_u is **pushed away** from the ZMP
- in the literature, the latter is often called the **Divergent Component of Motion** (DCM): many methods focus on making sure the DCM doesn't diverge

locomotion example with the LIP



3D ZMP dynamics

- there are other ways to approximate the CoM-ZMP dynamics that don't require the CoM to be on a horizontal plane

- the first thing we have to do is to go back to the moment equations and not impose $p_z^z = 0$

$$\dot{L}_c^x = -m(p_c^y - p_z^y)(\ddot{p}_c^z - g^z) + m(p_c^z - p_z^z)(\ddot{p}_c^y - \cancel{g^y})$$
$$\dot{L}_c^y = m(p_c^x - p_z^x)(\ddot{p}_c^z - g^z) - m(p_c^z - p_z^z)(\ddot{p}_c^x - \cancel{g^x})$$

- we get

$$\ddot{p}_c^{x,y} = \frac{\ddot{p}_c^z + g}{p_c^z - p_z^z}(p_c^{x,y} - p_z^{x,y})$$

3D ZMP dynamics

- to get the LIP, we set z equal to a constant height; what happens if we just force the coefficient of $p_c^{x,y} - p_z^{x,y}$ to be equal to a constant?

$$\frac{\ddot{p}_c^z + g}{p_c^z - p_z^z} = a$$

- the horizontal components of the dynamics now look just like the LIP, but we set the pendulum frequency arbitrarily to be some constant a

$$\ddot{p}_c^{x,y} = a(p_c^{x,y} - p_z^{x,y})$$

- we get a third equation that looks a lot like the horizontal LIP dynamics, except that there is an additional drift term $-g$

$$\ddot{p}_c^z = a(p_c^z - p_z^z) - g$$

3D ZMP dynamics

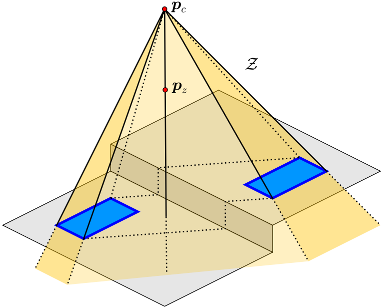
- we can write compactly this 3D model as

$$\ddot{p}_c = a(p_c - p_z) + \overrightarrow{g}$$

- this means that we can represent 3D CoM-ZMP dynamics using a linear model!
- however, for the ZMP on flat ground, we just had to make sure that it's inside the support polygon

3D ZMP dynamics

- in order to guarantee that contact forces are feasible, the 3D ZMP must be inside a **pyramidal region**, with the CoM as its vertex



- this is slightly annoying, because this condition is **nonlinear** (but we can approximate it with a linear condition)