Autonomous and Mobile Robotics

Prof. Giuseppe Oriolo

Humanoid Robots 1:

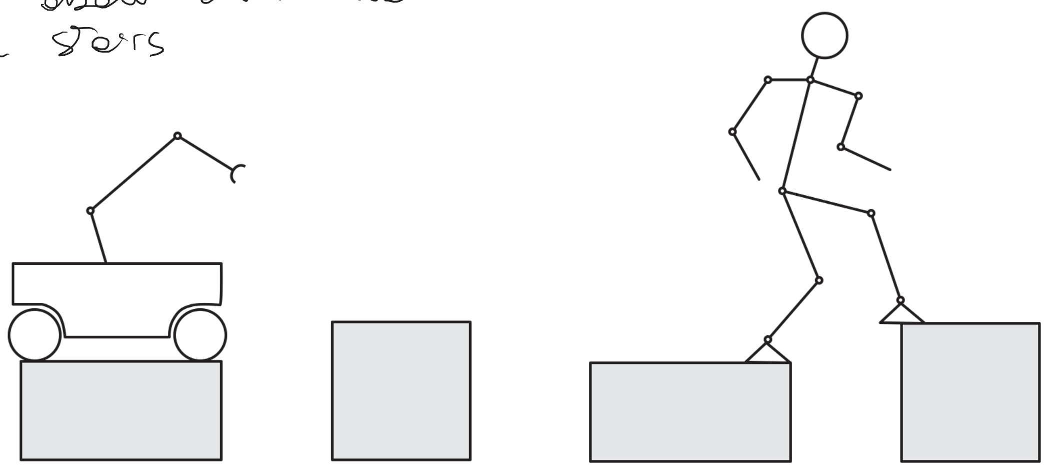
Introduction

DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI
SAPIENZA
UNIVERSITÀ DI ROMA

why humanoids

- practical reasons:
  in many cases humanoids are the most sensible choice

think about environment
with stairs



- psychological and commercial reasons:
  humanoids have a major appeal

why humanoids

- multipurpose: sensing, manipulation, locomotion etc…
- adaptability: humanoids can work in environments suitable for humans and expand their capabilities by using machines designed for humans
- collaboration: humanoid motion is easy for humans to understand and predict
- human-like appearance: empathy

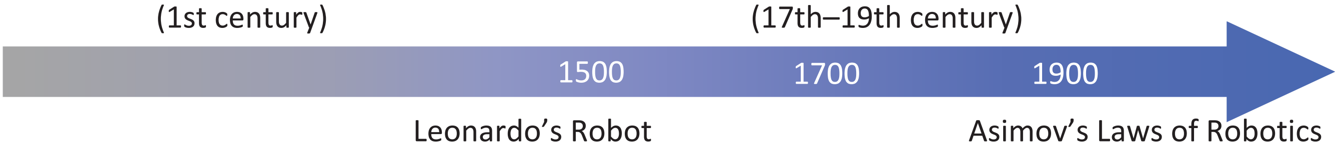
some history

- pre-research period: humans always fascinated by the idea of building anthropomorphic machines
- pioneering period (1970s-1990s): initial research on biped prototypes
- new millennium: industrial companies showed that building actual humanoids was possible
- today: research focusing on humanoid robustness, efficiency and versatility

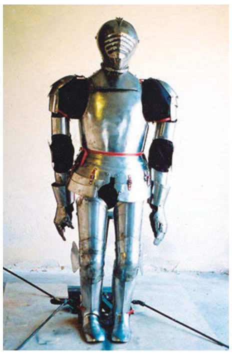
pre-research period

Hero’s Automata
(1st century)

Karakuri Dolls
(17th–19th century)



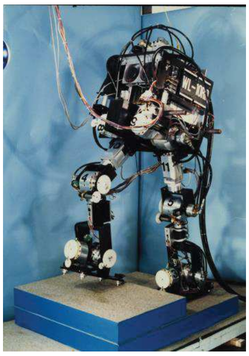
Leonardo’s Robot
(1495)



Asimov’s Laws of Robotics
(1942)

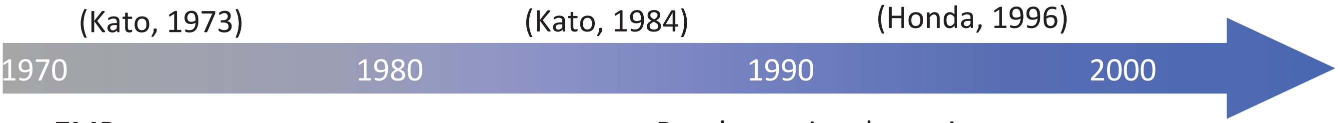
pioneering period

WABOT-1
(Kato, 1973)



WL-10RD
(Kato, 1984)

P2
(Honda, 1996)

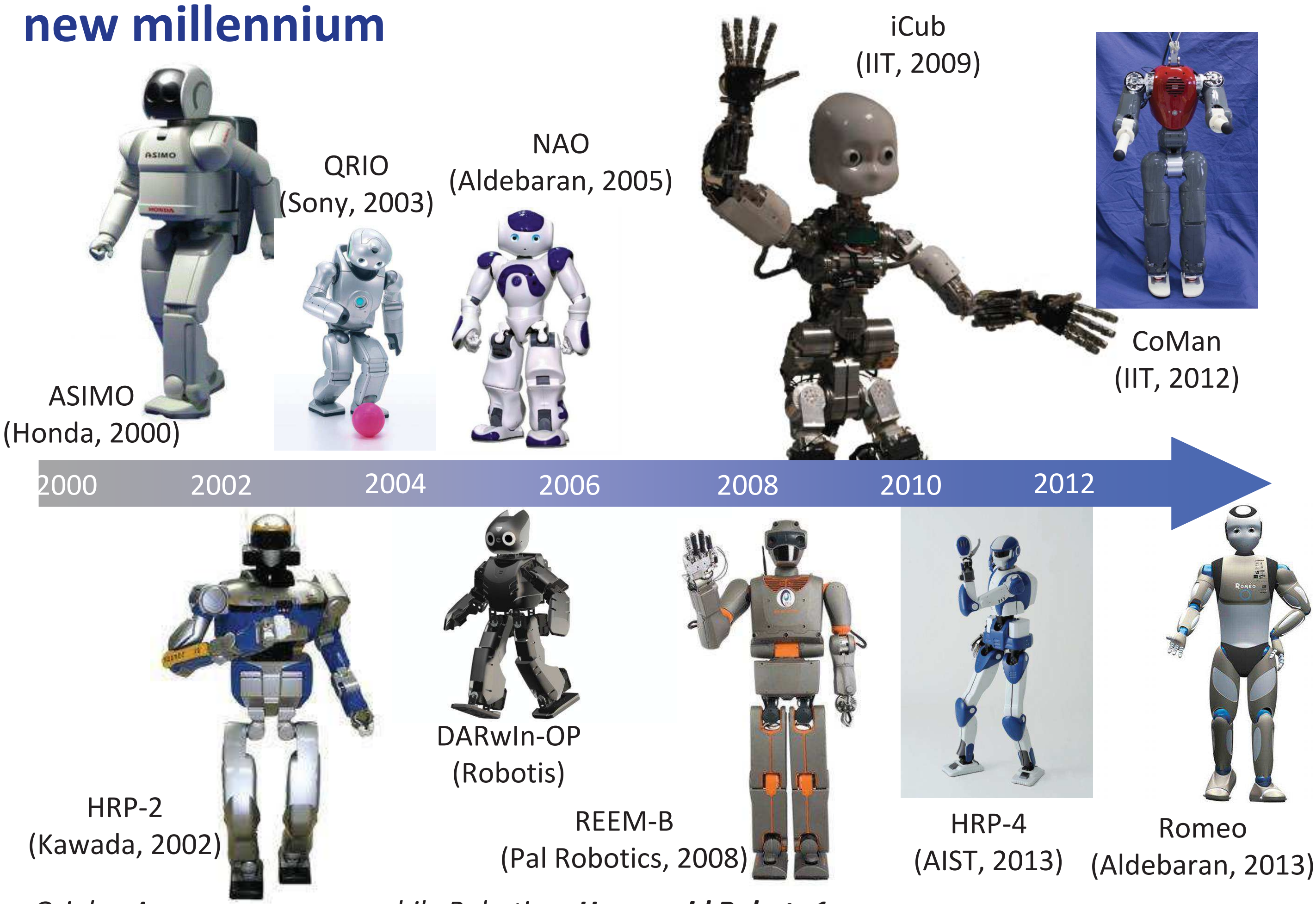


ZMP concept
(Vukobratović, 1972)

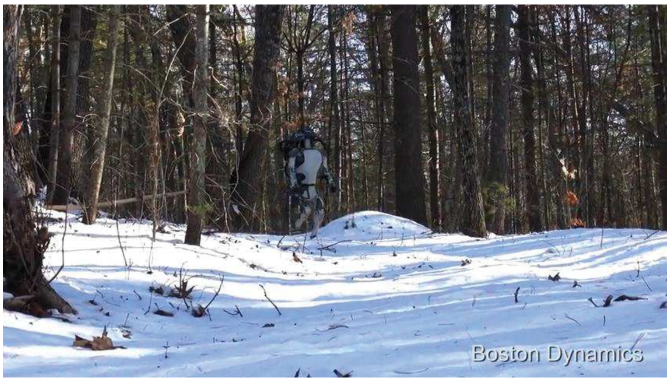
First computer-controlled robot
(Raibert, 70s-80s)

Purely passive dynamics
(McGeer, 1990)

new millennium

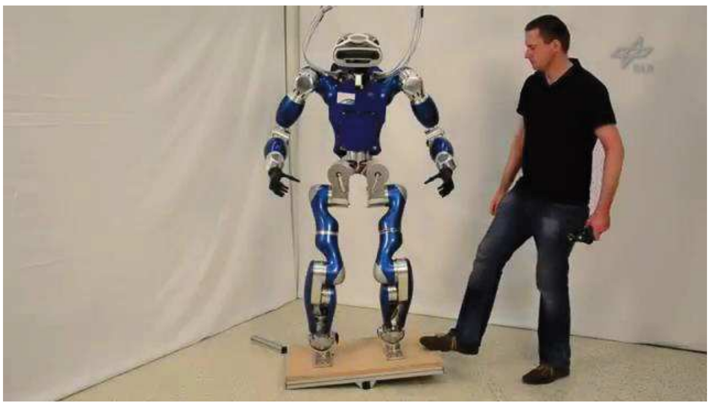


today



ATLAS
(Boston Dynamics)

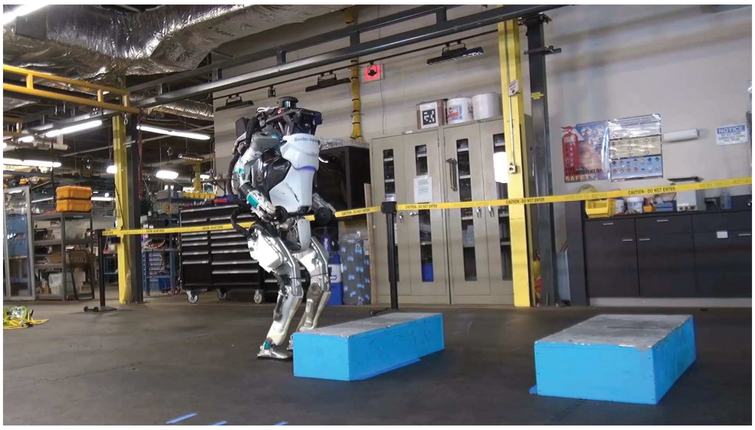
TORO
(DLR)



not only walking

running

jumping

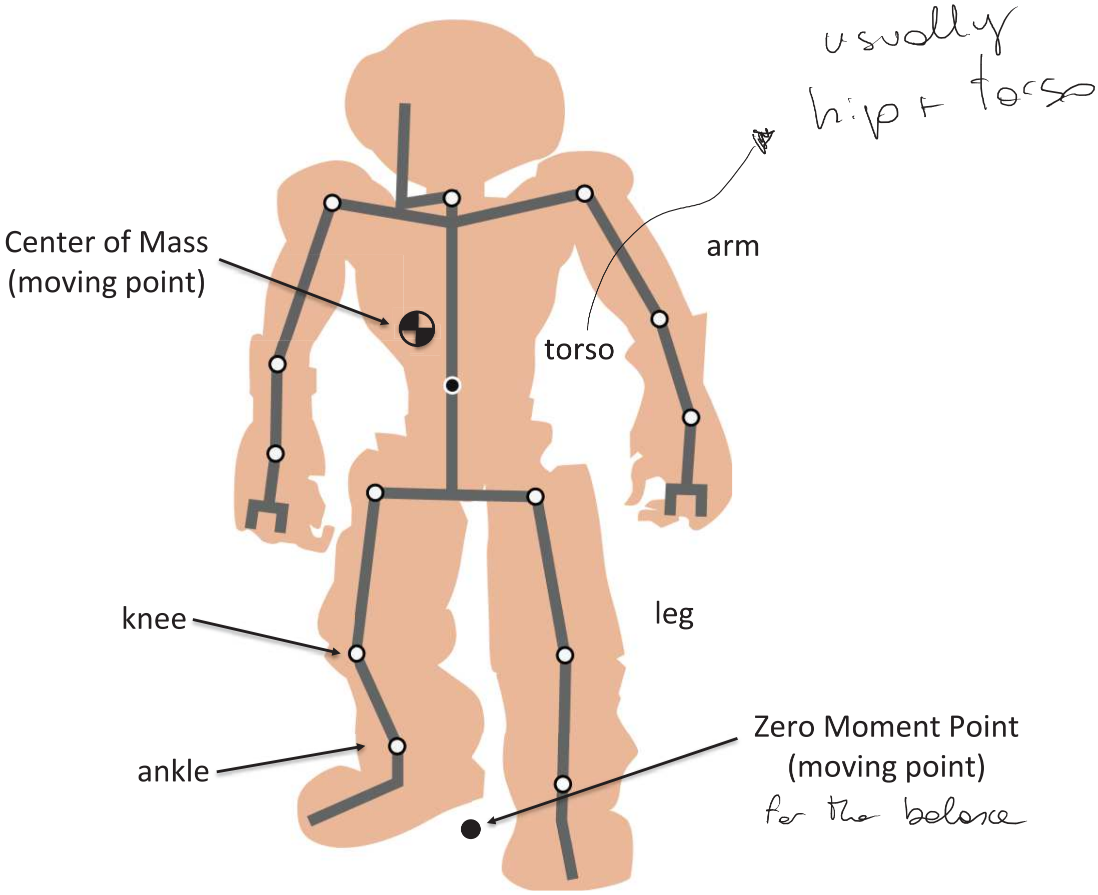


whole-body control

ASIMO

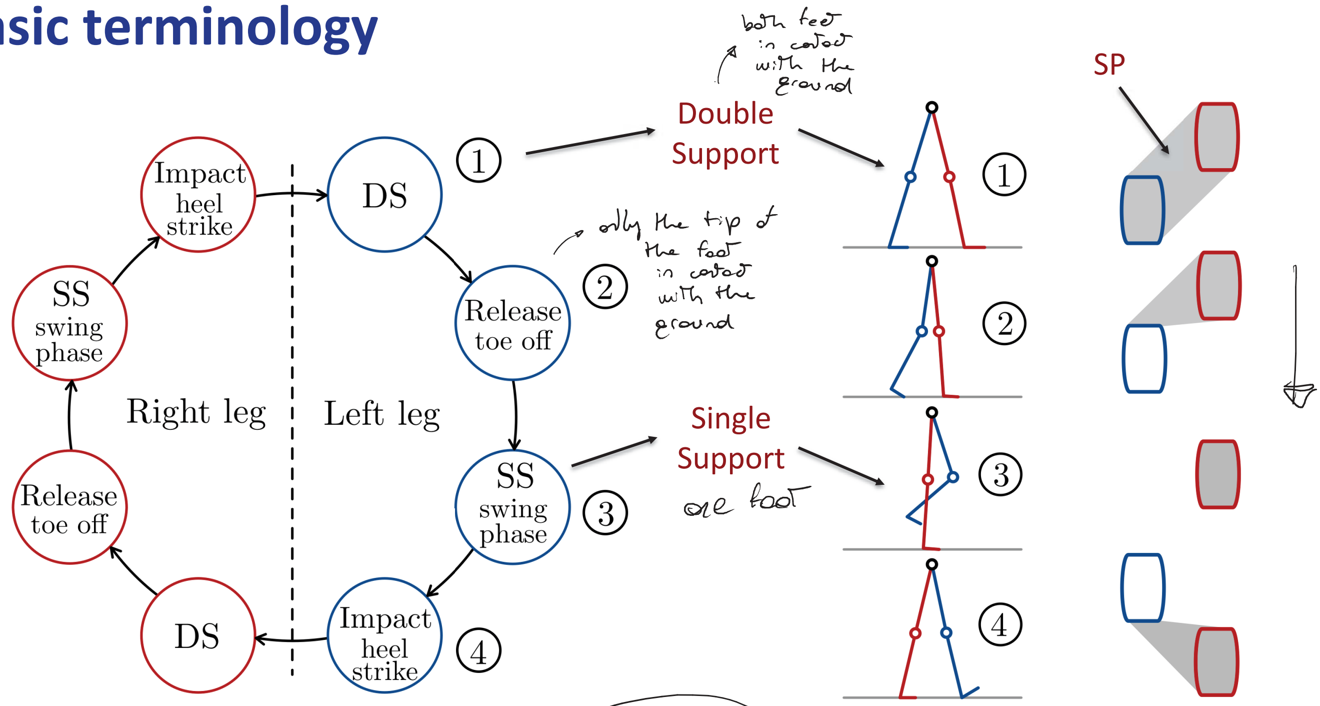
basic terminology

- Anatomy



human walking: analysis

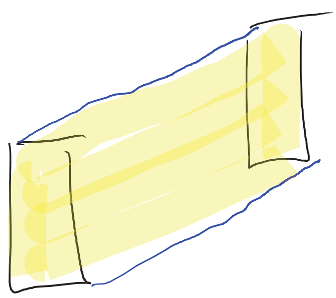
basic terminology

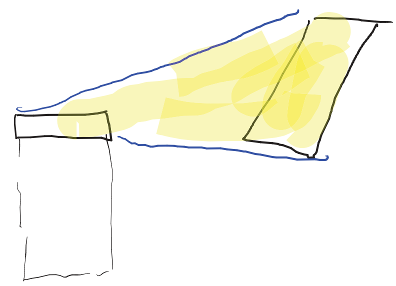


- walking: cyclic alternation of 4 phases
- Support Polygon (SP): convex hull of the contact points with the ground (support points)
- robots with flat feet have only Single and Double Support phases

Convex Hull

feet in contact with the ground





Smallest convex region which includes both contact surfaces

Def: The convex hull of X is the smallest convex set that contains X

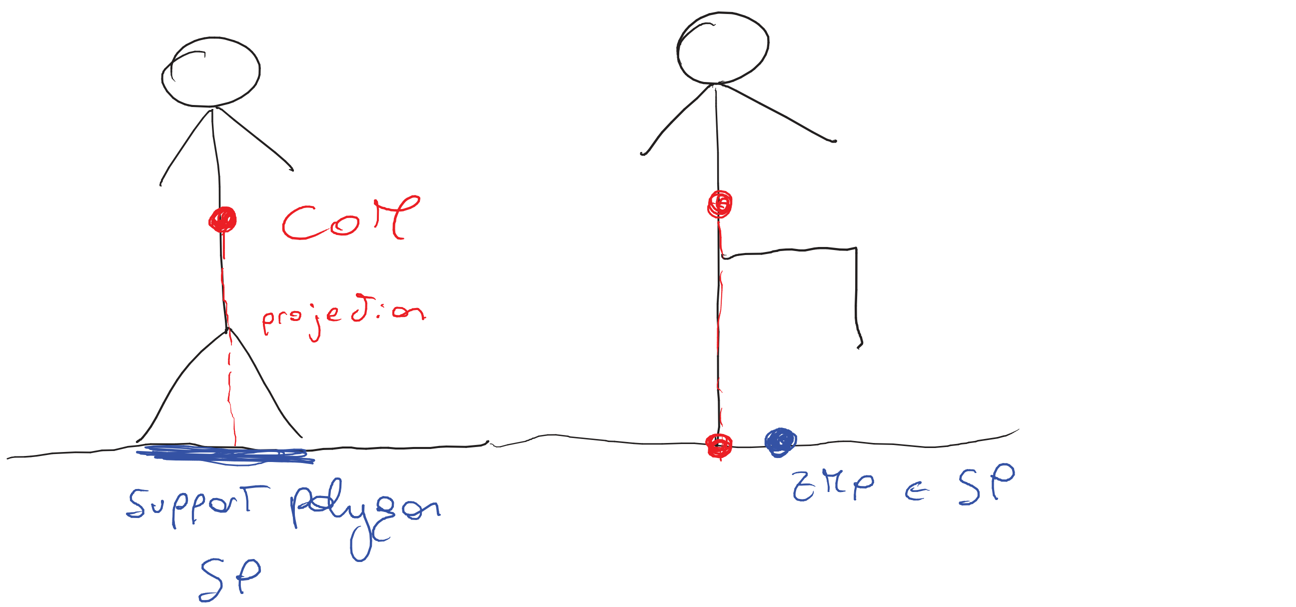
gaits → (IT Camminata)

- static(ally stable) gait: the projection of the CoM on the ground is always inside the SP
- however, static gaits are very slow and conservative
- Zero Moment Point (ZMP): point on the ground where the resultant of the reaction forces acts (more on this later)
- dynamic(ally stable) gait: the ZMP is always inside the SP

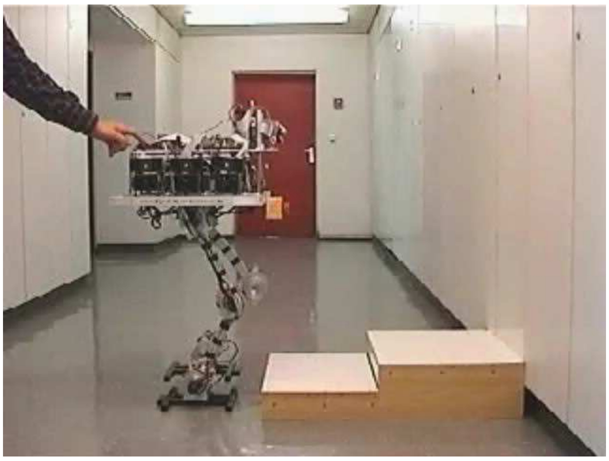
x-axis direction: longitudinal
↳ to the direction of the gaits

y-axis direction: lateral

STATIC BALANCING DYNAMIC WALKING



gaits



static walk

very very slow
(shift of the CoM inside the SP)

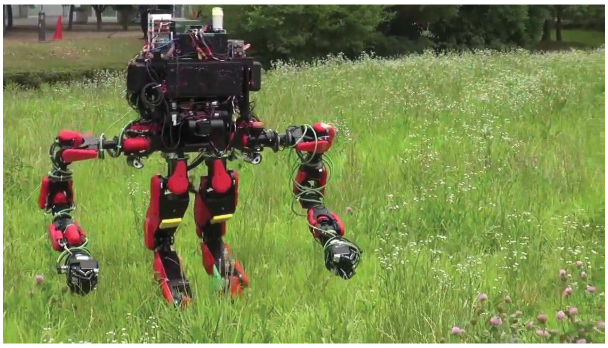
dynamic walk

faster
control the ZMP
(harder than controlling the CoM)

passive (dynamic) walkers

- energy-efficient, natural gait (limit cycle)
- does not work on horizontal ground
- limited agility and responsiveness of motion

active (dynamic) walkers



- actuated joints (energy consumption)
- feedback control needed
- robots with flat feet or non-trivial feet

Resume

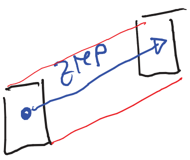
- Balancing (dynamic balancing in particular)

  CoM $\longleftrightarrow$ ZMP

  ZMP $\in$ support polygon

- Phases of walking

  single support phase : 1 foot in contact with the ground
  double support phase : 2 feet in contact with the ground



the ZMP can be transfered by one foot to the other without going out the support polygon

Autonomous and Mobile Robotics

Prof. Giuseppe Oriolo

Humanoid Robots 2:

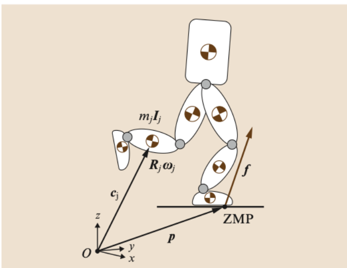
Dynamic Modeling

DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI

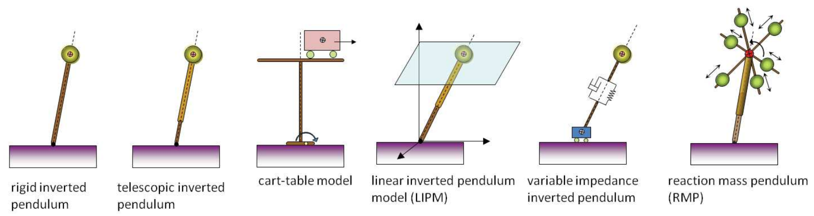
SAPIENZA
UNIVERSITÀ DI ROMA

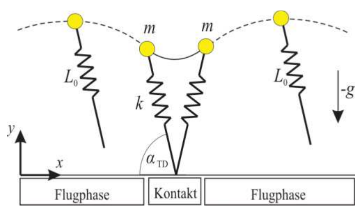
modeling

- multi-body free floating complete model



- conceptual models





for walking/balancing

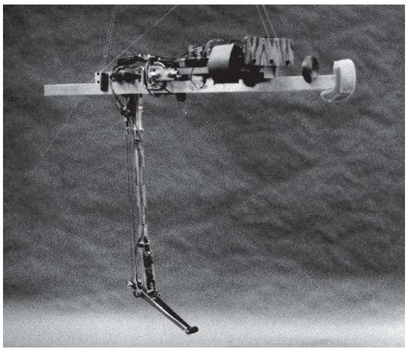
for running

like a manipulator?

can we consider this as a part (leg) of a legged robot?

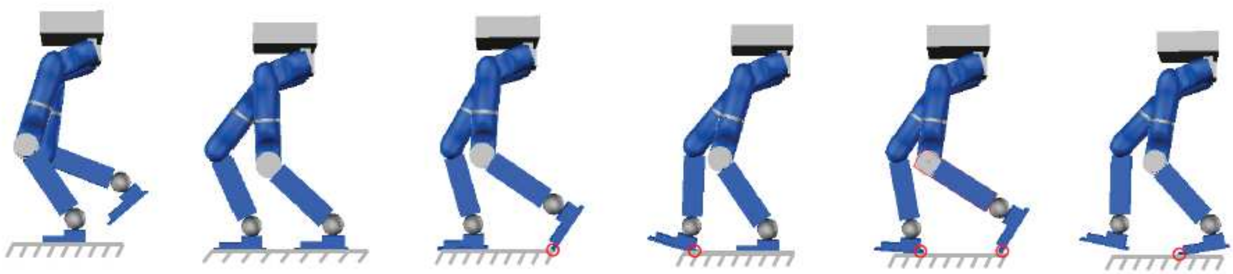
NO: this manipulator cannot fall because its base is clamped to the ground

this is a one-legged robot: Monopod from MIT



floating-base model

the difference lies in the contact forces

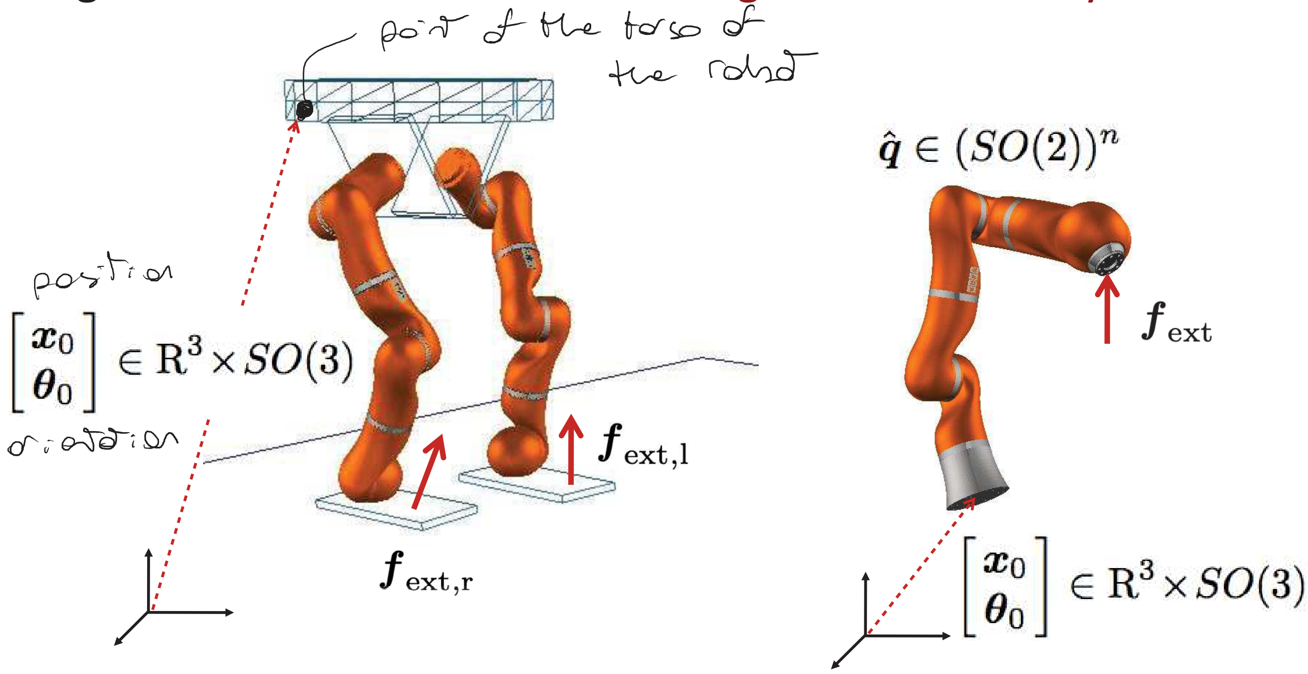


one may look at these contact configurations as different fixed-base robots, each with a specific kinematic and dynamic model

→ or consider a single floating-base system with limbs that may establish contacts

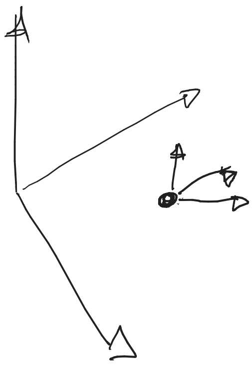
floating-base model

the general model is that of a floating-base multi-body

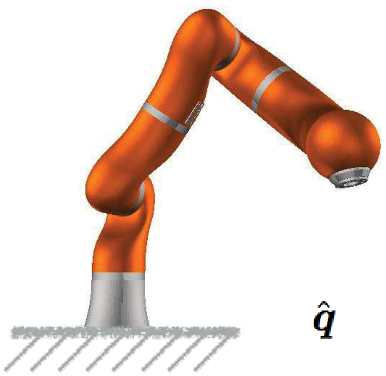


$\{q_i\}$ joints configuration

$\{q_i\} \cup \{\overset{3\ \text{vec}}{x_b}, \overset{3\ \text{vec}}{\theta}, \}$

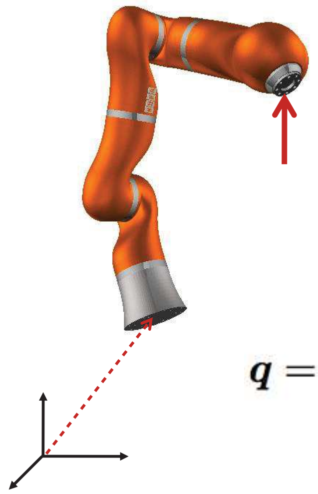


configuration



$\hat{\boldsymbol{q}}$

vs.



$$\boldsymbol{q} = \begin{bmatrix} \hat{\boldsymbol{q}} \\ \boldsymbol{x}_0 \\ \boldsymbol{\theta}_0 \end{bmatrix}$$

$$\hat{\boldsymbol{B}}(\hat{q})\ddot{\hat{q}} + \hat{\boldsymbol{C}}(\hat{q}, \dot{\hat{q}})\dot{\hat{q}} + \hat{e}(\hat{q}) = \hat{\boldsymbol{\tau}} + \hat{\boldsymbol{J}}^T(\hat{q})\boldsymbol{f}_{ext}$$

formally similar

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{e}(\boldsymbol{q}) = \boldsymbol{\tau} + \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{f}_{ext}$$

Lagrangian dynamics

assumption: we consider as humanoid only a mechanical leg

dynamic equations (general form)

$$\boldsymbol{B}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{e}(\boldsymbol{q}) = \boldsymbol{\tau} + \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{f}_{ext}$$

but here we have a special structure

$$\boldsymbol{B}(\boldsymbol{q})\left(\begin{bmatrix} \ddot{\hat{\boldsymbol{q}}} \\ \ddot{\boldsymbol{x}}_0 \\ \ddot{\boldsymbol{\theta}}_0 \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{g} \\ \boldsymbol{0} \end{bmatrix}\right) + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \begin{bmatrix} \boldsymbol{\tau} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} + \sum_i \boldsymbol{J}_i^T(\boldsymbol{q})\boldsymbol{f}_i$$

where $-\boldsymbol{g}$ is the (Cartesian) gravity acceleration vector and $\boldsymbol{J}_i$ is the Jacobian matrix associated to the i-th contact force $\boldsymbol{f}_i$

Lagrangian dynamics

$$\boldsymbol{B}(\boldsymbol{q})\left(\begin{bmatrix}\ddot{\boldsymbol{q}}\\ \ddot{\boldsymbol{x}}_0\\ \ddot{\boldsymbol{\theta}}_0\end{bmatrix}+\begin{bmatrix}\boldsymbol{0}\\ \boldsymbol{g}\\ \boldsymbol{0}\end{bmatrix}\right)+\boldsymbol{n}(\boldsymbol{q},\dot{\boldsymbol{q}})=\begin{bmatrix}\boldsymbol{\tau}\\ \boldsymbol{0}\\ \boldsymbol{0}\end{bmatrix}+\sum_i \boldsymbol{J}_i^T(\boldsymbol{q})\boldsymbol{f}_i$$

mass/inertia — accelerations — forces/torques

- centrifugal/Coriolis terms
- joint torques
- contact forces

joint torques only affect joint coordinates!
to move $\boldsymbol{x}_0$ (i.e., the position of the reference body) the contact forces are necessary

General definition

A humanoid leg has as configuration the vector x made up of the following 3 groups of parameters:
- → the configuration q of the leg (as a fixed base)
- → the position p and the orientation θ of the base

The dynamic model is:

$$B(q)\begin{pmatrix}\ddot{q}\\ \ddot{p}+g\\ \ddot{\theta}\end{pmatrix}+n(q,\dot{q})=\begin{pmatrix}\tau\\ 0\\ 0\end{pmatrix}+\tau_{ext}$$

where
- $-g$ is the Cartesian gravity acceleration vector
- τ is torque applied by the actuators
- τ_{ext} is the torque generated by the resulting of the external forces

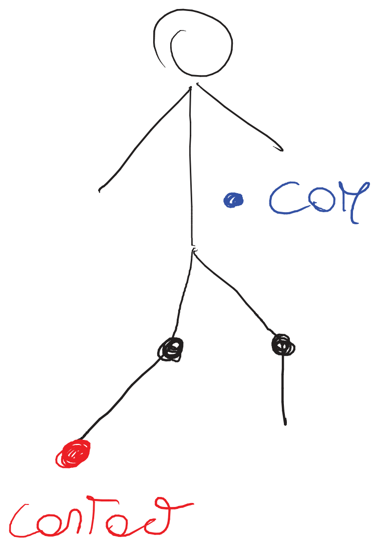
For sake of simplicity we consider only the contact forces as external forces, that is the reaction forces applied on the contact points of the feet with the ground.

If there are m contact forces:

$$\tau_{ext} \triangleq \sum_{i=1}^{m} J_i^T(q) f_i$$

↳ Jacobian associated to the i-th contact force

Note that τ acts only on the dynamics of q, therefore in order to modify p and θ, external forces are necessary



it depends on the configuration of every single joint

NEWTON - EULER EQ

Balance forces & moments acting on the entire robot

Newton - Euler equations

We will focus only on the subsets regarding the p and θ parameters

(p) the subset can be expressed as

$$M(\ddot{c}+g) = \sum_{i=1}^{m} f_i \iff \ddot{c} = \frac{\sum_{i=1}^{m} f_i - Mg}{M}$$

Newton equation: the resulting of the external forces, including the gravity, is equal to $M\ddot{c}$

- c is the position of the CoM (wrt p)
- M is the total mass of the robot

(θ) the subset can be expressed as

$$\dot{L} = \sum_{i=1}^{m} (p_i - c) \times f_i$$

Euler equation: the resulting of the moments wrt the CoM is the variation of the angular moment

- p_i is the position of the application point of the i-th force (wrt p)
- c is the position of the CoM (wrt p)
- L is the angular momentum of the robot

Newton-Euler equations

$$\boldsymbol{B}(\boldsymbol{q})\left(\begin{bmatrix} \ddot{\hat{\boldsymbol{q}}} \\ \ddot{\boldsymbol{x}}_0 \\ \ddot{\boldsymbol{\theta}}_0 \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{g} \\ \mathbf{0} \end{bmatrix}\right) + \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \begin{bmatrix} \boldsymbol{\tau} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} + \sum_i \boldsymbol{J}_i^T(\boldsymbol{q}) \boldsymbol{f}_i$$

the second and third rows of the Lagrangian dynamics express the linear and rotational dynamics of the whole robot

these correspond to the **Newton-Euler equations**, obtained by balancing **forces** and **moments** acting on the robot as a whole

Newton-Euler equations

Newton equation:
variation of linear momentum = force balance

$$M\ddot{\boldsymbol{c}} = \sum_i \boldsymbol{f}_i - M\boldsymbol{g}$$

$\boldsymbol{c}$: CoM position
M : total mass of the system

hence: we need contact forces to move the CoM in a direction different from that of gravity!

Newton-Euler equations

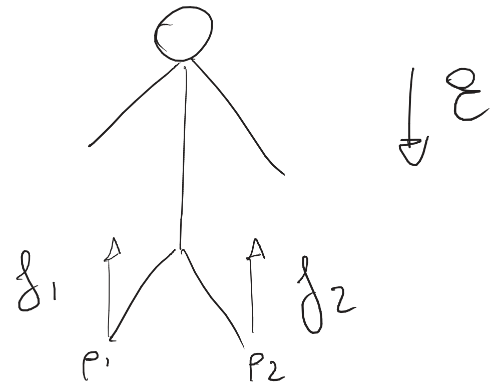
Euler equation:
variation of angular momentum = moment balance

$$(\boldsymbol{c}-\boldsymbol{o}) \times M\ddot{\boldsymbol{c}} + \dot{\boldsymbol{L}} = \sum_i (\boldsymbol{p}_i - \boldsymbol{o}) \times \boldsymbol{f}_i - (\boldsymbol{c}-\boldsymbol{o}) \times M\boldsymbol{g}$$

moments are computed wrt to a generic point $\boldsymbol{o}$

$\boldsymbol{p}_i$: position of the contact point of force $\boldsymbol{f}_i$
$\boldsymbol{L}$: angular momentum of the robot wrt its CoM

o any point



starting from

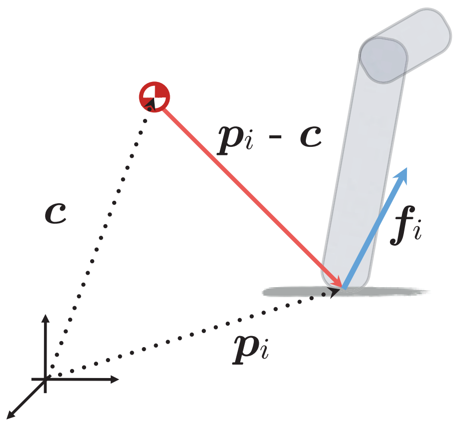
$$m\ddot{c} = \sum f_i - mg$$

Variation of the angular momentum $= \sum (p_i - o) \times f_i - (c - o) \times g$

$$(c-o)\,\ddot{c} + \dot{L} = \sum (p_i - o) \times f_i - (c - o) \times g$$

Newton-Euler equations

recall: the moment of a force (or torque) is a measure of its tendency to cause a body to rotate about a specific point or axis



$(\boldsymbol{p}_i - \boldsymbol{c}) \times \boldsymbol{f}_i$

moment generated by the contact force $\boldsymbol{f}_i$ around the CoM

angular momentum around the CoM:
sum of the angular momentum of each robot link

$$\boldsymbol{L} = \sum_k (\boldsymbol{x}_k - \boldsymbol{c}) \times m_k \dot{\boldsymbol{x}}_k + \boldsymbol{I}_k \boldsymbol{\omega}_k$$

$\boldsymbol{\omega}_k$: angular velocity of the k-th link

Flight phase

Is the phase when the leg is not in contact with the ground
therefore no forces act on it, a part for the gravity.
The condition for the flight phase is

$$\forall i, \ \boldsymbol{f}_i = 0$$

in that case the second and the third dynamical
subset become

$$\ddot{c} = -g \qquad \dot{L} = 0$$

Zero Moment Point

in the equation of moment balance

$$(\boldsymbol{c}-\boldsymbol{o}) \times M\ddot{\boldsymbol{c}} + \dot{\boldsymbol{L}} = \sum_i (\boldsymbol{p}_i - \boldsymbol{o}) \times \boldsymbol{f}_i - (\boldsymbol{c}-\boldsymbol{o}) \times M\boldsymbol{g}$$

choose the point $\boldsymbol{o}$ so that $\sum_i (\boldsymbol{p}_i - \boldsymbol{o}) \times \boldsymbol{f}_i$ is zero

this is the Zero Moment Point (ZMP), i.e., the point wrt to which the moment of the contact forces is zero

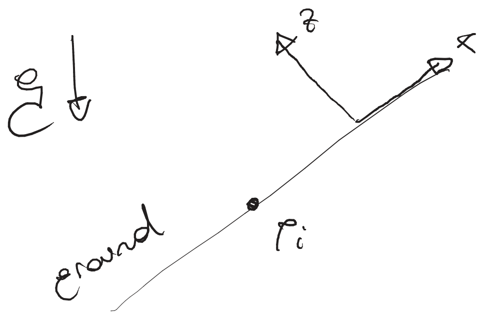
we denote this point by $\boldsymbol{z}$

Flat ground hypothesis

When the ground is at the same level (flat), that is when it is possible to build a reference frame such that

$$\forall i \quad p_{z,i} = 0 \quad \rightarrow \text{flat ground hypothesis verified}$$

Note: a flat ground doesn't mean that the ground is horizontal



Zero Moment Point ZMP

2 Definitions:

1: The point x_{ZMP} on the ground such that $\left[\sum_{i=1}^{n} (p_i - x_{ZMP}) \times f_i\right]_{x,y} = 0$

that is, the point on the ground such that the projection on the ground of the resulting moment of the contact forces computed wrt the first is null (i.e. the moment is orthogonal to the ground)

Newton-Euler on flat ground

combine the Newton and Euler equations: divide the Euler equation by the z-component of Newton equation

$$M(\ddot{c}^z + g^z) = \sum_i f_i^z$$

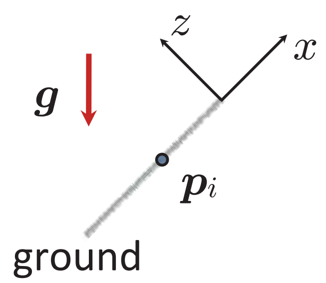
leads to

$$\frac{M(\boldsymbol{c}-\boldsymbol{z}) \times (\ddot{\boldsymbol{c}}+\boldsymbol{g}) + \dot{\boldsymbol{L}}}{m(\ddot{c}^z + g)} = \frac{\sum_i (\boldsymbol{p}_i - \boldsymbol{z}) \times \boldsymbol{f}^i}{\sum_i f_i}$$

flat ground hypothesis (not necessarily horizontal)

$$p_i^z = 0$$

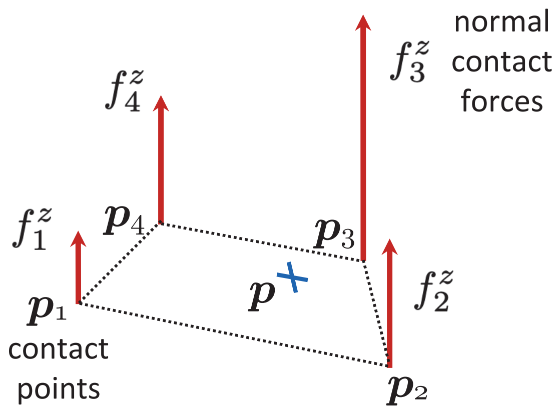
and we may have $\boldsymbol{g}^{x,y} \neq 0$



Center of Pressure

the **Center of Pressure (CoP)** is a point defined for a set of forces acting on a flat surface

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



flat ground: the CoP corresponds to the point of application of the **Ground Reaction Force vector** (GRF)

note: GRF can also have a horizontal component (friction)

Center of Pressure

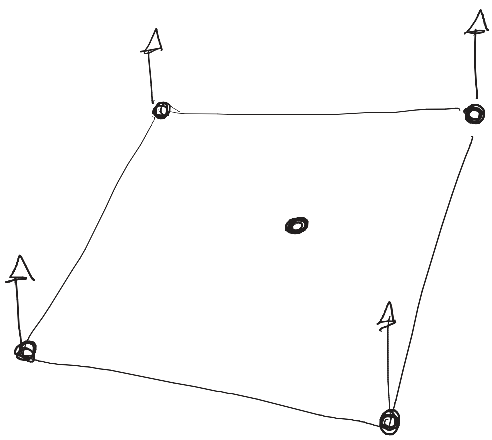
on flat ground, the moment balance equation tells us that the CoP and the ZMP coincide

$$\frac{M(\boldsymbol{c}-\boldsymbol{z})\times(\ddot{\boldsymbol{c}}+\boldsymbol{g})+\dot{\boldsymbol{L}}}{m(\ddot{c}^z+g)}=\frac{\sum_i(\boldsymbol{p}_i-\boldsymbol{z})\times\boldsymbol{f}^i}{\sum_i f_i}=\boldsymbol{p}-\boldsymbol{z}=0$$

$f_i^z \geq 0$ the vertical component of the contact forces can only be positive (**unilateral** force)

therefore the CoP/ZMP must belong to the convex hull of the contact points, i.e. the **Support Polygon**

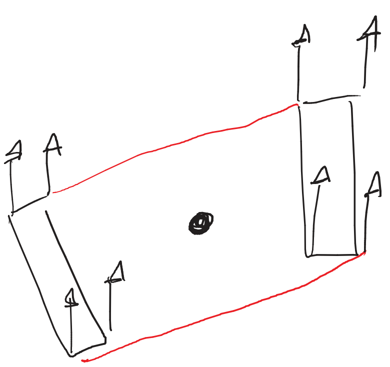
sufficient condition for balance

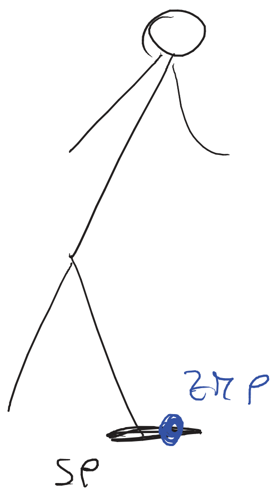


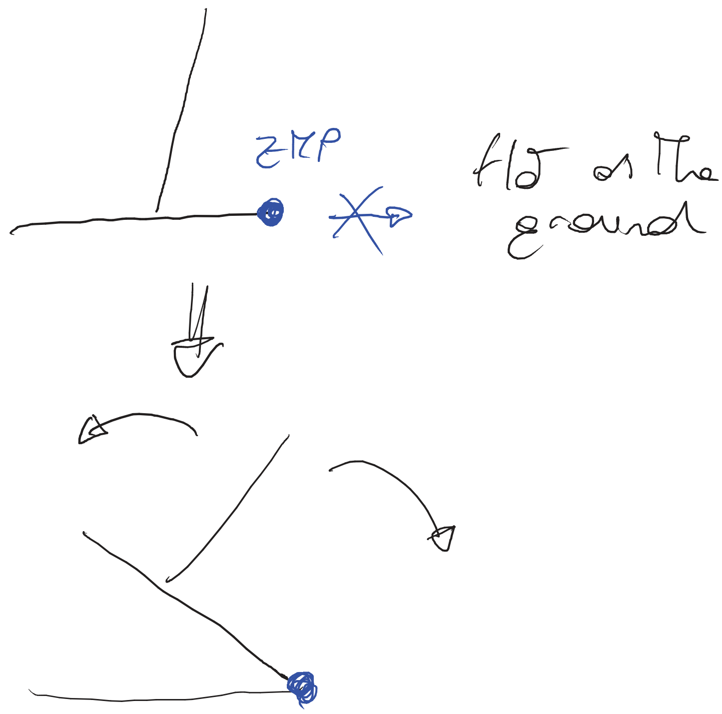
forces acting on the corners

$$\frac{\sum p_i f_i^z}{\sum f_i^z} = \text{CoP}$$

coincides with the ZMP







MODELING → LAGRANGE → floating base $\equiv$ CoM

MODELING → NEWTON-EULER equations

Center of Pressure

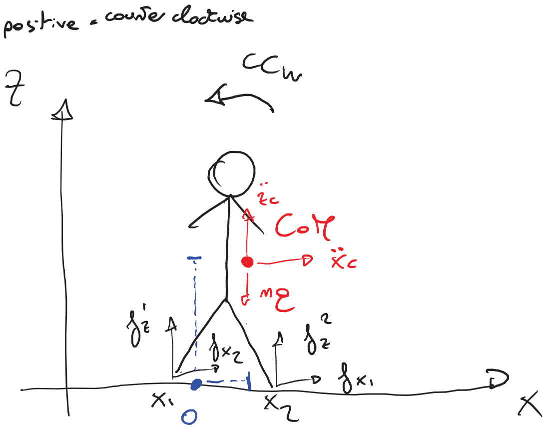
In the flat ground hypothesis is said CoP the point with position

$$x_{CoP} \triangleq \sum_{i=1}^{n} \left(\frac{f_{zi}}{\sum_{i=1}^{n} f_{zi}} \right) p_i \xrightarrow{\text{for the flat ground hyp}} \begin{pmatrix} p_{ix} \\ p_{iy} \\ 0 \end{pmatrix}$$

that is the centroid of the contact points wrt the weights $\dfrac{f_{zi}}{\sum_{i=1}^{n} f_{zi}}$

CoP has a geometric property:
By definition, $f_i^z \geq 0$ always, in fact the robot is always over the ground and not under, so the reaction forces are always positive.
x_{cop} is always in the convex hull of the contact points (inside the support polygon)

Flat ground hypothesis $\Longrightarrow$ CoP $\equiv$ ZMP



y-component of the moment balance equation

$\sum$ moments wrt O = variation of angular moment

$$\ddot{z}_c = -g + \sum f_z^i \rightarrow \ddot{z}_c + g = \sum f_z^i$$

CW CCW

$$-m\ddot{x}_c z_c + m\ddot{z}_c (x_c - x_0) + \dot{L}_y = -mg(x_c - x_0) - f_z^1 (x_0 - x_1) + f_z^2 (x_2 - x_0)$$

$$\frac{-m\ddot{x}_c + m(\ddot{z}_c + g)(x_c - x_0) + \dot{L}_y}{\ddot{z}_c + g} = \frac{\sum f_z^i (x_i - x_0)}{\sum f_z^i}$$

$$-\frac{\ddot{x}_c}{\ddot{z}_c + g} + x_c - x_0 + \dot{L}_y = \frac{\sum f_z^i (x_i - x_0)}{\sum f_z}$$

CoP $\equiv$ ZMP

Flat ground hypothesis → resulting forces

- Normal forces: $f^n = \begin{pmatrix} f_x^n \\ f_y^n \\ f_z^n \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \sum_{i=1}^{n} f_{zi} \end{pmatrix}$

- Tangential forces: $f^t = \begin{pmatrix} f_x^t \\ f_y^t \\ f_z^t \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} f_{xi} \\ \sum_{i=1}^{n} f_{yi} \\ 0 \end{pmatrix}$

Resulting moment of the tangential forces

The sum of the moments of the tangential forces wrt x_{CoP}

$$\tau^t \triangleq \sum_{i=1}^{n} (p_i - x_{CoP}) \times f_i^t$$

Newton-Euler on flat ground

flat ground $p_i^z = 0$ first two components (x and y)

$$c^y - \frac{c^z}{\ddot{c}^z + g^z}(\ddot{c}^y + g^y) + \frac{\dot{L}^x}{M(\ddot{c}^z + g^z)} = \frac{\sum_i f_i^z p_i^y}{\sum_i f_i^z}$$

$$c^x - \frac{c^z}{\ddot{c}^z + g^z}(\ddot{c}^x + g^x) - \frac{\dot{L}^y}{M(\ddot{c}^z + g^z)} = \frac{\sum_i f_i^z p_i^x}{\sum_i f_i^z}$$

or in compact form

CoM position

small or zero cause we're on flat ground

$$\boxed{\boldsymbol{c}^{x,y} - \frac{c^z}{\ddot{c}^z + g^z}(\ddot{\boldsymbol{c}}^{x,y} + \boldsymbol{g}^{x,y}) + \frac{\boldsymbol{S}\dot{\boldsymbol{L}}^{x,y}}{M(\ddot{c}^z + g^z)} = \frac{\sum_i f_i^z \boldsymbol{p}_i^{x,y}}{\sum_i f_i^z}}$$

→ very small term unless we are doing very dynamical movement (very fast)

with $\boldsymbol{S} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$

more on the CoP

the Center of Pressure (CoP) $\boldsymbol{z}$ is usually defined as the point on the ground where the resultant of the ground reaction force acts

we have 2 types of interaction forces at the foot/ground interface: **normal forces** f_i^z and tangential forces $\boldsymbol{f}_i^{x,y}$

the CoP may be defined as the point $\boldsymbol{z}$ where the resultant of the normal forces f_i^z acts

$$f^z = \sum_i f_i^z$$

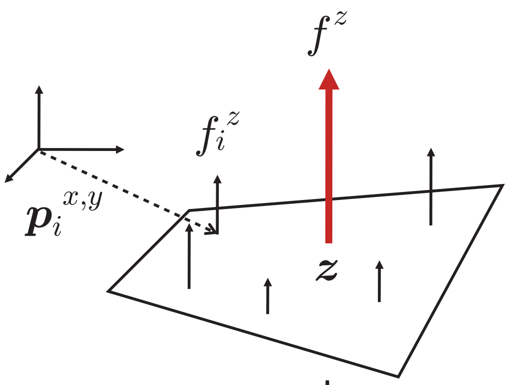
the resultant of the tangential forces may be represented at $\boldsymbol{z}$ by a force $\boldsymbol{f}^{x,y}$ and a moment M_t

$$\boldsymbol{f}^{x,y} = \sum_i \boldsymbol{f}_i^{x,y} \qquad\qquad M_t = \sum_i \boldsymbol{r}_i \times \boldsymbol{f}_i^{x,y}$$

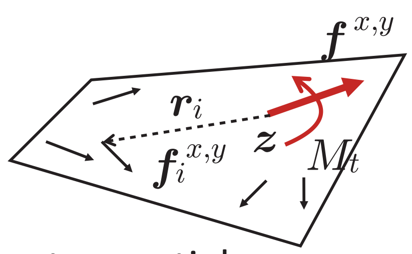
where $\boldsymbol{r}_i$ is the vector from z to the point of application of $\boldsymbol{f}_i^{x,y}$

more on the CoP

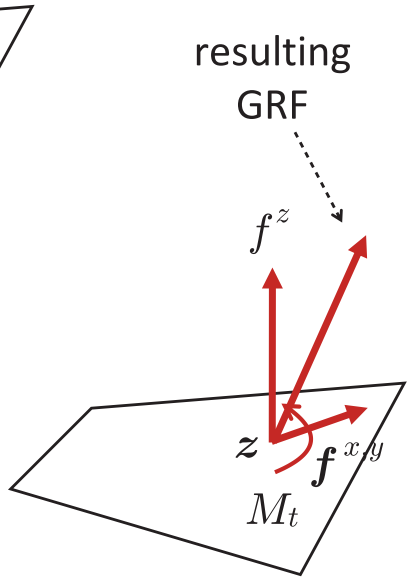
the sum of the normal and tangential components gives the resulting GRF



normal forces



tangential forces



$$f^z = \sum_i f_i^z$$

$$\frac{\sum_i f_i^z \boldsymbol{p}_i^{x,y}}{\sum_i f_i^z} = \boldsymbol{z}^{x,y}$$

$$\boldsymbol{f}^{x,y} = \sum_i \boldsymbol{f}_i^{x,y}$$

$$M_t = \sum_i \boldsymbol{r}_i \times \boldsymbol{f}_i^{x,y}$$

Ground Reaction Force Vector, GRF

Is the resulting of the contact forces, that is the vector

$$f^{tot} \triangleq \sum_{i=1}^{m} f_i \triangleq f^n + f^t$$

when it is applied in the CoP

Lagrangian dynamics: multi-body system

projection of the acceleration of the CoM vector on the ground

$$\boldsymbol{c}^{x,y} - \frac{c^z}{\ddot{c}^z + g^z}(\ddot{\boldsymbol{c}}^{x,y} + \boldsymbol{g}^{x,y}) + \frac{\boldsymbol{S}\dot{\boldsymbol{L}}^{x,y}}{M(\ddot{c}^z + g^z)} = \frac{\sum_i f_i^z \boldsymbol{p}_i^{x,y}}{\sum_i f_i^z}$$

$\downarrow$ rewritten as (isolating $\ddot{c}^{x,y}$)

constant

$$\frac{c^z}{\ddot{c}^z + g^z}(\ddot{\boldsymbol{c}}^{x,y} + \boldsymbol{g}^{x,y}) = (\boldsymbol{c}^{x,y} - \boldsymbol{z}^{x,y}) + \frac{\boldsymbol{S}\dot{\boldsymbol{L}}^{x,y}}{M(\ddot{c}^z + g^z)}$$

we can analyze the effect of the various terms on the CoM horizontal acceleration (horizontal = in the x-y plane)

Lagrangian dynamics: multi-body system

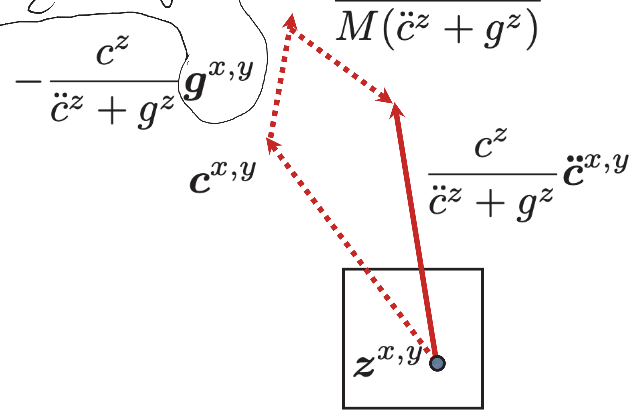
constant constant centrifugal force which pushes away $(c^{x,y})$

$$\frac{c^z}{\ddot{c}^z+g^z}\ddot{\boldsymbol{c}}^{x,y} = -\frac{c^z}{\ddot{c}^z+g^z}\boldsymbol{g}^{x,y} + (\boldsymbol{c}^{x,y}-\boldsymbol{z}^{x,y}) + \frac{\boldsymbol{S}\dot{\boldsymbol{L}}^{x,y}}{M(\ddot{c}^z+g^z)}$$

constant

projection on the ground of the gravity

variation of the angular momentum

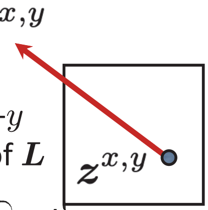


x-y plane

support polygon

aside from the effect of gravity (horizontal components) and variations of the angular momentum, the CoM horizontal acceleration is the result of a force pushing the CoM away from the CoP

along the line between them on the support polygon



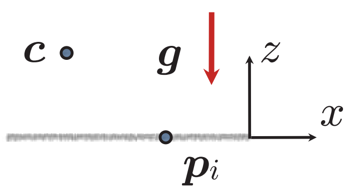
no gravity in x-y
no variations of $\boldsymbol{L}$

, with an intensity proportional to the distance

Lagrangian dynamics: approximations

on **horizontal** flat ground + CoM at **constant height** + neglect $\dot{\boldsymbol{L}}^{x,y}$

$$\boldsymbol{g}^{x,y} = 0 \qquad c^z = \text{constant}$$



ground

$$\frac{c^z}{\cancel{\ddot{c}^z}+g^z}(\ddot{\boldsymbol{c}}^{x,y}+\cancel{\boldsymbol{g}^{x,y}}) = (\boldsymbol{c}^{x,y}-\boldsymbol{z}^{x,y}) + \cancel{\frac{\boldsymbol{S}\dot{\boldsymbol{L}}^{x,y}}{M(\ddot{c}^z+g^z)}}$$

$$\boxed{\boldsymbol{c}^{x,y} - \frac{c^z}{g^z}\ddot{\boldsymbol{c}}^{x,y} = \boldsymbol{z}^{x,y}}$$

or

$$\boxed{\ddot{\boldsymbol{c}}^{x,y} = \frac{g^z}{c^z}(\boldsymbol{c}^{x,y}-\boldsymbol{z}^{x,y})}$$

Linear Inverted Pendulum (LIP)

Linear Inverted Pendulum interpretation

2 independent equations

$$\boxed{\ddot{\boldsymbol{c}}^{x,y} = \frac{g^z}{c^z}\left(\boldsymbol{c}^{x,y} - \boldsymbol{z}^{x,y}\right)}$$

acceleration of the CoM

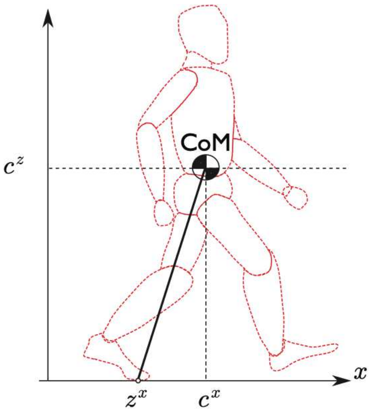
how the CoM moves in

$$\ddot{c}^x = \frac{g^z}{c^z}\left(c^x - z^x\right)$$

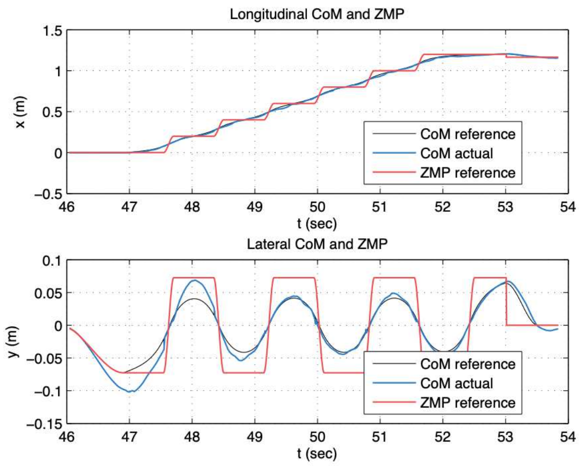
longitudinal direction (sagittal plane)

$$\ddot{c}^y = \frac{g^z}{c^z}\left(c^y - z^y\right)$$

lateral direction

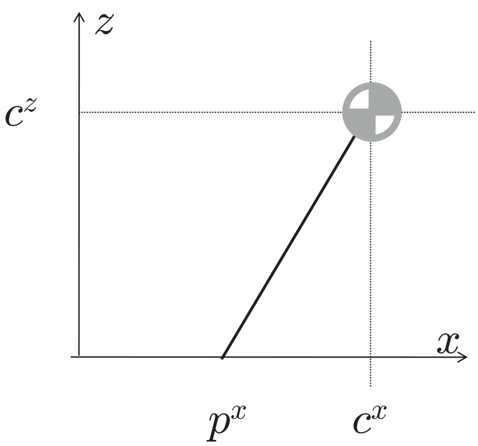


typical behaviors



Linear Inverted Pendulum interpretation

- Point foot
  the simplest interpretation of the LIP is that of a telescoping (so to remain at a constant height) massless leg in contact with the ground at p^x (point of contact)



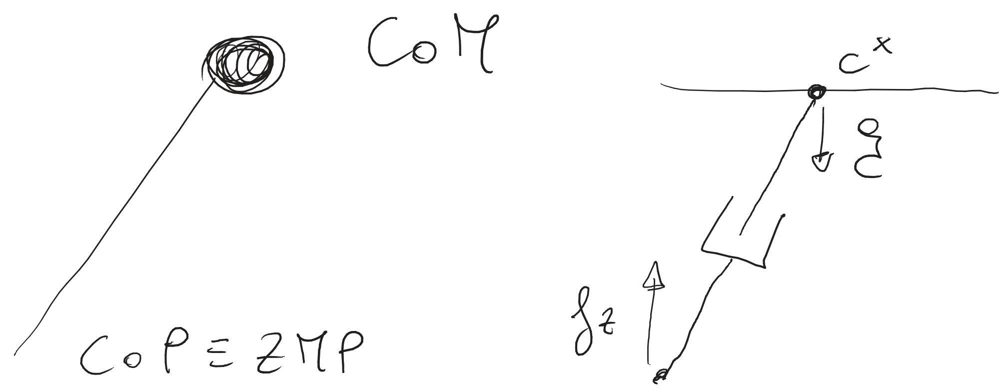
we can interpret the (**longitudinal** direction) LIP equation as a moment balance around p^x

$$M\ddot{c}^x c^z - Mg(c^x - p^x) = 0$$

i.e. $$\ddot{c}^x = \frac{g^z}{c^z}\left(c^x - p^x\right)$$

in this case the ZMP z^x coincides with the point of contact p^x of the fictitious leg

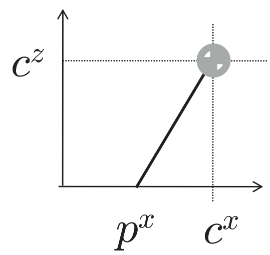
$$p^x = z^x$$

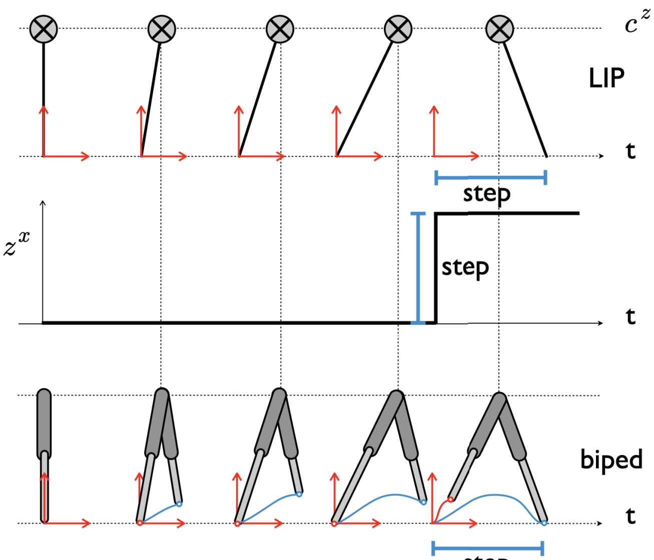


Linear Inverted Pendulum interpretation

- Point foot (longitudinal direction)

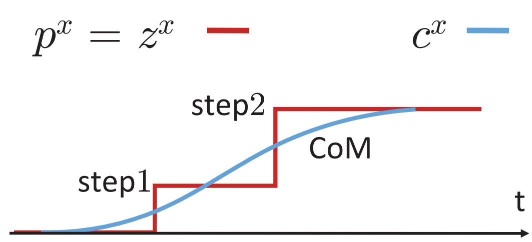
dynamic law: $\frac{c_z}{g_z}\ddot{c}_x = (c_x - z^x)$





The change of foot is when c_x passes a half of the step's lenght

typical footsteps and CoM



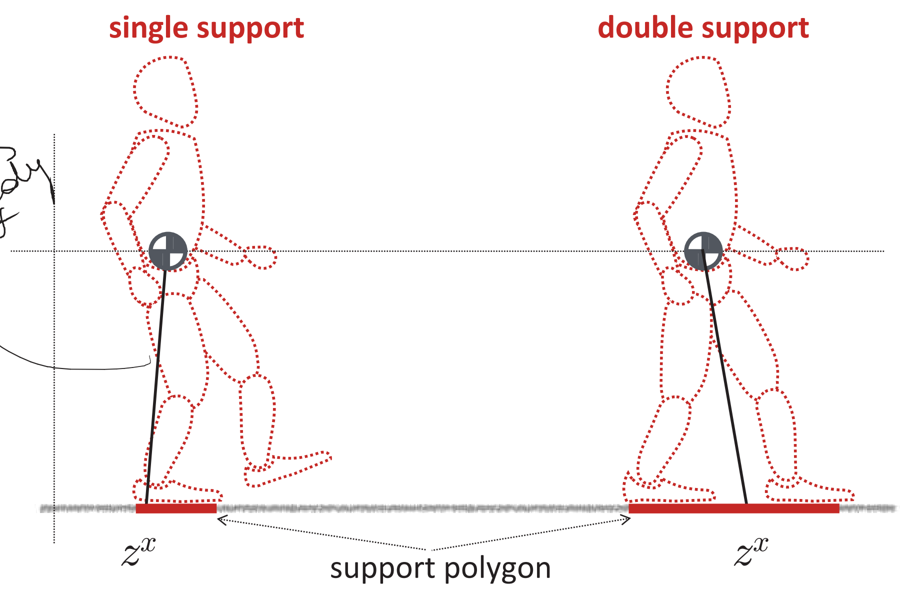
may also be seen as a compass biped with only one leg touching the ground at the same time

The robot doesn't touch the ground with both legs (no double support phase)

Linear Inverted Pendulum interpretation

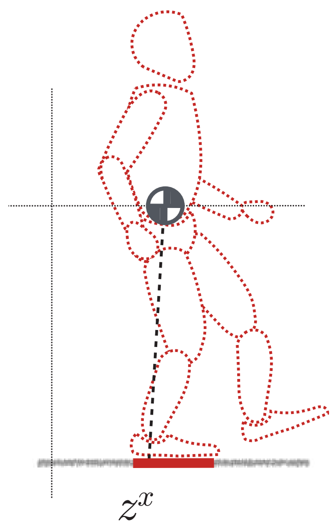
- Finite sized foot (with ankle torque τ_y)
  since z^x represents the ZMP location, there is no difficulty in extending the interpretation of the LIP considering both single and double support phases with a finite foot dimension

the leg is substituted by a line without mass and the body with a point of mass M

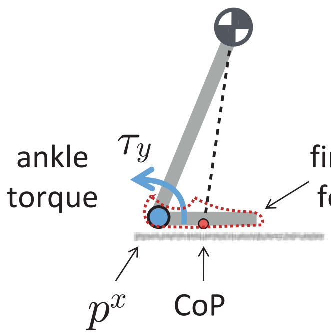


Linear Inverted Pendulum interpretation

- Finite sized foot (with ankle torque τ_y) → scalar
  we can see the single support phase from the stance foot point of view i.e. with the dynamics of the rest of the humanoid represented by an equivalent fictitious leg. A way to keep the CoM balanced is using an equivalent ankle torque (the real joint torques are such that an equivalent ankle torque is applied)



finite foot with equivalent massless leg plus ankle torque →

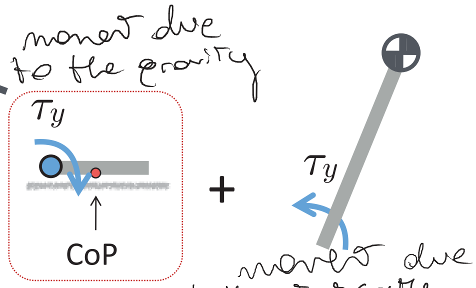


moment w.r.t. z^x (CoP) $= 0$

$$Mg(z^x - p^x) - \tau_y = 0$$

$$z^x = p^x + \frac{\tau_y}{Mg}$$

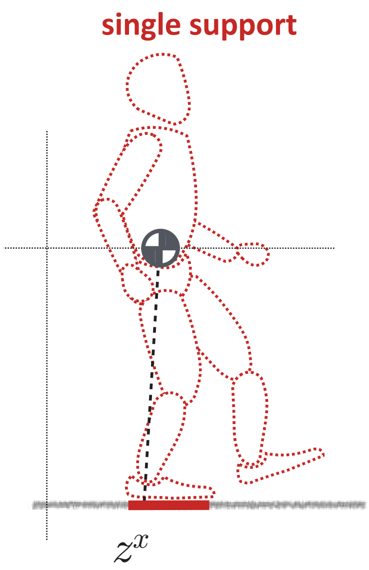
moment due to the gravity



moment due to the actuator at the ankle.

Linear Inverted Pendulum interpretation

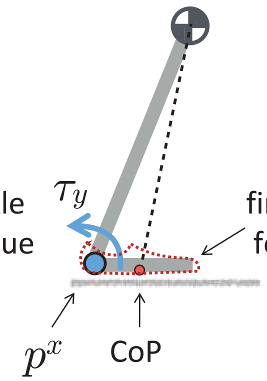
- Finite sized foot (with ankle torque τ_y)
  note: it is possible to move the CoP through the ankle torque τ_y without stepping



$$M\ddot{c}^x c^z - Mg(c^x - p^x) + \tau_y = 0$$

$$\ddot{c}^x = \frac{g^z}{c^z}\left(c^x \underbrace{- p^x - \frac{\tau_y}{Mg}}_{-\text{CoP} = -z^x}\right)$$

finite foot with equivalent massless leg plus ankle torque →



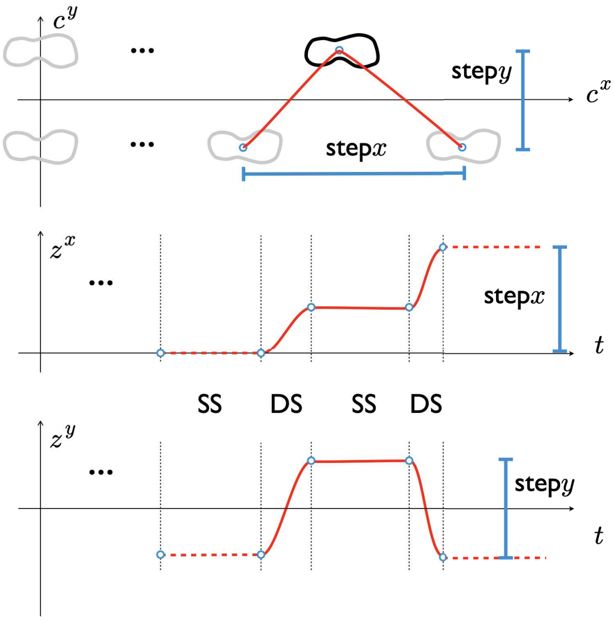
Linear Inverted Pendulum interpretation

- Finite sized foot (with ankle torque τ_y)

with $z^x = p^x + \dfrac{\tau_y}{Mg}$

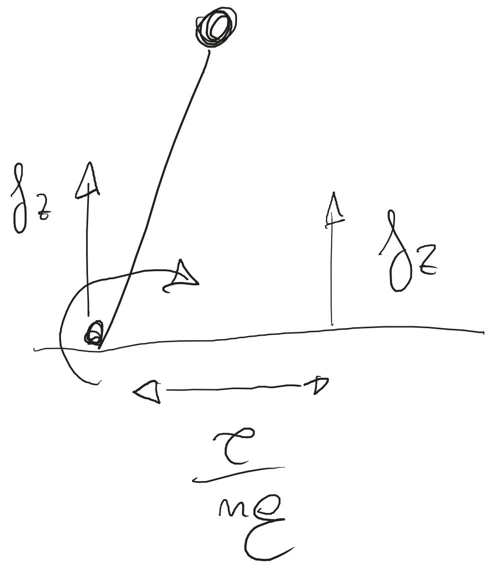
$$\boxed{\ddot{c}^x = \frac{g^z}{c^z}(c^x - z^x)}$$

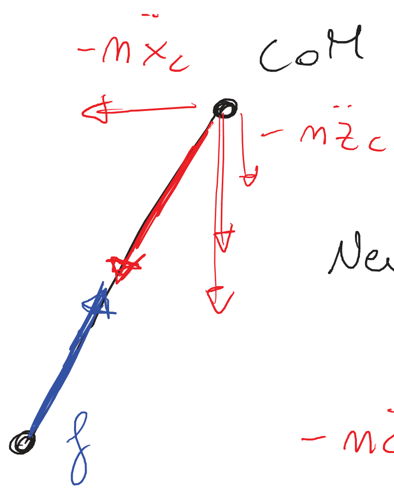
longitudinal direction



typical footsteps with single and double support: for example, in the first single support (- -) the left foot is swinging;
as soon as the right foot touches the ground the double support starts (—) and the ZMP moves from the left to the right foot (longitudinal and lateral motions)

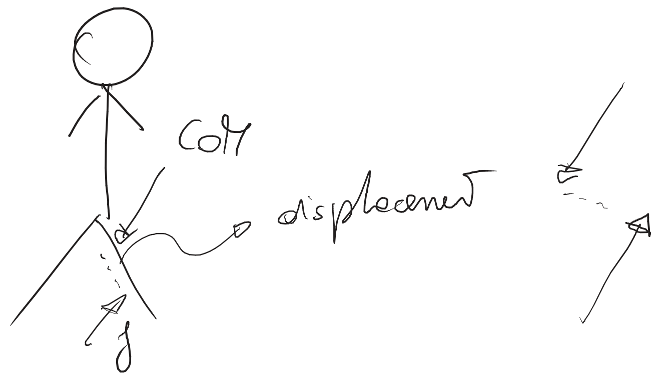
SS: single support
DS: double support



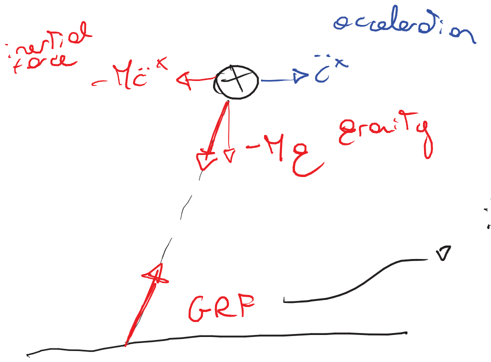


Newton Eq

$$-m\ddot{c} - mg = f$$

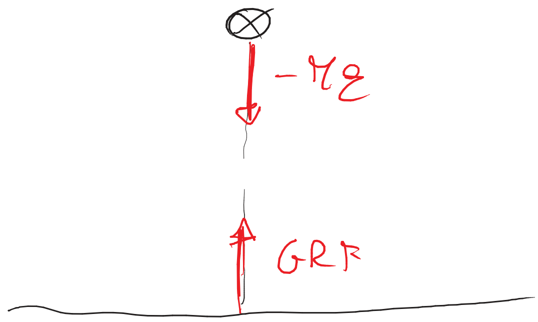


Forces



it is always directed to the CoM

this can be proved by the fact that the moment of the inertial force and of the gravity wrt the CoM is null, because the robot body is a point



- Robot in motion:

  The resulting of the inertial force and of the gravity is countered by GRF

- Robot in stand still

  In this case the only difference is that there's no inertial force

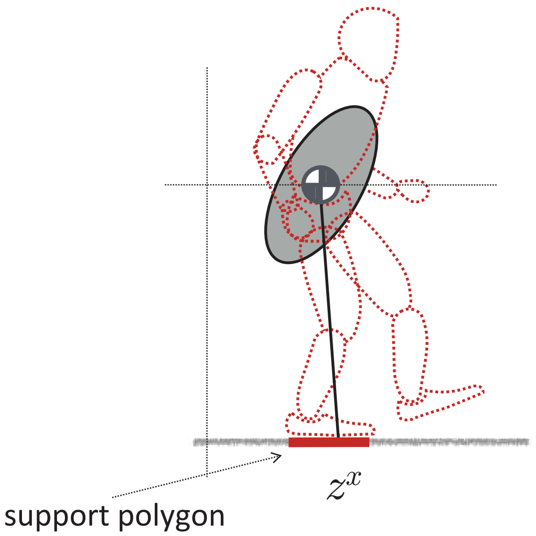
The secret of the gait is that:

a) The resulting of the inertial and gravity forces is discharged down the leg to the ground

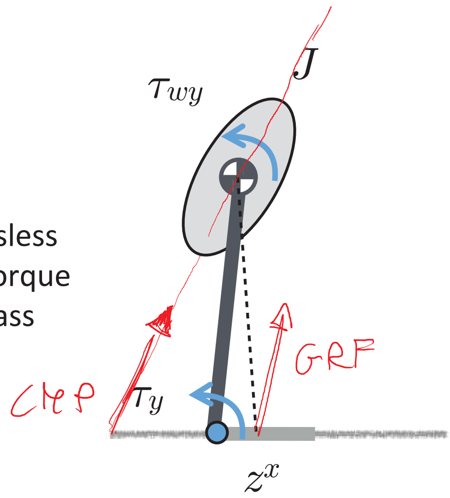
b) The horizontal component of that force produces a friction force (static) that pushes forward

Linear Inverted Pendulum interpretation

- Finite sized foot and reaction mass
  it is also possible to extend the point-mass to be a rigid body with its rotational inertia so that also a hip movement can be modelled



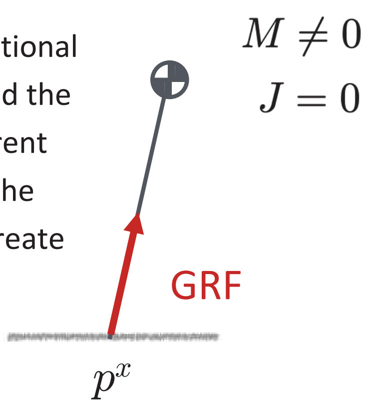
finite foot with equivalent massless leg plus ankle torque and reaction mass

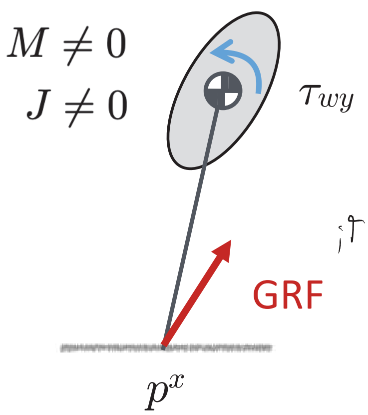


Linear Inverted Pendulum interpretation

- Finite sized foot and reaction mass
  what is the effect of the rotating inertia around the CoM?

since no rotational inertia around the CoM, a different direction of the GRF would create a non-zero moment





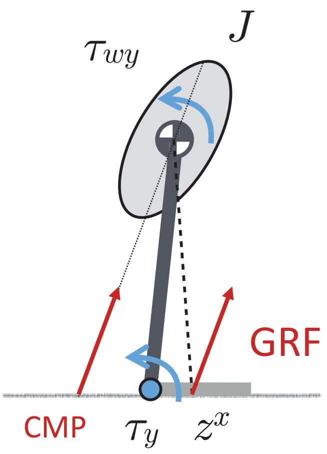
it isn't directed to the CoM but forward

a reaction mass type pendulum, by virtue of its non-zero rotational inertia, allows the ground reaction force to deviate from the lean line. This has important implication in gait and balance.

Linear Inverted Pendulum interpretation

- Finite sized foot and reaction mass

moment around ankle $$M\ddot{c}^x c^z - Mg(c^x - p^x) + \tau_y - \tau_{wy} = 0$$



i.e. $$\ddot{c}^x = \frac{g^z}{c^z}\left(c^x - z^x - \frac{\tau_{wy}}{Mg}\right)$$

- CMP

with $$z^x = p^x + \frac{\tau_y}{Mg}$$

ankle torque moves the CoP while the reaction mass torque changes the GRF direction

to highlight the presence of a non-zero moment around the CoM a new point named Centroidal Moment Pivot (CMP) is introduced and defined as the point where a line parallel to the ground reaction force, passing through the CoM, intersects with the external contact surface

$$X_{CMP} = p^x + \frac{\tau_y + \tau_{wy}}{Mg}$$

Linear Inverted Pendulum: basic scope

$$\boxed{\ddot{\boldsymbol{c}}^{x,y} = \frac{g^z}{c^z}\left(\boldsymbol{c}^{x,y} - \boldsymbol{z}^{x,y}\right)}$$

Although extremely simplified, the LIP equation describes in first approximation the time evolution of the CoM trajectory. Moreover

- it defines a differential relationship between the CoM trajectory and the ZMP (or CMP) time evolution
- it is easier to design a controller which makes the actual CoM follow a desired behaviour
- dynamic balancing will be characterized in terms of the ZMP
- the problem will then be to understand which CoM trajectory, solution of the LIP equation, guarantees that dynamic balancing is achieved

Autonomous and Mobile Robotics

Prof. Giuseppe Oriolo

Humanoid Robots 3:

Balance

Dipartimento di Ingegneria Informatica
Automatica e Gestionale Antonio Ruberti

Sapienza
Università di Roma

recap

- sufficient condition for balance: **ZMP** inside the **support polygon**
- ZMP dynamics modeled from Newton-Euler equations
- approximate model: **Linear Inverted Pendulum (LIP)**

$$\boxed{\ddot{\boldsymbol{c}}^{\boldsymbol{x},\boldsymbol{y}} = \frac{g^z}{c^z}\left(\boldsymbol{c}^{x,y} - \boldsymbol{z}^{x,y}\right)}$$

Linear Inverted Pendulum: basic scope

$$\boxed{\ddot{\boldsymbol{c}}^{x,y} = \frac{g^z}{c^z}\left(\boldsymbol{c}^{x,y} - \boldsymbol{z}^{x,y}\right)}$$

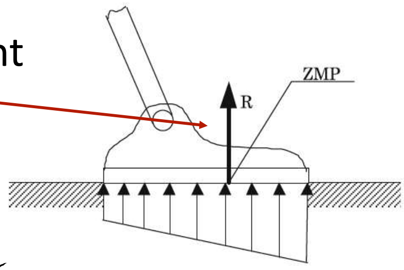
Although extremely simplified, the LIP equation describes in first approximation the time evolution of the CoM trajectory. Moreover

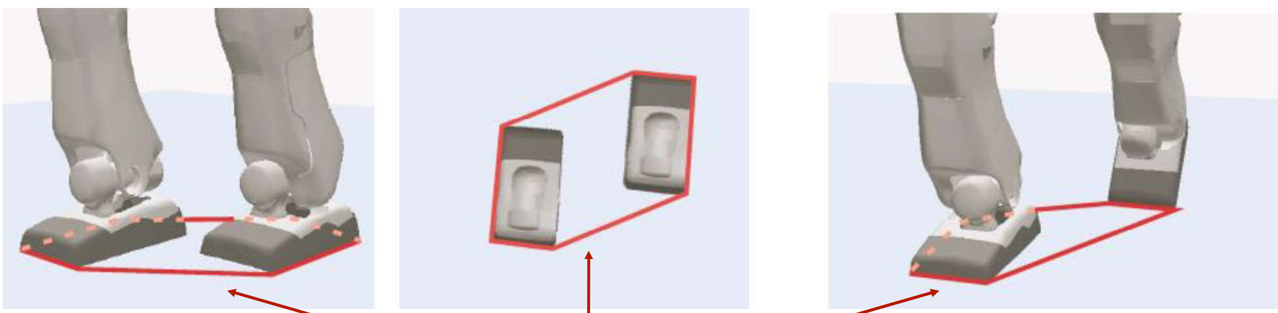
- it defines a differential relationship between the CoM trajectory and the ZMP (or CMP) time evolution
- it is easier to design a controller which makes the actual CoM follow a desired behaviour
- dynamic balancing will be characterized in terms of the ZMP
- the problem will then be to understand which CoM trajectory, solution of the LIP equation, guarantees that dynamic balancing is achieved

ZMP

vertical component of the GRF

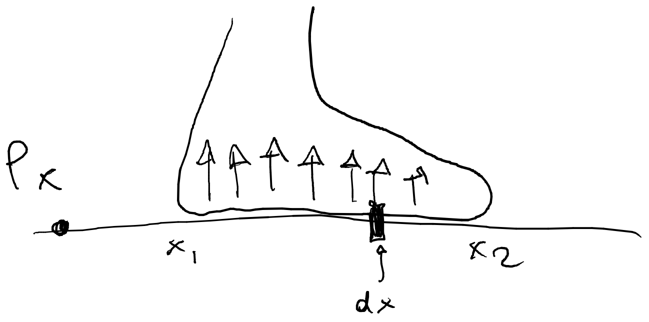
ground reaction force





Support Polygons during Double Support

foot in contact with the ground



$\rho(x)$ vertical distributed ground reaction

$\sigma(x)$ horizontal distributed ground reaction

$$f_x = \int_{x_1}^{x_2} \sigma(x)\,dx$$

$$f_z = \int_{x_1}^{x_2} \rho(x)\,dx$$

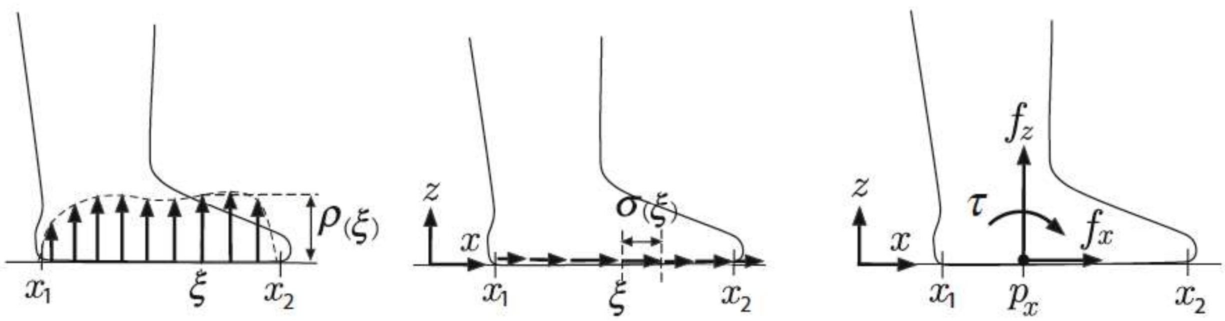
$$-\int_{x_1}^{x_2} (x - p_x)\,\rho(x)\,dx = \tau(p_x)$$

$$-\int_{x_1}^{x_2} x\,\rho(x)\,dx + \underbrace{p_x \int_{x_1}^{x_2} \rho(x)\,dx}_{p_x f_z} = \tau(p_x)$$

$$-\int x\,\rho(x)\,dx + p_x f_z = \underbrace{\tau(p_x)}_{0}$$

$$p_x = \frac{\int x\,\rho(x)\,dx}{f_z}$$

ZMP - 2D case



(a) Vertical force (b) Horizontal force

components of the GRF

$$f_x = \int_{x_1}^{x_2} \sigma(\xi)\,d\xi$$

$$f_z = \int_{x_1}^{x_2} \rho(\xi)\,d\xi$$

equivalent force/torque

$$\tau(p_x) = -\int_{x_1}^{x_2} (\xi - p_x)\rho(\xi)\,d\xi$$

generic p_x

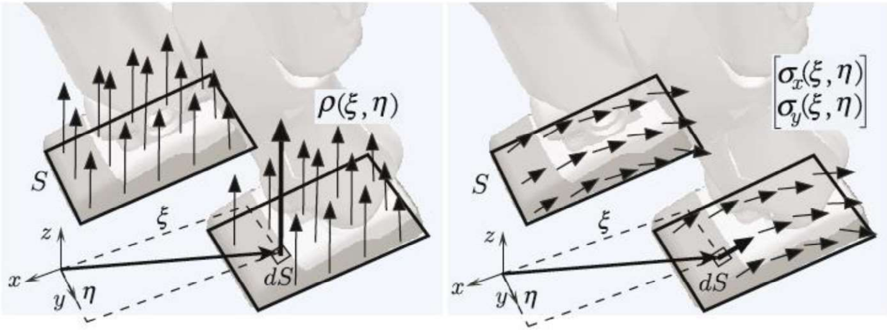
$$\tau(p_x) = 0$$

specific p_x

CoP/ZMP

$$p_x = \frac{\int_{x_1}^{x_2} \xi\rho(\xi)\,d\xi}{\int_{x_1}^{x_2} \rho(\xi)\,d\xi}.$$

ZMP - 3D case



(a) Vertical reaction forces (b) Horizontal reaction forces

vertical component of the GRF

$$f_z = \int_S \rho(\xi,\eta)dS$$

$$\boldsymbol{\tau}_n(\boldsymbol{p}) \equiv [\tau_{nx}\ \tau_{ny}\ \tau_{nz}]^T$$

$$\tau_{nx} = \int_S (\eta - p_y)\rho(\xi,\eta)dS$$

$$\tau_{ny} = -\int_S (\xi - p_x)\rho(\xi,\eta)dS$$

$$\tau_{nz} = 0.$$

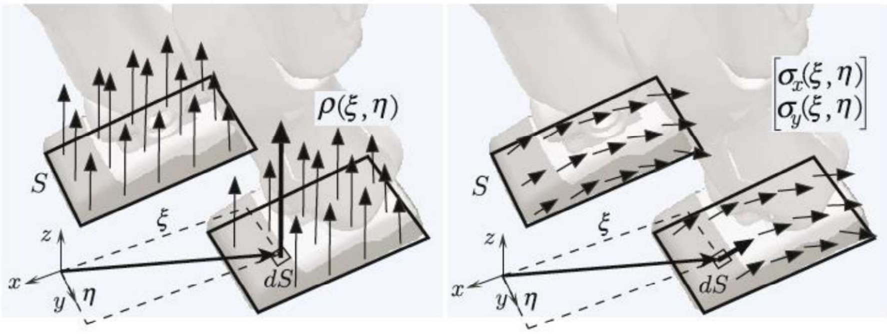
$$\tau_{nx} = 0$$

$$\tau_{ny} = 0$$

$$p_x = \frac{\int_S \xi\rho(\xi,\eta)dS}{\int_S \rho(\xi,\eta)dS}$$

$$p_y = \frac{\int_S \eta\rho(\xi,\eta)dS}{\int_S \rho(\xi,\eta)dS}.$$

ZMP - 3D case



(a) Vertical reaction forces (b) Horizontal reaction forces

horizontal component of the GRF

$$f_x = \int_S \sigma_x(\xi,\eta)dS$$

$$f_y = \int_S \sigma_y(\xi,\eta)dS.$$

$$\boldsymbol{\tau}_t(\boldsymbol{p}) \equiv [\tau_{tx}\ \tau_{ty}\ \tau_{tz}]^T$$

$$\tau_{tx} = 0$$

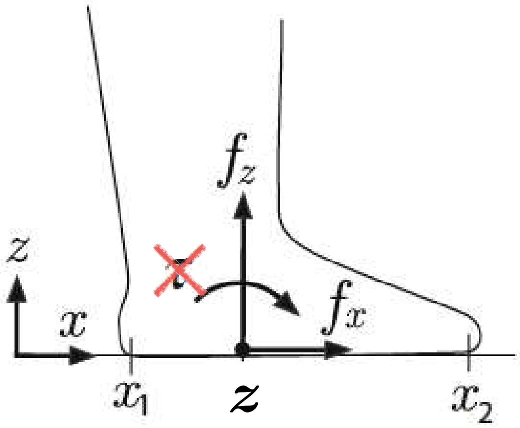
$$\tau_{ty} = 0$$

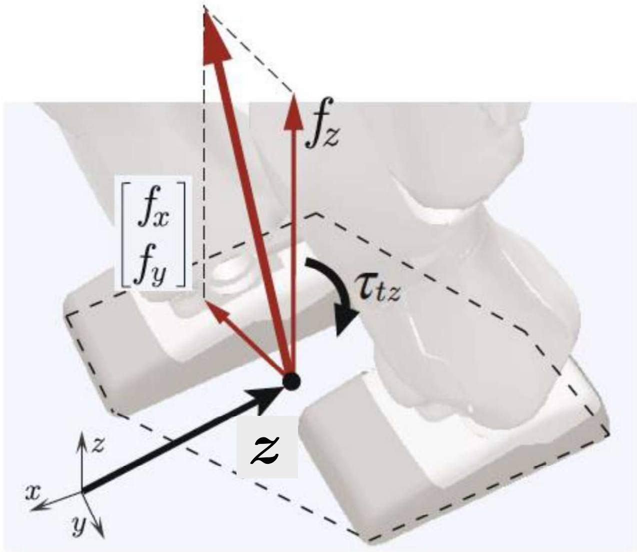
$$\tau_{tz} = \int_S \{(\xi - p_x)\sigma_y(\xi,\eta) - (\eta - p_y)\sigma_x(\xi,\eta)\}dS$$

$$\boldsymbol{\tau}_p = \boldsymbol{\tau}_n(\boldsymbol{p}) + \boldsymbol{\tau}_t(\boldsymbol{p}) = [0\ 0\ \tau_{tz}]^T,$$

if robot moves, the z component will be different from 0

ZMP

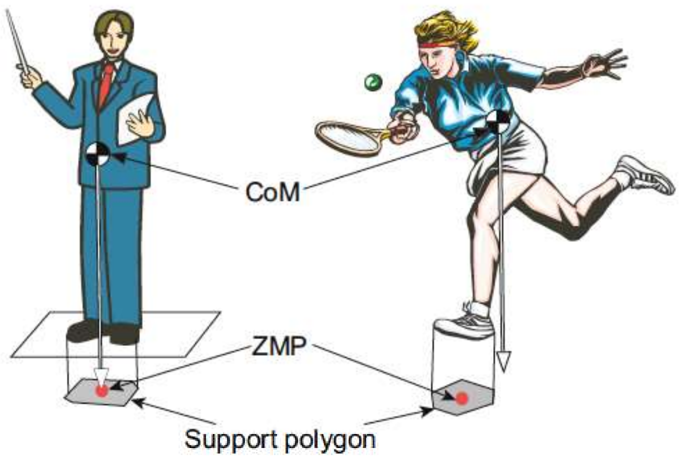




as long as the ZMP is in the Support Polygon,

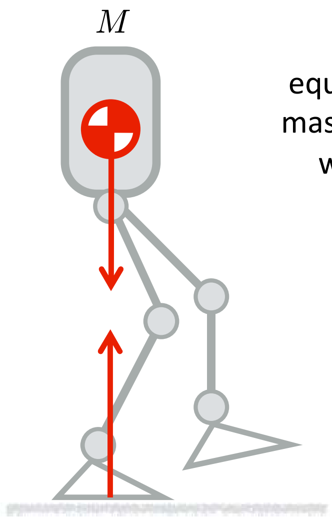
the support foot will not rotate

ZMP



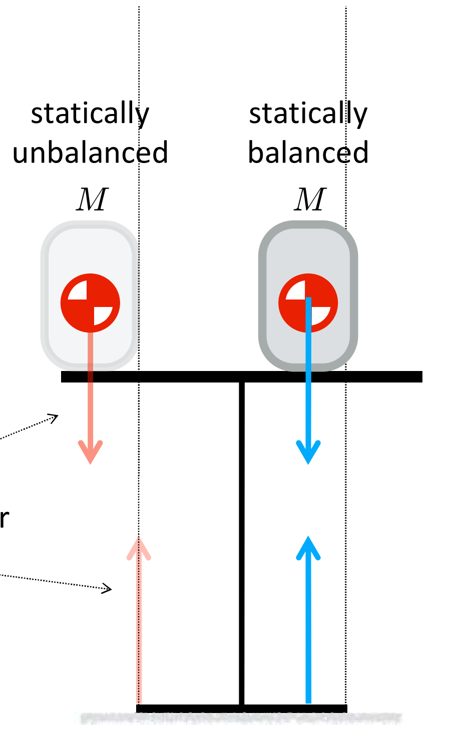
static balance

humanoid motionless:
statically balanced robots keep the center of mass within the polygon of support in order to maintain postural stability (sufficient when the robot moves slow enough so all the inertial forces are negligible)



equivalent representation:
mass M on a massless table
with finite length base

the table starts tipping over

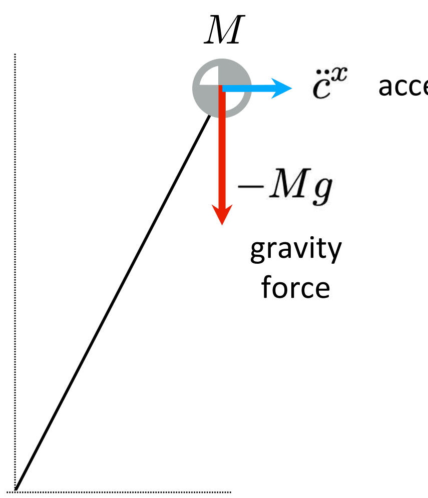


dynamic balance

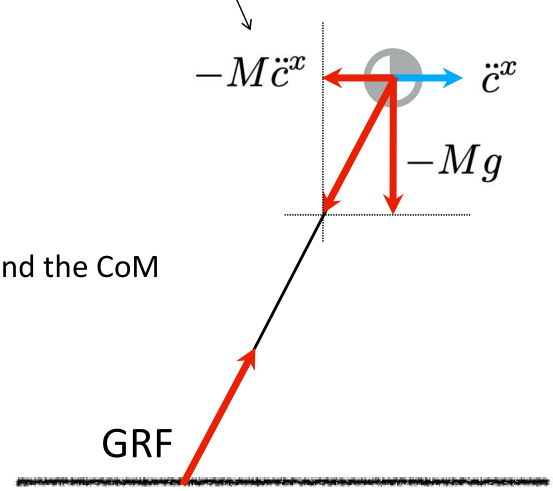
Question:
how do you keep a pendulum
in a non-vertical position?

Answer:
by continuously accelerating it

non-inertial frame
(pendulum stands still in an accelerating frame) → inertial force (fictitious force)



Hyp: no torque around the CoM



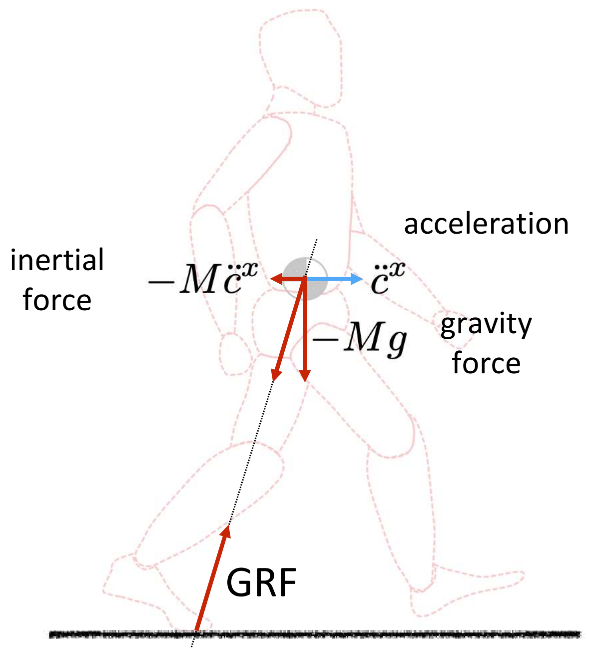
dynamic balance

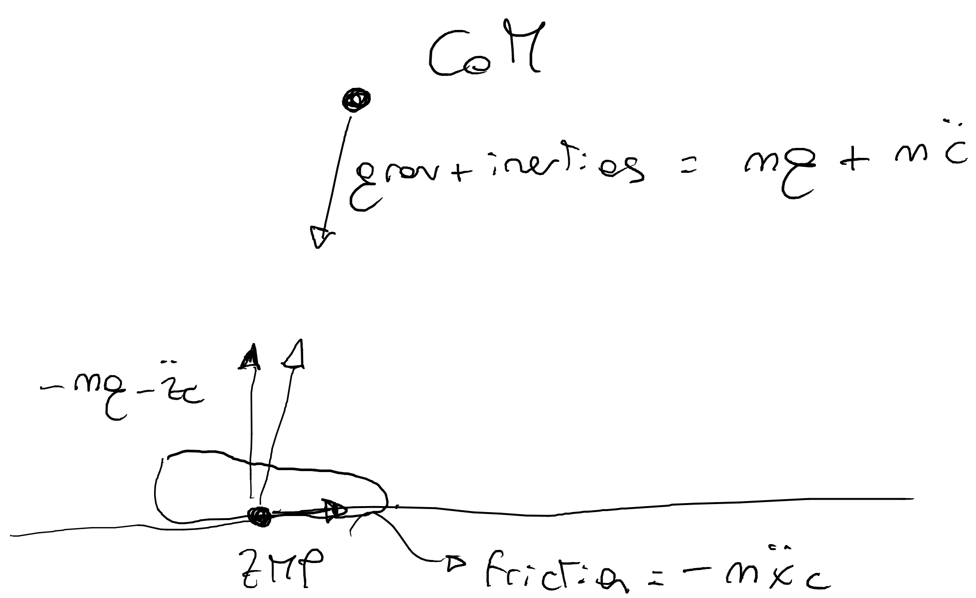
humanoid walking:

the GRF will also have a component parallel to the ground;

the motion requires the exchange of horizontal frictional force with the ground

hyp: no torque around the CoM



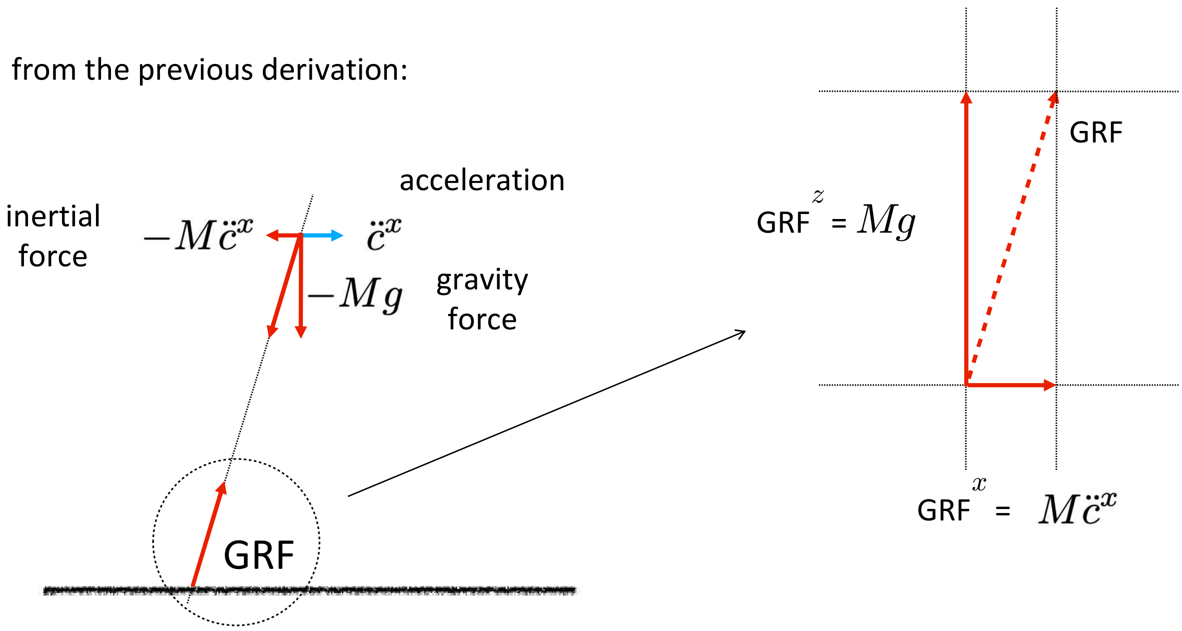


dynamic balance

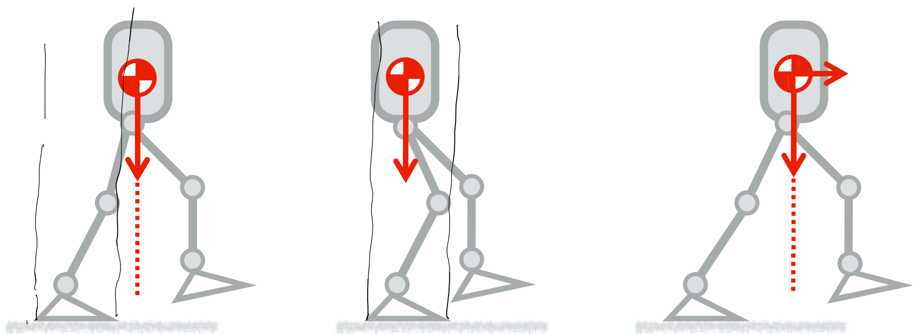
Ground Reaction Force (GRF): 2D components (x,z)

hyp: no torque around the CoM

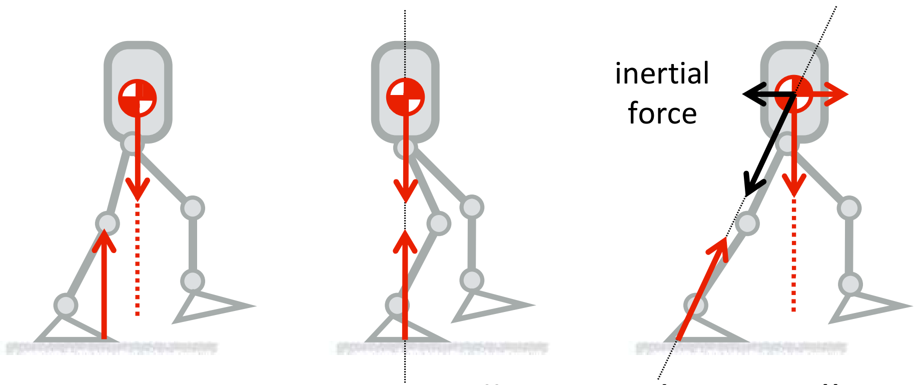
from the previous derivation:



which robot falls down?



which robot falls down?



falls

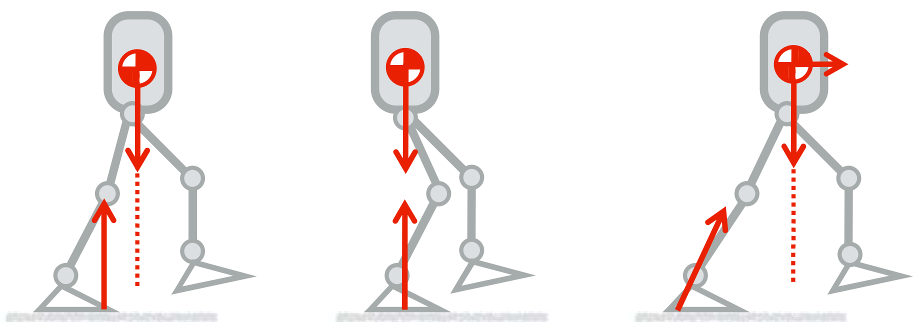
statically
balanced

dynamically
balanced

where is the ZMP?

z^x (ZMP): point on the ground where the GRF is applied

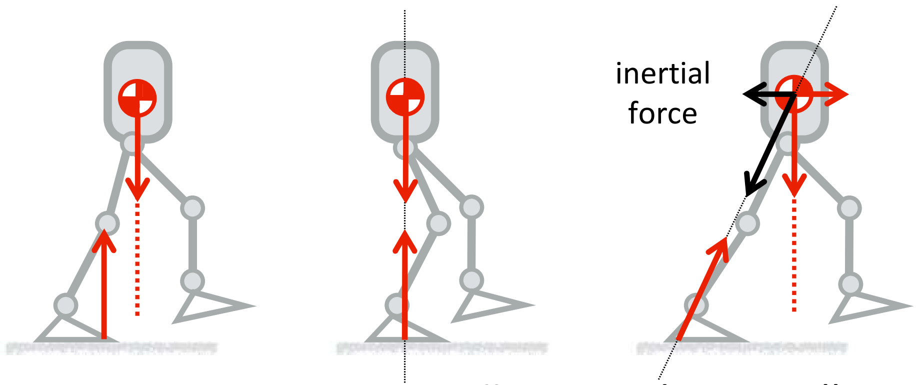
use the dynamics equation on horizontal flat ground and neglect $\dot{\boldsymbol{L}}^{x,y}$



$$\frac{c^z}{\ddot{c}^z + g^z}(\ddot{\boldsymbol{c}}^{x,y} + \cancel{\boldsymbol{g}^{x,y}}) = (\boldsymbol{c}^{x,y} - \boldsymbol{z}^{x,y}) + \cancel{\frac{\boldsymbol{S}\dot{\boldsymbol{L}}^{x,y}}{M(\ddot{c}^z + g^z)}}$$

where is the ZMP?

hyp CoM at **constant height** $c^z = \text{constant}$ $\longrightarrow$ LIP equation in the sagittal plane



falls

statically balanced

$$z^x = c^x$$

dynamically balanced

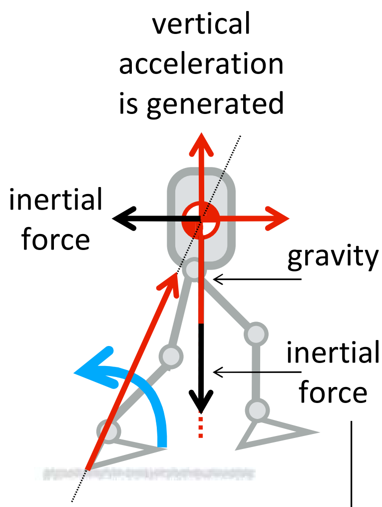
$$z^x = c^x - \frac{c^z}{g^z}\ddot{c}^x$$

what if CoM acceleration is increased?

from the previous analysis one could think that the ZMP, increasing the CoM acceleration, would leave the support foot support, but it doesn't

- once the ZMP has reached the foot border, a rotation starts around that point
- with the rotation of the foot, the center of mass starts accelerating vertically
- with a vertical acceleration of the CoM, its height does not remain constant
- model changes, ZMP remains constant

$$z^x = c^x - \frac{c^z}{\ddot{c}^z + g^z}\ddot{c}^x$$



the vertical CoM acceleration generates a vertical inertial force

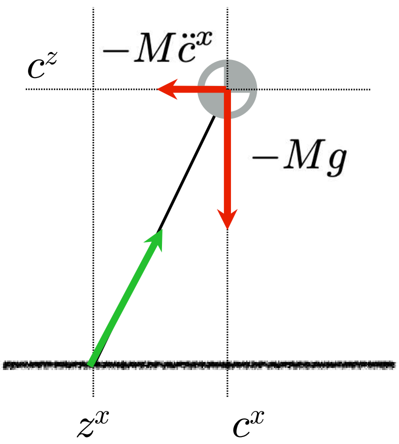
dynamically balanced

Hyp: no torque around the CoM

sum of moments around z^x

$$-Mg(c^x - z^x) + M\ddot{c}^x c^z = 0$$

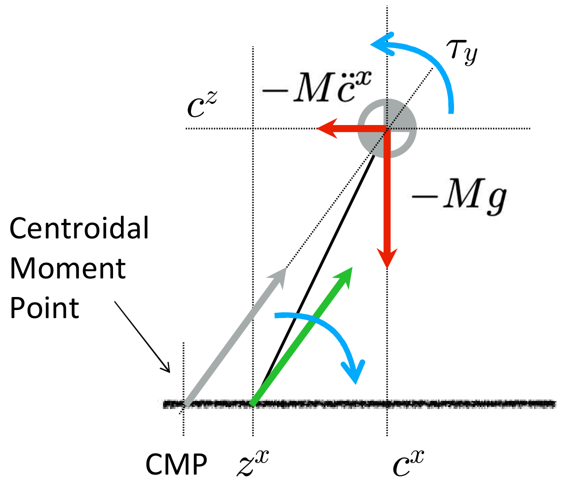
$$\longrightarrow \quad z^x = c^x - \frac{c^z}{g^z}\ddot{c}^x$$



+ torque τ_y around the CoM

$$-Mg(c^x - z^x) + M\ddot{c}^x c^z + \tau_y = 0$$

$$\longrightarrow \quad z^x + \frac{\tau_y}{Mg} = c^x - \frac{c^z}{g^z}\ddot{c}^x$$



dynamically balanced

+ torque τ_y around the CoM (or equivalently an ankle torque τ_y)

$$-Mg(c^x - z^x) + M\ddot{c}^x c^z + \tau_y = 0$$

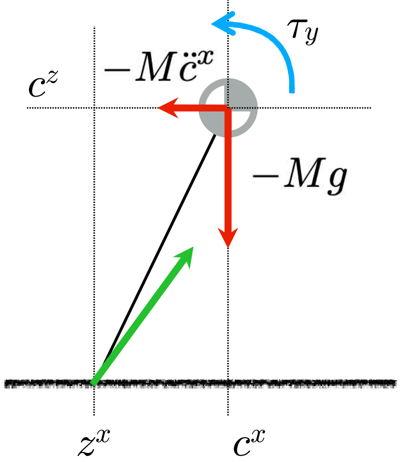
$$\longrightarrow \quad z^x + \frac{\tau_y}{Mg} = c^x - \frac{c^z}{g^z}\ddot{c}^x$$

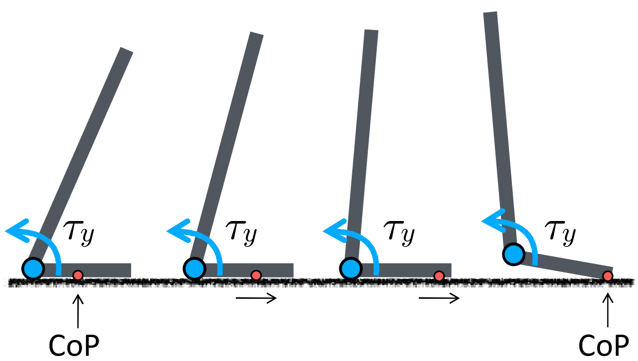
positive torque τ_y (counter-clockwise)

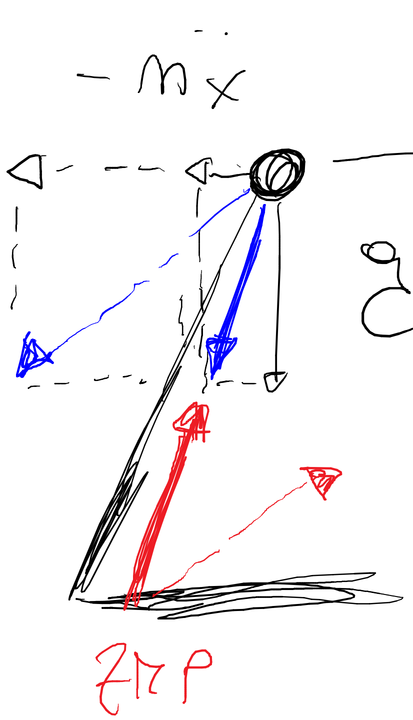
moves the Center of Pressure CoP to the right

z^x is not the CoP anymore

$$\text{CoP} \quad = z^x + \frac{\tau_y}{Mg}$$







dynamically balanced locomotion

generate a gait for walking while maintaining balance

maintaining balance	assumed to be equivalent to	**not tipping over the support foot**
		↓
ZMP needs to stay <u>inside</u> the Support Polygon	←	**ZMP-based criterion**

Autonomous and Mobile Robotics

Prof. Giuseppe Oriolo

Humanoid Robots 4:

Gait Generation

Dipartimento di Ingegneria Informatica
Automatica e Gestionale Antonio Ruberti

Sapienza
Università di Roma

gait generation

generate a gait for walking while maintaining balance (also called walking pattern generation)

balance: what is it? how do we guarantee it?

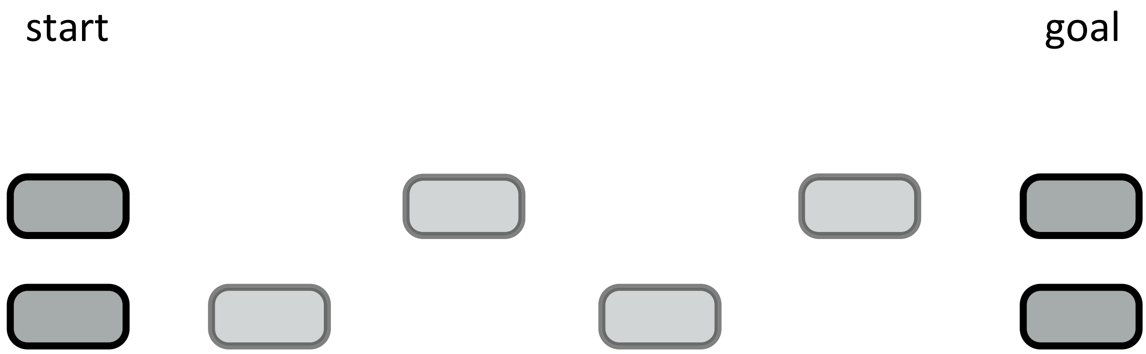
for this, different approaches available:

- nonlinear analysis
- Poincaré maps
- Viability
- Capturability
- Zero Moment Point ⟵ will use this (most popular)

ZMP-based gait generation

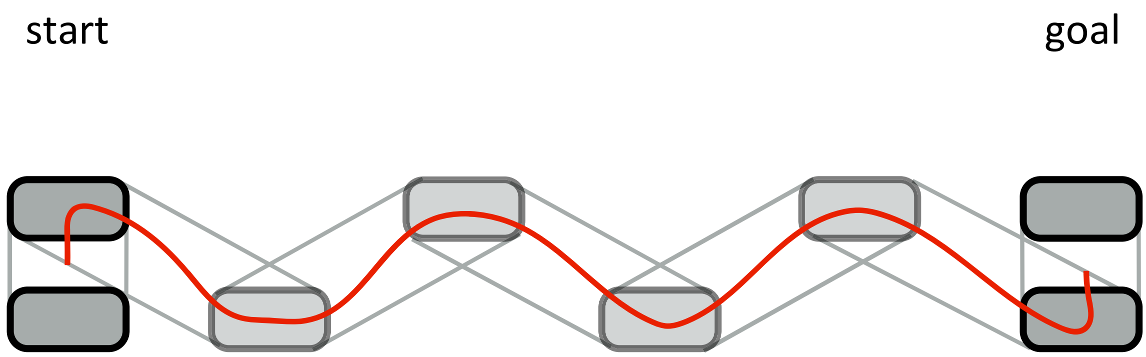
an algorithm based on the strategy: keep the ZMP inside the Support Polygon (SP)

1. plan the footsteps…



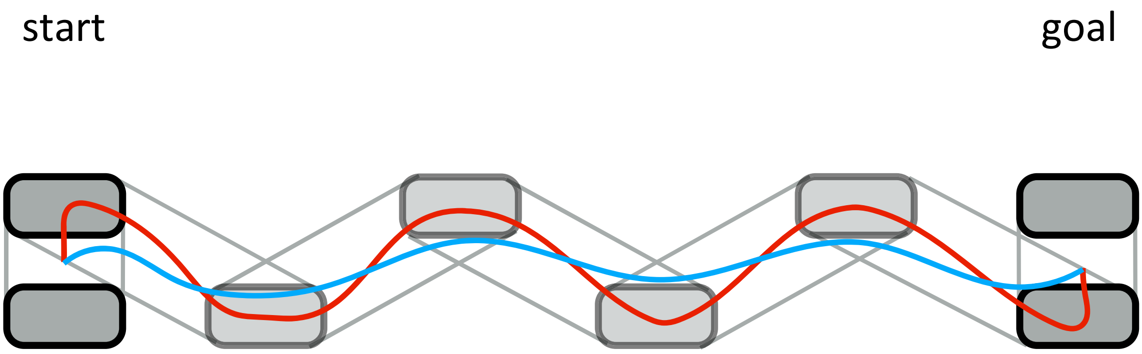
ZMP-based gait generation

…2. plan a ZMP trajectory such that is the ZMP is always inside the current SP…



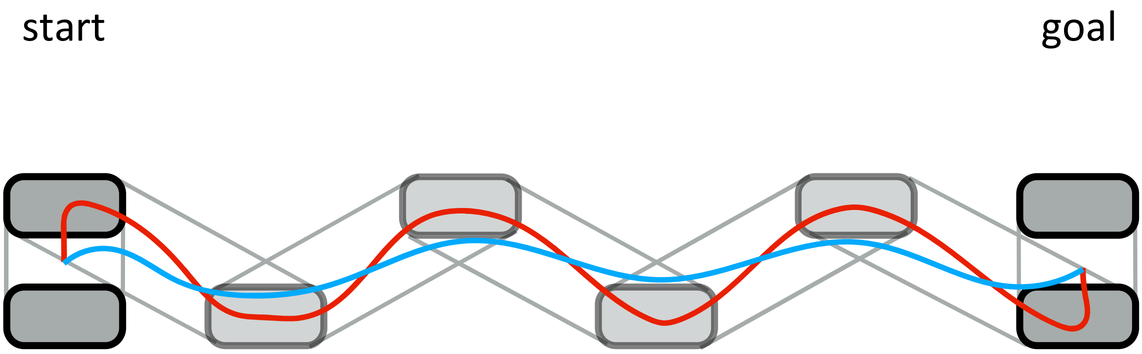
ZMP-based gait generation

…3. compute a CoM trajectory such that the ZMP moves as planned…



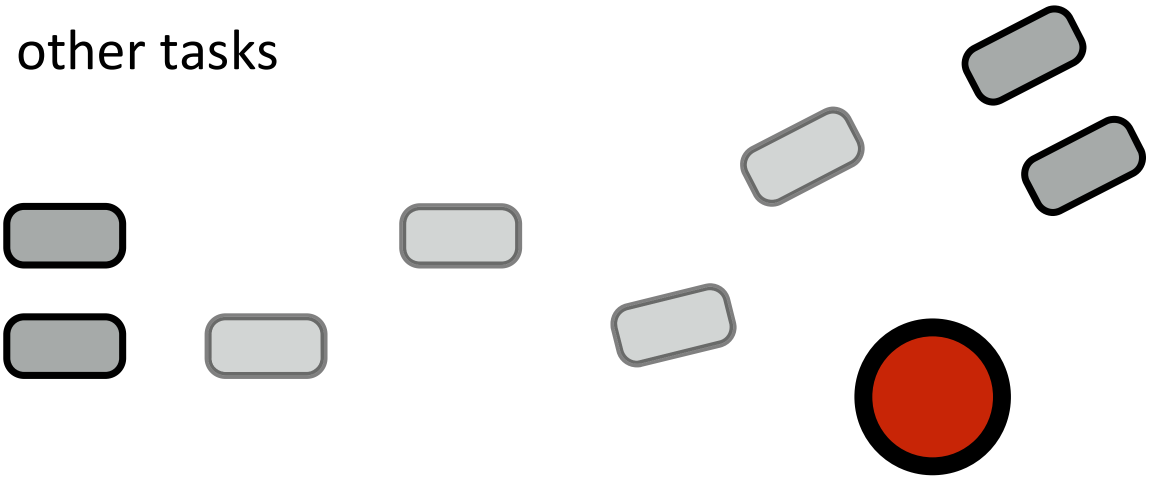
ZMP-based gait generation

…4. track the CoM trajectory

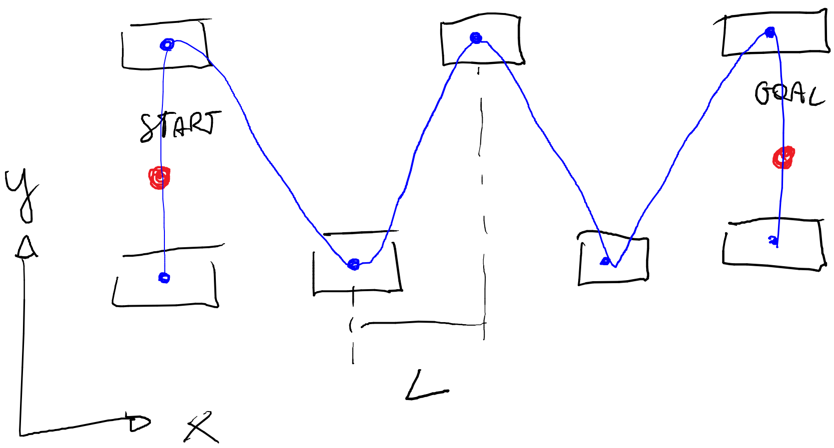


algorithm steps in detail

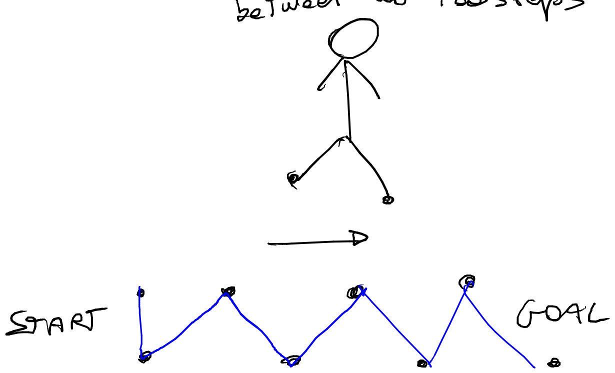
1 plan the footsteps (offline)
timing and lengths (desired speed)
obstacles (obstacle avoidance)
other tasks



— ZMP trajectory (center of each footstep)



if we move rapidly the support polygon is made up by the single line between two footsteps



$$\frac{L}{T} = v$$

algorithm steps in detail

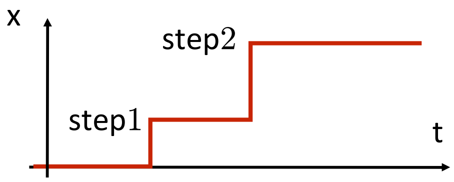
In order to remain in the support polygon:

→ In the Double support phase $x_{ref,ZMP}$ goes out of the feet which is going to move and it goes on the feet just landed on the ground

→ In the Single support phase $x_{ref,ZMP}$ rest under the same feet

2 plan ZMP trajectory (interpolation)

point foot (ZMP = point of contact)

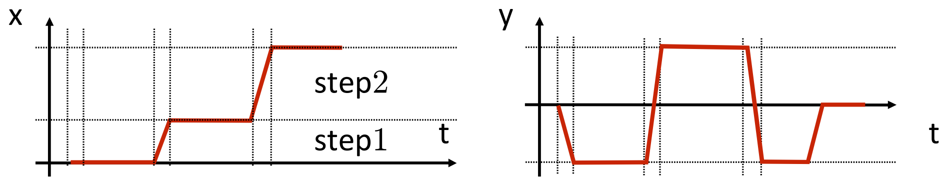


finite-sized foot

cycling through
SS and DS phases

↓ single support phase

↓ double support phase

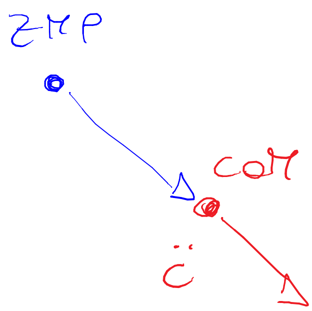


NEWTON - EULER EQS

CoM ↔ ZMP

1. Flat ground
2. Constant height of CoM
3. Intend angular momentum neglected

$$\ddot{c}^x = \frac{g}{c^z}\left(\underbrace{c^x}_{\text{CoM}} - \underbrace{z^x}_{\text{ZMP}}\right)$$



algorithm steps in detail

3 compute a (desired) CoM trajectory consistent with the planned ZMP trajectory

use the LIP model! e.g., for the sagittal direction

$$\ddot{c}^x = \frac{g^z}{c^z}\left(c^x - z^x\right)$$

planned ZMP trajectory

the CoM trajectory should be a solution of this 2nd order differential equation driven by the forcing term $z^x(t)$

potential instability problem! more on this later…

algorithm steps in detail

4 track the desired CoM trajectory

A. define a swinging foot trajectory
B. use kinematic control to obtain reference joint trajectories that realize the CoM and foot trajectories
C. send the reference joint profiles to the joint servos (for a position-controlled humanoid)

other approaches possible

instability problem: a control perspective

$$\ddot{c}^x = \frac{g^z}{c^z}\left(c^x - z^x\right) \qquad \omega^2 = \frac{g^z}{c^z}$$

pendulum frequency

ZMP ⟶ CoM $\quad \dfrac{c^x(s)}{z^x(s)} = \dfrac{\omega^2}{s^2 - \omega^2} \qquad \dfrac{\ddot{c}^x(s)}{z^x(s)} = \dfrac{\omega^2 s^2}{s^2 - \omega^2}$

inversion

$$\frac{z^x(s)}{j^x(s)} = \frac{s^2 - \omega^2}{s^3\omega^2} \qquad j^x(t) = \frac{d}{dt}\left(\ddot{c}^x(t)\right) \qquad \frac{z^x(s)}{\ddot{c}^x(s)} = \frac{s^2 - \omega^2}{s^2\omega^2}$$

jerk

CoM ⟶ ZMP

Instability problem

$$\ddot{c} = \underbrace{\frac{g}{c^z}}_{\omega^2}\left(c^x - z^x\right) \Rightarrow \ddot{c}^x = \omega^2\left(c^x - z^x\right)$$

2 eigenvalues

unstable ← ω $\quad$ $-\omega$ → stable

we want to invert this system in order to have the ZMP as input and the CoM as output

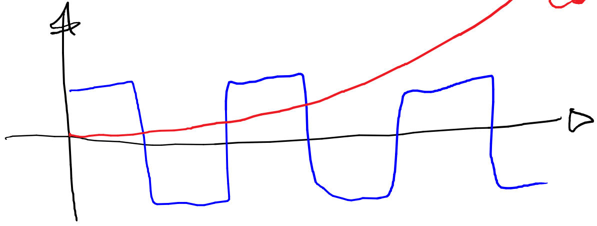
CoM → $\boxed{\text{SYS}}$ → ZMP

⇓

ZMP → $\boxed{\text{LIP}}$ → CoM

but the system is unstable!

CoM will diverge



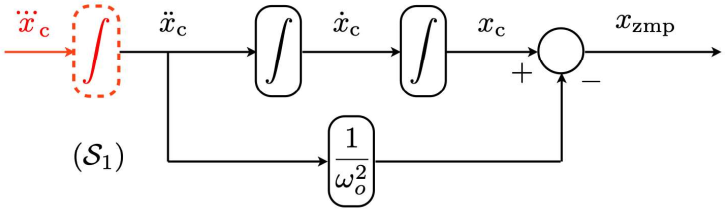
divergence problem: a control perspective

$$\ddot{c}^x = \frac{g^z}{c^z}\left(c^x - z^x\right) \qquad \omega^2 = \frac{g^z}{c^z}$$

reference ZMP

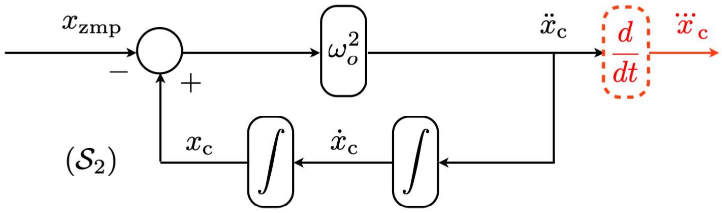
CoM ⟶ ZMP

output tracking problem



ZMP ⟶ CoM

stable inversion problem



Linear inverted pendulum (stable inversion)

$$\ddot{c} = \overbrace{\eta^2}^{\omega^2}(c - z) \quad \begin{cases} \to \eta \\ \to -\eta \end{cases}$$

$$c = x_u - \frac{\dot{c}}{\eta}$$

$$x_u = c + \frac{\dot{c}}{\eta} \;\to\; \dot{x}_u = \dot{c} + \frac{\ddot{c}}{\eta} \;\to\; \ddot{c} = \eta(\dot{x}_u - \dot{c})$$

substituting inside the original equation

$$\eta(\dot{x}_u - \dot{c}) = \eta^2(c - z)$$

$$\eta(\dot{x}_u - \cancel{\dot{c}}) = \eta^2\left(x_u - \cancel{\frac{\dot{c}}{\eta}} - z\right)$$

$$\boxed{\dot{x}_u = \eta(x_u - z)}$$

first order dynamics

unstable

$$x_s = c - \frac{\dot{c}}{\eta}$$

$$\to \boxed{\dot{x}_s = (-x_s + z)}$$

stable

$$\ddot{c} = \eta^2(c - z) \longrightarrow x_u = c + \frac{\dot{c}}{\eta} \longrightarrow \dot{x}_u = \eta(x_u - z)$$

$$x_u(t) = \underbrace{x_u(0)\, e^{\eta t}}_{\text{free evolution}} - \underbrace{\eta \int_0^t e^{\eta(t-\tau)} z(\tau)\, d\tau}_{\text{forced evolution}}$$

state evolution

multiplying each term by $e^{-\eta t}$

$$e^{-\eta t} e^{\eta(t-\tau)} = e^{-\eta\tau}$$

$$e^{-\eta t} x_u(t) = x_u(0) - \eta e^{-\eta t} \int_0^t e^{\eta(t-\tau)} z(\tau)\, d\tau$$

$$\cancel{e^{-\eta t}} x_u(t) = x_u(0) - \eta \int_0^t e^{-\eta\tau} z(\tau)\, d\tau$$

$t \to \infty$ x_u not diverge

$\to 0$ because $t \to \infty$

$$\boxed{x_u(0) = \eta \int_0^\infty e^{-\eta\tau} z(\tau)\, d\tau}$$

condition on the initial state for stability

stable inversion

- using a change of coordinates, the LIP can be decoupled in **stable** and **unstable** dynamics

$$x_s = c - \dot{c}/\eta$$
$$x_u = c + \dot{c}/\eta$$

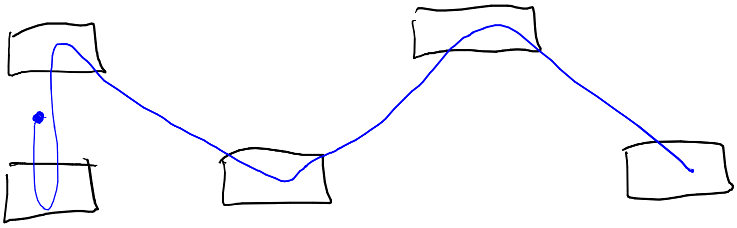
- the decoupled dynamics are

stable $\quad \dot{x}_s = \eta(-x_s + x_z)$

unstable $\quad \dot{x}_u = \eta(x_u - x_z)$

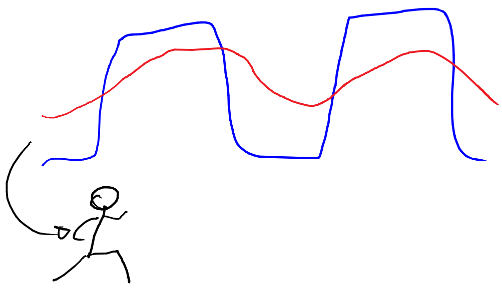
- the CoM evolution is bounded if and only if

$$x_u(t_0) = \eta \int_{t_0}^{\infty} e^{-\eta(\tau - t_0)} z(\tau)\, d\tau \qquad \textbf{stability condition}$$

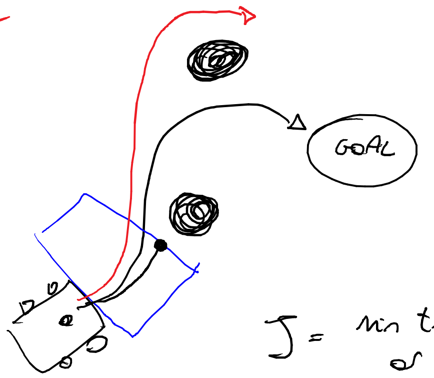


LIP model: $\ddot{c} = \eta^2 (c - z)$ → CoM not diverging!

$$c(t_0) + \frac{\dot{c}(t_0)}{\eta} = \int_0^{\infty} e^{-\eta \tau} z(\tau)\, d\tau$$



MPC → Model Predictive Control



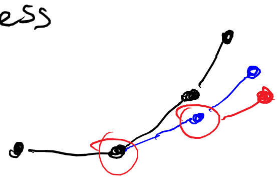
Optimal Control

controls → $\boxed{\text{SYSTEM}}$

J = cost function to minimize

J = min time or min fuel or min time to close to an obstacle

Long process with errors



The MPC uses (shifting) prediction horizon (—)

On each step MPC computes the next optimal step starting from the actual position

linear MPC

- example: regulation using linear **Model Predictive Control**

- linear prediction model

$$\begin{cases} x_{k+1} = Ax_k + Bu_k \\ y_k = Cx_k \end{cases}$$

$x \in \mathbb{R}^n$
$u \in \mathbb{R}^m$
$y \in \mathbb{R}^p$

- relationship between input and states

$$x_k = \underbrace{A^k x_0}_{\text{current state}} + \sum_{j=0}^{k-1} A^j B u_{k-1-j}$$

$x_{k+1} = Ax_k + Bu_k$

prediction $\underbrace{x_{k+1},\ x_{k+2},\ x_{k+3}, \dots x_{k+N}}$

$u_k,\ u_{k+1},\ u_{k+2}, \dots\ u_{k+N-1}$ input sequence

$x_{k+1} = Ax_k + Bu_k$

$x_{k+2} = Ax_{k+1} + Bu_{k+1} = A^2 x_k + ABu_k + Bu_{k+1}$

$x_{k+3} = Ax_{k+2} + Bu_{k+2} = A^3 x_k + A^2 Bu_k + ABu_{k+1} + Bu_{k+2}$

$\vdots$

$x_{k+N} = A^N x_k + A^{N-1} Bu_k + A^{N-2} Bu_{k+1} + \dots + Bu_{k+N-1}$

$$\underbrace{\begin{pmatrix} x_{k+1} \\ x_{k+2} \\ \vdots \\ \vdots \end{pmatrix}}_{X} = \underbrace{\begin{pmatrix} A \\ A^2 \\ A^3 \\ \vdots \end{pmatrix}}_{T} x_k + \underbrace{\begin{pmatrix} B & 0 & 0 & \dots \\ AB & B & 0 & \dots \\ A^2B & AB & B & \dots \\ \vdots & \dots & \dots & \dots \end{pmatrix}}_{S} \underbrace{\begin{pmatrix} u_k \\ u_{k+1} \\ \vdots \\ \vdots \end{pmatrix}}_{U}$$

$X = T x_k \quad S \quad U$

$$J = X^T Q X + U^T R U$$

$$= (\bar{S} U + \bar{T} x_k)^T Q (\bar{S} U + \bar{T} x_k) + U^T R U$$

linear MPC – cost function

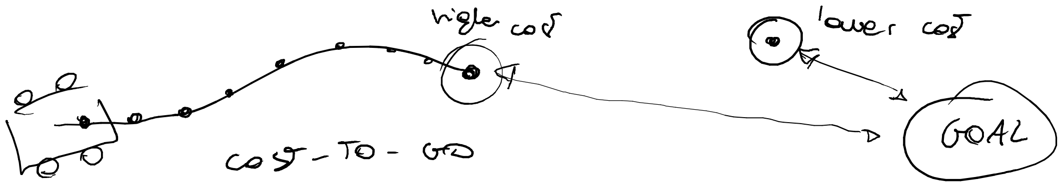
- cost function (quadratic)

x_N transposed → final state in the prediction horizon

$$J(z, x_0) = x'_N P x_N + \sum_{k=0}^{N-1} \left(x'_k Q x_k + u'_k R u_k \right) \qquad \begin{array}{lcl} R & = & R' \succ 0 \\ Q & = & Q' \succeq 0 \\ P & = & P' \succeq 0 \end{array} \qquad z = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

since we are doing a regulation we want to bring the state to zero

- goal: find a sequence $z^\star(t)$ that minimizes $J(z, x_0)$, i.e., that steers the state x to the origin optimally



linear MPC – cost function

- cost function

$$J(z, x_0) = x_0' Q x_0 + \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix}' \overbrace{\begin{bmatrix} Q & 0 & 0 & \dots & 0 \\ 0 & Q & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & Q & 0 \\ 0 & 0 & \dots & 0 & P \end{bmatrix}}^{\bar{Q}} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix}$$

$$+ \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}' \underbrace{\begin{bmatrix} R & 0 & \dots & 0 \\ 0 & R & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & R \end{bmatrix}}_{\bar{R}} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \underbrace{\begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \dots & B \end{bmatrix}}_{\bar{S}} \underbrace{\begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}}_{z} + \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}}_{\bar{T}} x_0$$

linear MPC – unconstrained case

- if the system is linear the cost function is **quadratic** in the input

$$\begin{aligned} J(z,x_0) &= (\bar{S}z+\bar{T}x_0)'\bar{Q}(\bar{S}z+\bar{T}x_0)+z'\bar{R}z+x_0'Qx_0 \\ &= \frac{1}{2}z'\underbrace{2(\bar{R}+\bar{S}'\bar{Q}\bar{S})}_{H}z+x_0'\underbrace{2\bar{T}'\bar{Q}\bar{S}}_{F'}z+\frac{1}{2}x_0'\underbrace{2(Q+\bar{T}'\bar{Q}\bar{T})}_{Y}x_0 \end{aligned} \qquad z=\begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

$$\Downarrow$$

$$J(z,x_0)=\frac{1}{2}z'Hz+x_0'F'z+\frac{1}{2}x_0'Yx_0$$

- without constraints, optimal solution by zeroing the gradient

$$\nabla_z J(z,x_0)=Hz+Fx_0=0 \quad \Rightarrow \quad z^*=\begin{bmatrix} u_0^* \\ u_1^* \\ \vdots \\ u_{N-1}^* \end{bmatrix}=-H^{-1}Fx_0$$

linear MPC – constraints

- add output and input **constraints**

$$\begin{cases} u_{\min} \le u(t) \le u_{\max} \\ y_{\min} \le y(t) \le y_{\max} \end{cases}$$

- what results is a **constrained optimal control problem**

$$\min_z \quad x_N'Px_N+\sum_{k=0}^{N-1}x_k'Qx_k+u_k'Ru_k$$

$$\begin{aligned} \text{s.t.} \quad & u_{\min} \le u_k \le u_{\max},\ k=0,\ldots,N-1 \\ & y_{\min} \le y_k \le y_{\max},\ k=1,\ldots,N \end{aligned}$$

$$y_{min} \le y \le y_{max}$$

$$u_{min} \le u \le u_{max}$$

$$u \le u_{max}$$

$$Az \le b$$

$$I \quad \begin{pmatrix} u_{max} \\ \vdots \\ u_{max} \end{pmatrix}$$

$$z = \begin{pmatrix} u_k \\ u_{k+1} \\ \vdots \\ u_{k+N-1} \end{pmatrix}$$

linear MPC – constraints

- we want to formulate this as a standard convex **Quadratic Program (QP)** (Numerical!)

$$V(x_0) = \tfrac{1}{2}x_0'Yx_0 + \min_{z} \quad \tfrac{1}{2}z'Hz + x_0'F'z \qquad \text{(quadratic objective)}$$

$$\text{s.t.} \quad Gz \le W + Sx_0 \qquad \text{(linear constraints)}$$

- we already found the cost function: how do we write the constraints?

linear MPC – constraints

- input constraints

$u_{min} \leq u \leq u_{max}$

I

$$\begin{cases} u_k & \leq & u_{\max} \\ -u_k & \leq & -u_{\min} \end{cases} \Rightarrow \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & 1 \\ -1 & 0 & \dots & 0 \\ 0 & -1 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & -1 \end{bmatrix} z \leq \begin{bmatrix} u_{\max} \\ u_{\max} \\ \vdots \\ u_{\max} \\ -u_{\min} \\ -u_{\min} \\ \vdots \\ -u_{\min} \end{bmatrix} \qquad z = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

$-I$

linear MPC – constraints

- we write the output in terms of predicted inputs using

$$y_k = CA^k x_0 + \sum_{i=0}^{k-1} CA^i B u_{k-1-i} \leq y_{\max}, \; k = 1, \dots, N$$

- the upper output constraint is

$$\begin{bmatrix} CB & 0 & \dots & 0 \\ CAB & CB & \dots & 0 \\ \vdots & & & \vdots \\ CA^{N-1}B & \dots & CAB & CB \end{bmatrix} z \leq \begin{bmatrix} y_{\max} \\ y_{\max} \\ \vdots \\ y_{\max} \end{bmatrix} - \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^N \end{bmatrix} x_0$$

and the lower output constraint is analogous

linear MPC – algorithm

at each step:

- measure or estimate the **current state**
- compute the **prediction** (optimal control sequence) by solving the QP starting from the current state
- apply only the **first input** of the predicted sequence

Plan footsteps (support polygon)

ZMP $\in$ supp. polygon

LIP model $\rightarrow$ CoM trajectory

MPC $\left\{ \begin{array}{l} \text{prediction horizon} \\ \text{minimize cost} \\ \text{subject to constraints} \end{array} \right\}$ every iteration

preview control

ZMP ⟶ $\boxed{\text{LIP}}$ ⟶ CoM

CoM ⟶ $\boxed{\text{CT}}$ ⟶ ZMP ∈ SP ⟶ constrained

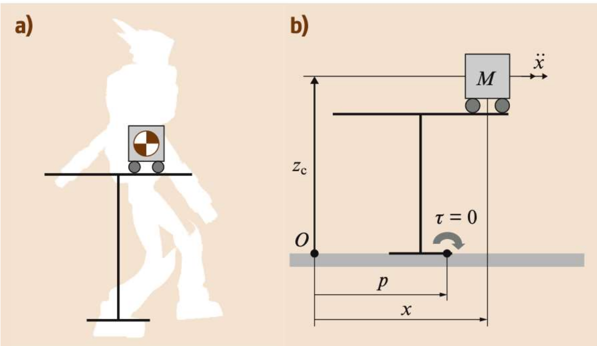
⟶ cart table model

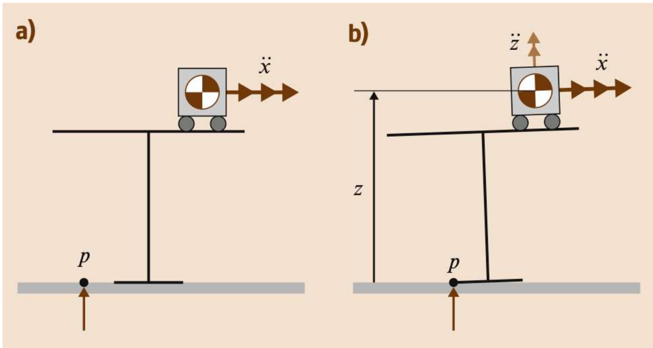
CoM trajectory generation can be seen as the

design of a ZMP tracking controller

CoM ⟶ ZMP

$$\frac{z^x(s)}{j^x(s)} = \frac{s^2 - \omega^2}{s^3 \omega^2}$$





$\ddot{c} = \eta^2 (c - z)$

$\dddot{c}$ jerk ⟶ $\ddot{c}$ ⟶ $\dot{c}$ ⟶ c

input of MPC

$z = c - \frac{\ddot{c}}{\eta^2}$ output

preview control

- Cart-Table (CT) model: the state of the system is defined by position, velocity and acceleration of the CoM

$$\boldsymbol{x} = [\boldsymbol{c}, \quad \dot{\boldsymbol{c}}, \quad \ddot{\boldsymbol{c}}]^T$$

- the control input is the jerk of the CoM

$$\boldsymbol{u} = \dddot{\boldsymbol{c}}$$

- the state dynamics is a triple integrator

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} \boldsymbol{c} \\ \dot{\boldsymbol{c}} \\ \ddot{\boldsymbol{c}} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} \boldsymbol{c} \\ \dot{\boldsymbol{c}} \\ \ddot{\boldsymbol{c}} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}\boldsymbol{u}$$

- the output is the ZMP

$$\boldsymbol{z} = \begin{pmatrix} 1 & 0 & -c^z/g \end{pmatrix}\begin{pmatrix} \boldsymbol{c} \\ \dot{\boldsymbol{c}} \\ \ddot{\boldsymbol{c}} \end{pmatrix}$$

preview control

- by discretizing we obtain the dynamical system

$$\begin{cases} \boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{C}\boldsymbol{u}_k \\ \boldsymbol{z}_k = \boldsymbol{C}\boldsymbol{x}_k \end{cases}$$

where

$$\boldsymbol{A} = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \quad \boldsymbol{B} = \begin{bmatrix} T^3/6 \\ T^2/2 \\ T \end{bmatrix}$$

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & -\frac{c^z}{g} \end{bmatrix}$$

preview control

- if the ZMP reference can be previewed for N_L future steps, the controller can be written as

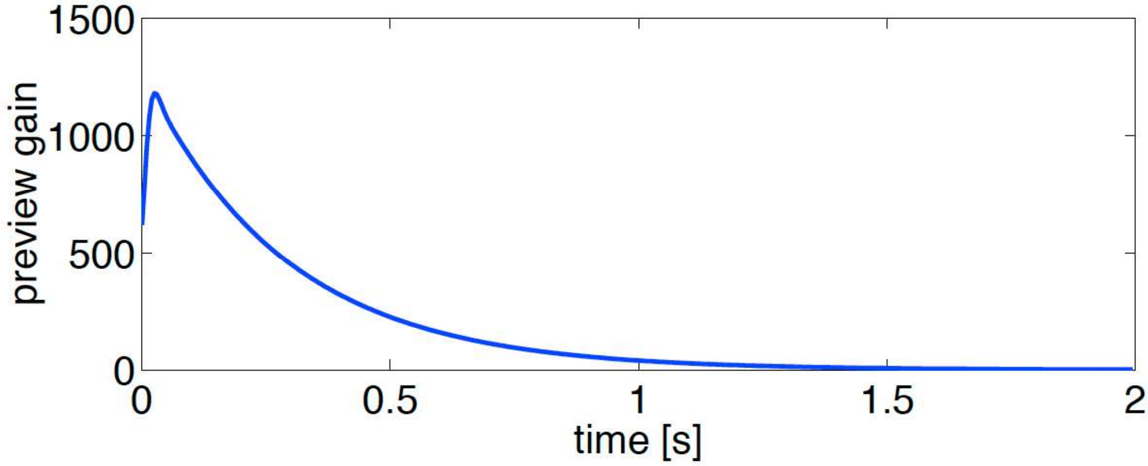
$$\boldsymbol{u}_k = \underbrace{-G_i \sum_{i=0}^{k} \boldsymbol{e}_k}_{\text{integral action on the tracking error}} \underbrace{- G_x \boldsymbol{x}_k}_{\text{state feedback}} \underbrace{- \sum_{j=1}^{N_L} G_p(j)\boldsymbol{z}^{ref}(k+j)}_{\text{preview action using future ZMP references}}$$

- this is the control law that minimizes the following cost function (unconstrained MPC)

$$J = \sum_{i=k}^{\infty} \{Q_e \boldsymbol{e}_i^2 + \Delta\boldsymbol{x}_i^T Q_x \Delta\boldsymbol{x}_i + R\Delta\boldsymbol{u}_i^2\}$$

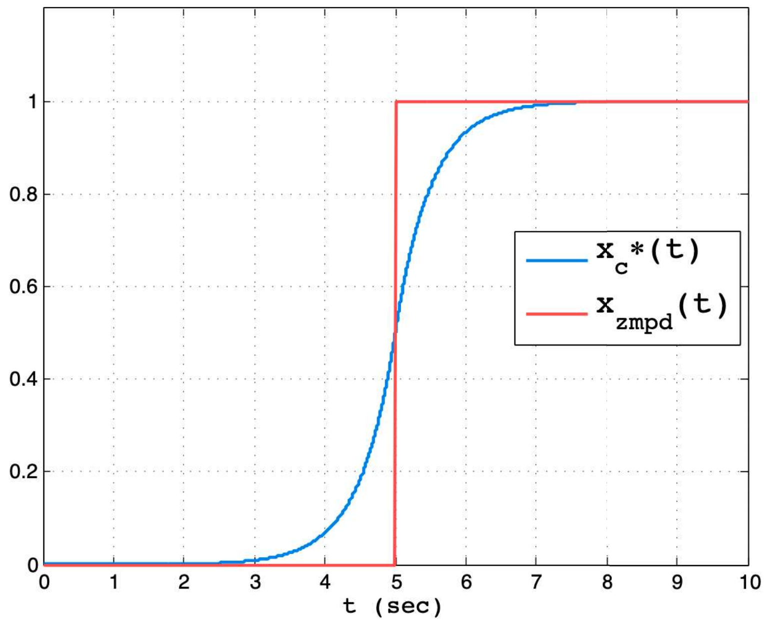
$$\boldsymbol{e}_k = \boldsymbol{z}_k - \boldsymbol{z}^{ref} \qquad \Delta\boldsymbol{x}_k = \boldsymbol{x}_k - \boldsymbol{x}_{k-1} \qquad \Delta\boldsymbol{u}_k = \boldsymbol{u}_k - \boldsymbol{u}_{k-1}$$

preview control



the magnitude of the preview gain quickly becomes very small; hence, the controller does not need information on far future

preview control



MPC gait generation

same formulation of the preview controller

- CoM jerk is the input $\boldsymbol{u} = \dddot{\boldsymbol{c}}$

- state is CoM position, velocity and acceleration

$$\boldsymbol{x} = [\boldsymbol{c}, \quad \dot{\boldsymbol{c}}, \quad \ddot{\boldsymbol{c}}]^T$$

- the state dynamics is a triple integrator

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} \boldsymbol{c} \\ \dot{\boldsymbol{c}} \\ \ddot{\boldsymbol{c}} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} \boldsymbol{c} \\ \dot{\boldsymbol{c}} \\ \ddot{\boldsymbol{c}} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}\boldsymbol{u}$$

- the output is the ZMP

$$\boldsymbol{z} = \begin{pmatrix} 1 & 0 & -c^z/g \end{pmatrix}\begin{pmatrix} \boldsymbol{c} \\ \dot{\boldsymbol{c}} \\ \ddot{\boldsymbol{c}} \end{pmatrix}$$

MPC gait generation

Supp. polygon

- instead of tracking a ZMP reference, impose **ZMP constraints**

↳ ZMP ∈ region

$$\begin{pmatrix} z_{k+1}^{x,min} \\ z_{k+2}^{x,min} \\ \vdots \\ z_{k+N}^{x,min} \\ z_{k+1}^{y,min} \\ z_{k+2}^{y,min} \\ \vdots \\ z_{k+N}^{y,min} \end{pmatrix} \leq \begin{pmatrix} z_{k+1}^{x} \\ z_{k+2}^{x} \\ \vdots \\ z_{k+N}^{x} \\ z_{k+1}^{y} \\ z_{k+2}^{y} \\ \vdots \\ z_{k+N}^{y} \end{pmatrix} \leq \begin{pmatrix} z_{k+1}^{x,max} \\ z_{k+2}^{x,max} \\ \vdots \\ z_{k+N}^{x,max} \\ z_{k+1}^{y,max} \\ z_{k+2}^{y,max} \\ \vdots \\ z_{k+N}^{y,max} \end{pmatrix}$$

(simplified case:
in general x-y
are not decoupled)

- the cost function can just minimize the square of the input over the prediction horizon

$$J = \sum_{i=k}^{k+N-1}\left((\dddot{c}_i^x)^2 + (\dddot{c}_i^y)^2\right)$$

MPC gait generation

- the idea is to shift the the balance condition from the **cost function** (tracking a reference) to the **constraints** (ZMP in support polygon)

- more **guarantees** on the ZMP at the cost of a more **complex** controller (need to solve a constrained optimization at each step)

- good QP solvers can still guarantee **real-time** execution

MPC on the LIP model

- we can use the **Linear Inverted Pendulum (LIP)** as the prediction model of the MPC

$$\frac{d}{dt}\begin{pmatrix} c \\ \dot{c} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\eta^2 & 0 \end{pmatrix}\begin{pmatrix} c \\ \dot{c} \end{pmatrix} + \begin{pmatrix} 0 \\ \eta^2 \end{pmatrix} z$$

- the ZMP is now the **input**

- the LIP is **unstable**! how do we guarantee that we do not get a divergent CoM trajectory?

COST FUNCTION

$\sum_{\text{horizon}} \dddot{c}^2 \rightarrow$ stable

ZMP $\rightarrow \boxed{\text{LIP}} \rightarrow$ CoM

CoM $\rightarrow \boxed{\text{CT}} \rightarrow$ ZMP

$$\underbrace{x(t_0) = \eta \int_0^\infty e^{-\eta\tau} z(\tau)\, d\tau}_{\text{constraint}}$$

$\rightarrow$ prediction

MPC on the LIP model

- decompose the LIP in **stable** and **unstable** dynamics

$$x_s = c - \dot{c}/\eta$$
$$x_u = c + \dot{c}/\eta$$

- we want to impose the condition **at every MPC iteration**

$$x_u^k = \eta \int_{t_k}^{\infty} e^{-\eta(\tau - t_k)} z(\tau) d\tau \quad \Longrightarrow \quad \textbf{stability constraint}$$

the stability constraint

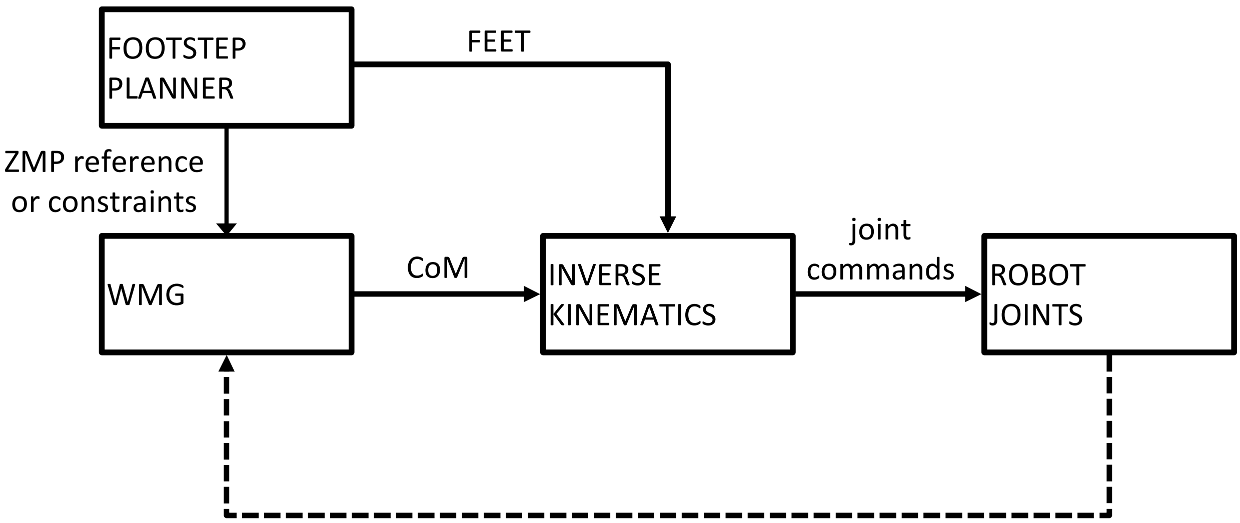
$$\text{current state} \longrightarrow x_u^k = \eta \int_{t_k}^{\infty} e^{-\eta(\tau - t_k)} z(\tau) d\tau \longleftarrow \text{predicted ZMP}$$

- the integral requires the predicted ZMP trajectory up to infinity, but MPC has a **finite** prediction horizon
- we **conjecture** the ZMP after the horizon (e.g., with the footstep plan)

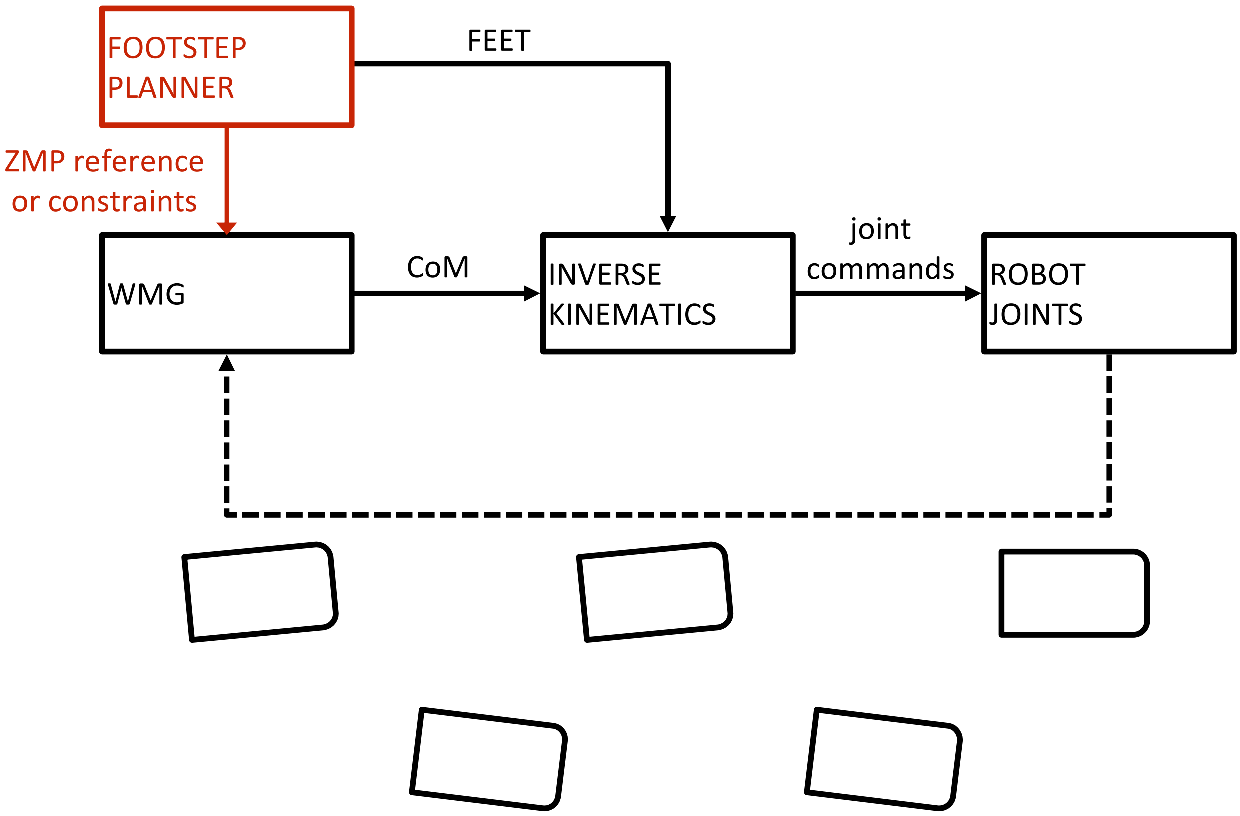
⇒ **recursive feasibility**

⇒ **stability**

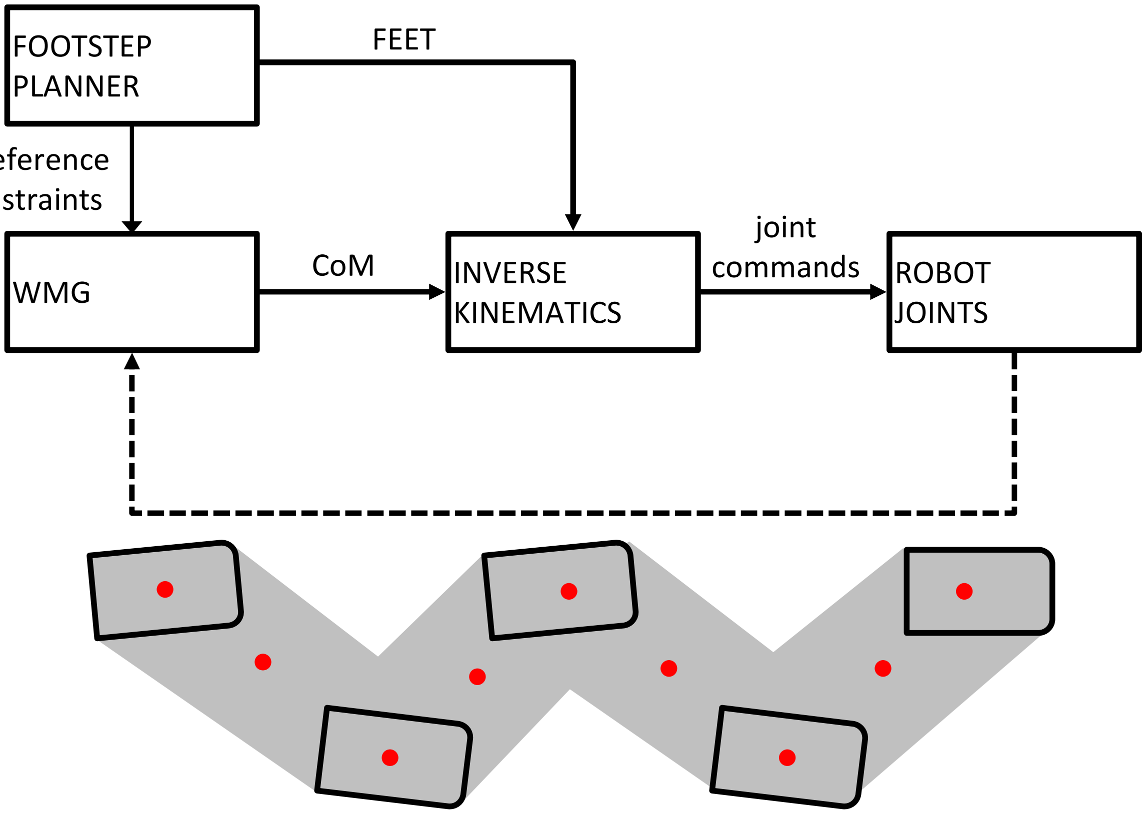
walking pattern generator (preview control)



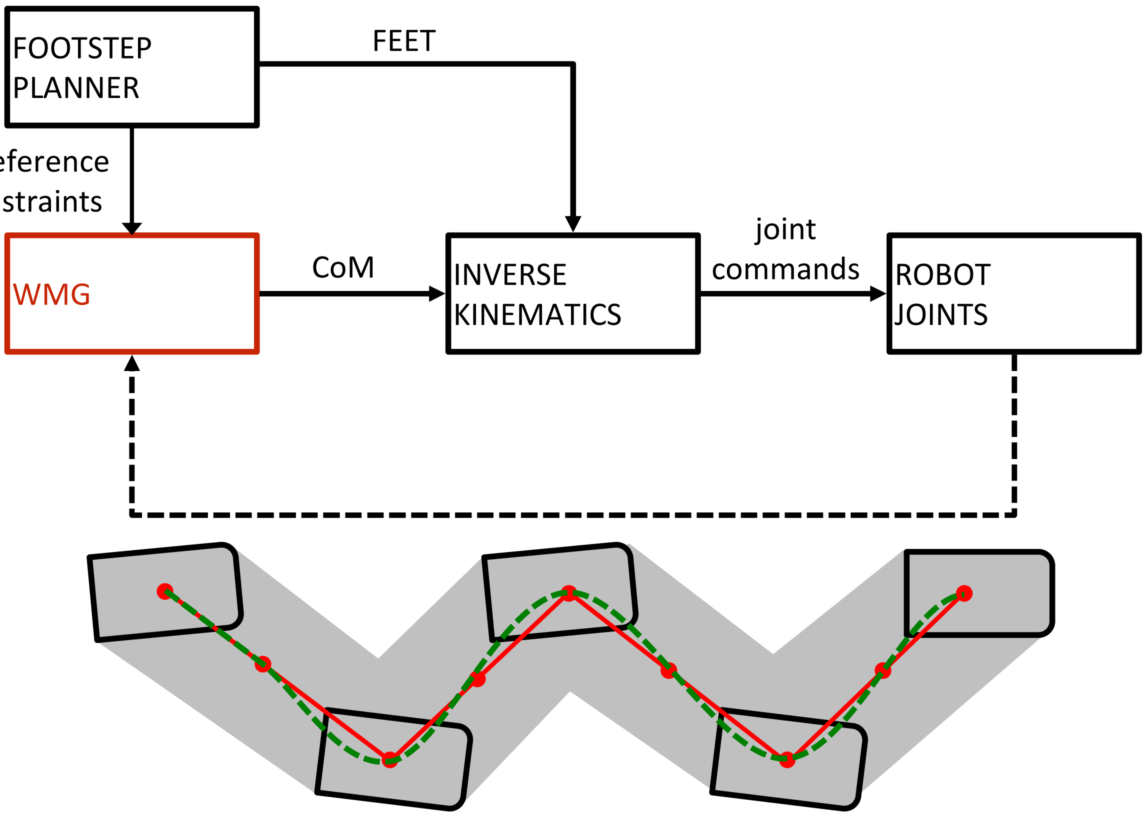
walking pattern generator



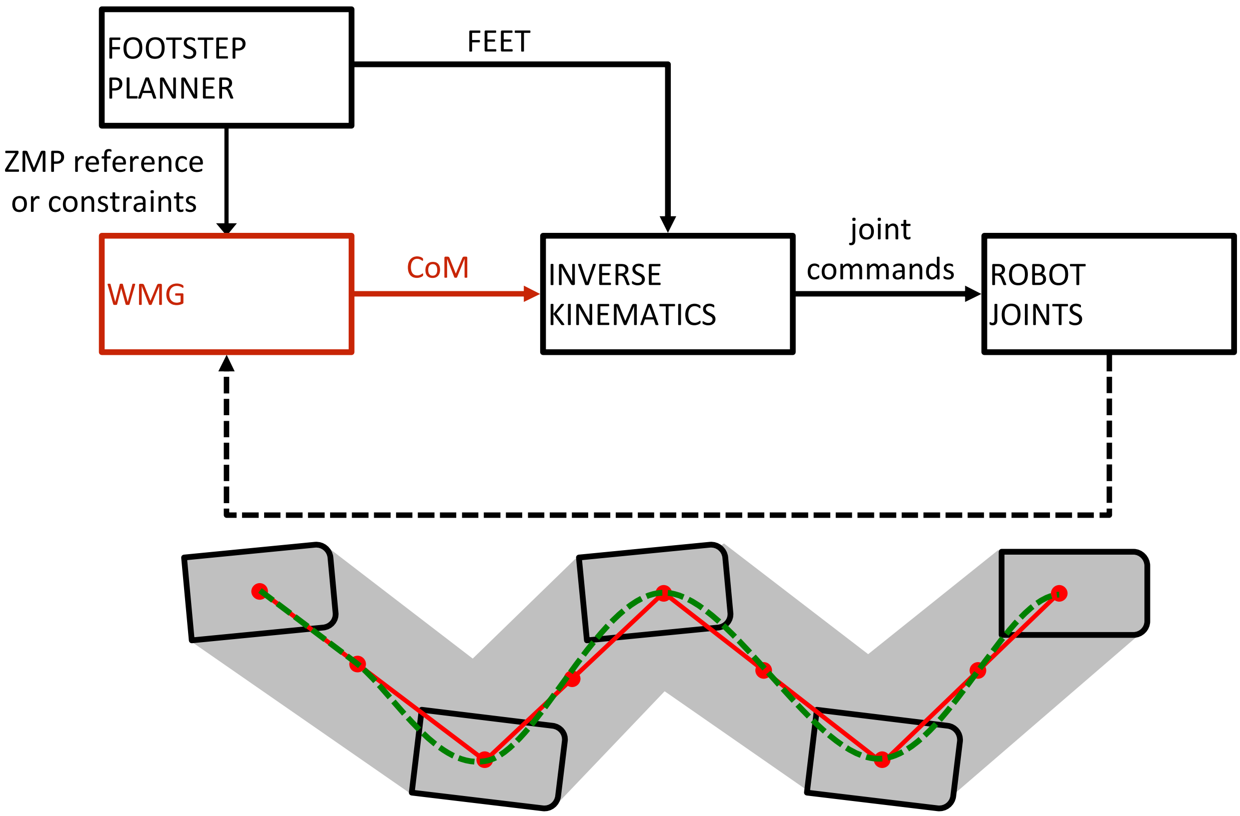
walking pattern generator



walking pattern generator



walking pattern generator



kinematic control

- how do we make the robot execute the desired CoM trajectory?
- simple solution: **kinematic tracking**

$$\boldsymbol{c} = \begin{pmatrix} c^x \\ c^y \\ c^z \end{pmatrix}, \quad \theta = \begin{pmatrix} \theta^x \\ \theta^y \\ \theta^z \end{pmatrix}$$

position of the CoM and orientation of the torso

$$\boldsymbol{f} = \begin{pmatrix} f^x \\ f^y \\ f^z \end{pmatrix}, \quad \phi = \begin{pmatrix} \phi^x \\ \phi^y \\ \phi^z \end{pmatrix}$$

position and orientation of the swing foot

everything is expressed wrt to the **current support foot**

kinematic control

- differential kinematics

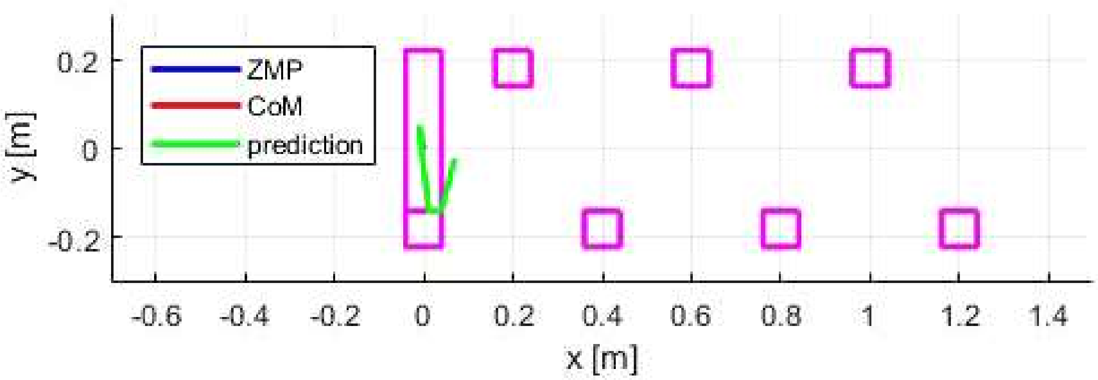
$$\dot{\boldsymbol{c}} = J_c(\boldsymbol{q})\dot{\boldsymbol{q}}$$
$$\dot{\boldsymbol{\theta}} = J_\theta(\boldsymbol{q})\dot{\boldsymbol{q}}$$
$$\dot{\boldsymbol{f}} = J_f(\boldsymbol{q})\dot{\boldsymbol{q}}$$
$$\dot{\boldsymbol{\phi}} = J_\phi(\boldsymbol{q})\dot{\boldsymbol{q}}$$

$$\begin{pmatrix} \dot{\boldsymbol{c}} \\ \dot{\boldsymbol{\theta}} \\ \dot{\boldsymbol{f}} \\ \dot{\boldsymbol{\phi}} \end{pmatrix} = \begin{pmatrix} J_c(\boldsymbol{q}) \\ J_\theta(\boldsymbol{q}) \\ J_f(\boldsymbol{q}) \\ J_\phi(\boldsymbol{q}) \end{pmatrix} \dot{\boldsymbol{q}}$$

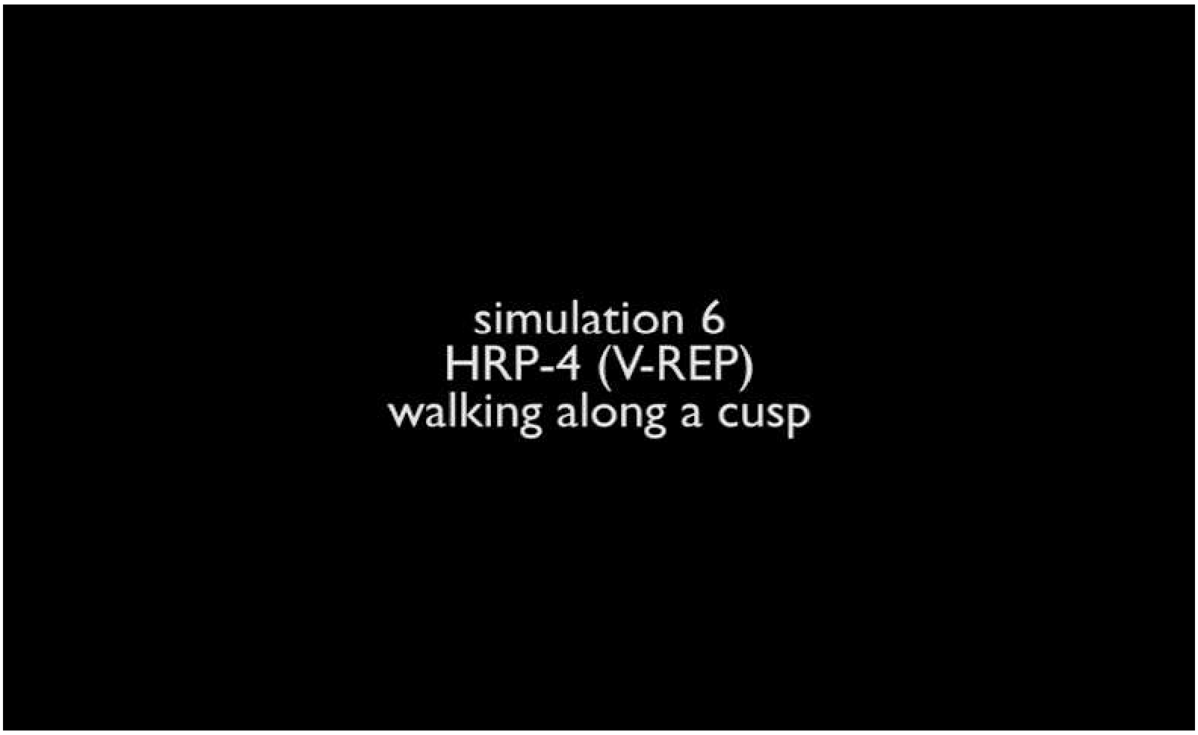
velocity task   stack of jacobians

- can be written as a **stack of tasks** $\quad \dot{\boldsymbol{t}} = J_t(q)\dot{\boldsymbol{q}}$
- the classic solution is the **pseudoinverse** $\quad \dot{\boldsymbol{q}} = J_t^{\#}(q)\dot{\boldsymbol{t}}_{des}$ open loop controller
- add a **position error** to avoid **drifting** $\quad \dot{\boldsymbol{q}} = J_t^{\#}(q)\left(\dot{\boldsymbol{t}}_{des} + k(\boldsymbol{t}_{des} - \boldsymbol{t})\right)$

examples – MPC gait generation



examples – simulations and experiments



other relevant topics

- **robust** gait generation (for disturbances)
- vertical CoM motion (for **uneven ground**)
- more accurate models (e.g., with **angular momentum**)
- **footstep planning** in complex environments

some more examples – robust gait generation

some more examples – uneven ground

Multimedia Contents

48. Modeling and Control of Legged Robots

Pierre-Brice Wieber, Russ Tedrake, Scott Kuindersma

The promise of legged robots over wheeled robots is to provide improved mobility over rough terrain. Unfortunately, this promise comes at the cost of a significant increase in complexity. We now have a good understanding of how to make legged robots walk and run dynamically, but further research is still necessary to make them walk and run efficiently in terms of energy, speed, reactivity, versatility, and robustness. In this chapter, we will discuss how legged robots are usually modeled, how their stability analysis is approached, how dynamic motions are generated and controlled, and finally summarize the current trends in trying to improve their performance. The main problem is avoiding to fall. This can prove difficult since legged robots have to rely entirely on available contact forces to do so. The temporality of leg motions appears to be a key aspect in this respect, as current control solutions include continuous anticipation of future motion (using some form of model predictive control), or focusing more specifically on limit cycles and orbital stability.

The promise of legged robots over standard wheeled robots is to provide improved mobility over rough terrain. This promise builds on the decoupling between the environment and the main body of the robot that the presence of articulated legs allows, with two consequences. First, the motion of the main body of the robot can be made largely independent from the roughness of the terrain, within the kinematic limits of the legs: legs provide an active suspension system. Indeed, one of the most advanced hexapod robots of the 1980s was aptly called the Adaptive Suspension Vehicle [48.1]. Second, this decoupling allows legs to temporarily leave their

contact with the ground: isolated footholds on a discontinuous terrain can be overcome, allowing to visit places absolutely out of reach otherwise. Note that having feet firmly planted on the ground is not mandatory here: skating is an equally interesting option, although rarely approached so far in robotics.
Unfortunately, this promise comes at the cost of a hindering increase in complexity. It is only with the unveiling of the Honda P2 humanoid robot in 1996 [48.2], and later of the Boston Dynamics BigDog quadruped robot in 2005 that legged robots finally began to deliver real-life capabilities that are just beginning to match the long sought animal-like mobility over rough terrain. Not matching yet the capabilities of humans and animals, legged robots do contribute however already to understanding their locomotion, as evidenced by the many fruitful collaborations between robotics and biomechanics researchers.

48.1 A Brief History of Legged Robots

Before the advent of digital computers, legged machines could be approached only by electromechanical means, lacking in advanced feedback control. This *pre-robotics* period culminated with the General Electric Walking Truck developed by Ralph Mosher, which inspired awe in the mid 1960s. The limb motions of this elephant-size quadruped machine were directly reflecting the limb motions of the onboard operator, who was responsible for all motion control and synchronization. Unfortunately, the strenuous concentration that this required limited operation to less than 15 minutes.

Digitally controlled legged robots started to appear in the late 1960s. Among early pioneers, Robert McGhee initiated a series of quadruped and hexapod robots first at University of South California, then at Ohio State University, culminating in the mid 1980s with the Adaptive Suspension Vehicle, a human carrying hexapod vehicle walking on natural and irregular outdoor terrain [48.1], while Ichiro Kato initiated a long series of biped and humanoid robots in the Waseda University, a series still continuing nearly half a century later [48.3]. But by the end of the 1970s, all legged robots were still limited to quasi-static gaits, i. e., slow walking motions with the center of mass of the robot always kept above its feet.

The transition to dynamic legged locomotion occurred in the beginning of the 1980s, with the first dynamically walking bipedal robot demonstrated at the Tokyo University [48.4], and the famous series of hopping and running monopedal, bipedal and quadrupedal robots developed at the MIT (Massachusetts Institute of Technology) LegLab under the direction of *Marc Raibert* [48.5]. Key theoretical breakthroughs came in the end of the 1980s, when *Tad McGeer* demonstrated that stable dynamic walking motions could be obtained by pure mechanical means, giving rise to a whole new field of research, *passive dynamic walking*, introducing new, key concepts such as orbital stability using Poincaré maps [48.6], with one simple conclusion: you need not have complete (or any) control to be able to walk dynamically and efficiently.

Legged robots were still mostly research laboratory curiosities working in limited situations when Honda unveiled the P2 humanoid robot in 1996 [48.2], a decade long secret project demonstrating unprecedented versatility and robustness, followed in 2000 by the Asimo humanoid robot. The world of humanoid and legged robots was ripe for companies to begin investing. A handful of other Japanese companies such as Toyota or Kawada were quick to follow with their own humanoid robots, while Sony began selling more than 150 000 of its Aibo home companion robot dogs. Boston Dynamics, a company *Marc Raibert* founded after leaving the MIT LegLab, finally unveiled its BigDog quadruped robot in 2005 [48.7], which was the first to demonstrate true animal-like locomotion capabilities on rough terrain.

The progress over the last decades has been remarkable. Profound questions have finally been answered: we now understand how to make legged robots walk and run dynamically. But other profound questions still have to be answered, such as how best to make them walk and run *efficiently*. The performance of legged robots needs to be improved in many ways: energy, speed, reactivity, versatility, robustness, etc. We will therefore discuss in this chapter how legged robots are usually modeled in Sect. 48.2 and how dynamic motions are currently generated and controlled in Sections 48.4 and 48.5, before discussing in Sect. 48.6 the current trends in improving their efficiency.

48.2 The Dynamics of Legged Locomotion

One of the major difficulties in making a legged robot walk or run is keeping its balance: where should the robot place its feet, how should it move its body in order to move safely in a given direction, even in case of strong perturbations? This difficulty comes from the fact that contact forces with the environment are an absolute necessity to generate and control locomotion, but they are limited by the mechanical laws of unilateral contact.

This essential role of the contact forces is particularly clear in the derivatives of the total linear and angular momenta of the robot, the former involving the motion of its center of mass. Because of the importance of contact forces for legged locomotion, we briefly discuss here their different models.

48.2.1 Lagrangian Dynamics

Structure of the Configuration Space

As for every robot moving in their 3-D environment (in space or underwater for example), the configuration space of legged robots combines the configuration $\hat{q} \in \mathbb{R}^N$ of their N joints with a global position $x_0 \in \mathbb{R}^3$ and orientation $\theta_0 \in \mathbb{R}^3$ (representing an element of $SO(3)$)

$$q = \begin{pmatrix} \hat{q} \\ x_0 \\ \theta_0 \end{pmatrix} . \tag{48.1}$$

The position x_0 and orientation θ_0 are typically those of a central body (pelvis or trunk) or of an extremity (foot or hand).

Structure of the Lagrangian Dynamics

The specific structure of the configuration space outlined above is naturally reflected in the Lagrangian dynamics

$$\mathbf{M}(q)\left[\begin{pmatrix} \ddot{\hat{q}} \\ \ddot{x}_0 \\ \ddot{\theta}_0 \end{pmatrix} + \begin{pmatrix} \mathbf{0} \\ g \\ \mathbf{0} \end{pmatrix}\right] + n(q,\dot{q}) = \begin{pmatrix} u \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} + \sum_i C_i(q)^{\mathrm{T}} f_i \tag{48.2}$$

of the system, where $\mathbf{M}(q) \in \mathbb{R}^{(N+6)\times(N+6)}$ is the generalized inertia matrix of the robot, $-g \in \mathbb{R}^3$ is the constant gravity acceleration vector, $n(q,\dot{q}) \in \mathbb{R}^{N+6}$ is the vector of Coriolis and centrifugal effects, $u \in \mathbb{R}^N$ is the vector of joint torques, and for all i, $f_i \in \mathbb{R}^3$ is a force exerted by the environment on the robot and $\mathbf{C}_i(q) \in \mathbb{R}^{(N+6)\times 3}$ is the associated Jacobian matrix [48.8].

Since the vector u of joint torques has the same size as the vector $\hat{q}$ of joint positions, the whole dynamics including the global position x_0 and orientation θ_0 appears to be underactuated if no external forces f_i are exerted.

48.2.2 Newton and Euler Equations of Motion

Center of Mass and Angular Momentum

A consequence of the structure (48.2) of the Lagrangian dynamics is that the part of this dynamics which is not directly actuated involves the Newton and Euler equations of motion of the robot taken as a whole ([48.8] for detailed derivations). The Newton equation can be written in the following way

$$m(\ddot{c} + g) = \sum_i f_i , \tag{48.3}$$

with m the total mass of the robot and c the position of its center of mass (COM). The Euler equation can be expressed with respect to the COM in the following way

$$\dot{\mathbf{L}} = \sum_i (p_i - c) \times f_i , \tag{48.4}$$

with p_i the points of applications of the forces f_i and

$$\mathbf{L} = \sum_k (x_k - c) \times m_k \dot{x}_k + \mathbf{I}_k \omega_k \tag{48.5}$$

the angular momentum of the whole robot with respect to its COM, with $\dot{x}_k$ and ω_k the translation and rotation velocities of the different parts k of the robot, m_k and $\mathbf{I}_k$ their masses and inertia tensor matrices (expressed in global coordinates).

The Newton equation makes it obvious that the robot needs external forces f_i in order to move its COM in a direction other than that of gravity. The Euler equation is more subtle, as we will see during flight phases.

Flight Phases

During flight phases, when a legged robot is not in contact with its environment, not experiencing any contact forces f_i, the Newton equation (48.3) simplifies to

$$\ddot{c} = -g . \tag{48.6}$$

In this case, the COM invariably accelerates along the gravity vector $-g$ with constant horizontal speed, following a standard falling motion: there is absolutely no possibility to control the COM to move in any different way. The Euler equation (48.4) simplifies in the same way to

$$\dot{\mathbf{L}} = 0 , \tag{48.7}$$

imposing a conservation of the angular momentum $\mathbf{L}$. In this case, however, the robot is still able to generate and control both joint motions and global rotations, this is how cats are able to fall back on their feet when

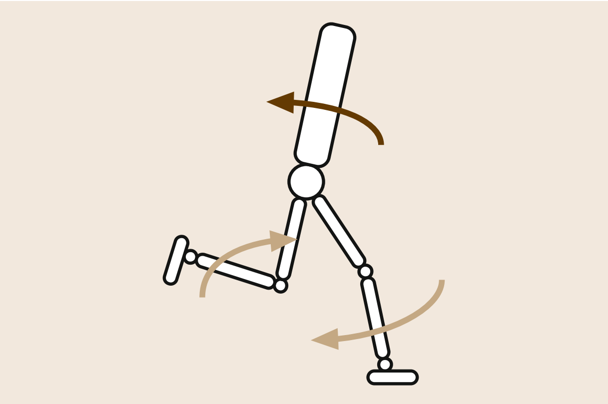


Fig. 48.1 Even though the angular momentum is constant during flight phases, the robot is still able to generate and control rotations of the whole body with the help of leg or arm motions (*light brown*), as a result of the *nonholonomy* of the angular momentum. This is how cats fall back on their feet when dropped from any initial orientation

dropped from any initial orientation (Fig. 48.1). This is a result of the *nonholonomy* of the angular momentum (48.5) which is not the derivative of any function of the configuration of the robot [48.9]. As a result, even though the angular momentum $\mathbf{L}$ is kept constant during the whole flight phase, the joint configuration $\hat{\boldsymbol{q}}$, and the global orientation $\boldsymbol{\theta}_0$ of the robot can be driven to any desired value at the end of the flight phase. We will see in Sect. 48.5 how this impacts the control of legged robots. Note that the dynamics of legged robots during flight phases is similar to the dynamics of free-floating space robots discussed in Chap. 55. Further discussion and developments can be found there.

In Contact with a Flat Ground: The Center of Pressure

In case the forces applied by the environment on the robot are due to contacts with a flat ground (while standing still, walking or running), let us consider a reference frame oriented along the ground, with the z axis orthogonal to it (therefore tilted if the ground is tilted, see Fig. 48.5). Without loss of generality, let us suppose that the points of contact, $\boldsymbol{p}_i$, with the ground are all such that $p_i^z = 0$.

Let us consider then the sum of the Euler (48.4) and the cross product of the COM $\boldsymbol{c}$ with the Newton (48.3)

$$m\boldsymbol{c} \times (\ddot{\boldsymbol{c}} + \boldsymbol{g}) + \dot{\mathbf{L}} = \sum_i \boldsymbol{p}_i \times \boldsymbol{f}_i\,, \tag{48.8}$$

and let us divide the result by the z coordinate of the Newton equation to obtain

$$\frac{m\boldsymbol{c} \times (\ddot{\boldsymbol{c}} + \boldsymbol{g}) + \dot{\mathbf{L}}}{m(\ddot{c}^z + g^z)} = \frac{\sum_i \boldsymbol{p}_i \times \boldsymbol{f}_i}{\sum_i f_i^z}\,. \tag{48.9}$$

Since $p_o^z = 0$, the x and y coordinates of this equation can be simplified in the following way

$$\boldsymbol{c}^{x,y} - \frac{c^z}{\ddot{c}^z + g^z}(\ddot{\boldsymbol{c}}^{x,y} + \boldsymbol{g}^{x,y}) + \frac{1}{m(\ddot{c}^z + g^z)}\mathbf{S}\dot{\mathbf{L}}^{x,y} = \frac{\sum_i f_i^z \boldsymbol{p}_i^{x,y}}{\sum_i f_i^z}\,, \tag{48.10}$$

with a simple rotation matrix

$$\mathbf{S} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$

On the right hand side of (48.10) appears the definition of the center of pressure (COP) $\boldsymbol{z}$ of the contact forces, $\boldsymbol{f}_i$. These contact forces are usually unilateral (the robot can push on the ground, not pull)

$$f_i^z \geq 0\,, \tag{48.11}$$

which implies that the CoP is bound to lie in the convex hull of the contact points (Fig. 48.2)

$$\boldsymbol{z}^{x,y} = \frac{\sum_i f_i^z \boldsymbol{p}_i^{x,y}}{\sum_i f_i^z} \in \operatorname{conv}\left\{\boldsymbol{p}_i^{x,y}\right\}\,. \tag{48.12}$$

Combining this inclusion with the dynamic (48.10) reveals an ordinary differential inclusion (ODI)

$$\boldsymbol{c}^{x,y} - \frac{c^z}{\ddot{c}^z + g^z}(\ddot{\boldsymbol{c}}^{x,y} + \boldsymbol{g}^{x,y}) + \frac{1}{m(\ddot{c}^z + g^z)}\mathbf{S}\dot{\mathbf{L}}^{x,y} = \boldsymbol{z}^{x,y} \in \operatorname{conv}\left\{\boldsymbol{p}_i^{x,y}\right\}, \tag{48.13}$$

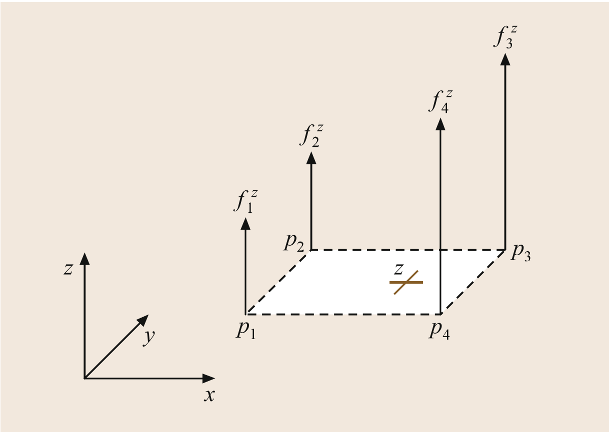


Fig. 48.2 The CoP $\boldsymbol{z}$ is bound to lie in the convex hull of contact points $\boldsymbol{p}_i$

which bounds the motion of the COM $\boldsymbol{c}$, of the robot and the variations of its angular momentum $\mathbf{L}$ with respect to the position $\boldsymbol{p}_i^{x,y}$ of the contact points.

This ODI can be reorganized in the following way

$$\frac{c^z}{\ddot{c}^z + g^z}(\ddot{\boldsymbol{c}}^{x,y} + \boldsymbol{g}^{x,y}) = (\boldsymbol{c}^{x,y} - \boldsymbol{z}^{x,y}) + \frac{1}{m(\ddot{c}^z + g^z)}\mathbf{S}\dot{\mathbf{L}}^{x,y} , \tag{48.14}$$

in order to expose its simple geometric meaning in Fig. 48.3. We can see especially that aside from the effects of gravity $\boldsymbol{g}^{x,y}$ and variations $\dot{\boldsymbol{L}}^{x,y}$ of the angular momentum, the horizontal acceleration $\ddot{\boldsymbol{c}}^{x,y}$ of the COM is the result of a force pushing the COM $\boldsymbol{c}^{x,y}$ away from the CoP $\boldsymbol{z}^{x,y}$, which is bound to lie in the convex hull of the contact points. We have here an intrinsically unstable dynamics.

Note finally that the definition (48.12) of the CoP can be reorganized to show that the horizontal momenta of the contact forces $\boldsymbol{f}_i$ with respect to the CoP $\boldsymbol{z}$ are equal to zero

$$\left[\sum_i (\boldsymbol{p}_i - \boldsymbol{z}) \times \boldsymbol{f}_i\right]^{x,y} = \sum_i (\boldsymbol{p}_i^{x,y} - \boldsymbol{z}^{x,y}) f_i^z = 0 . \tag{48.15}$$

Hence the CoP is also referred to as the zero moment point (ZMP) [48.10, 11].

In Contact with Multiple Surfaces

If the contact points $\boldsymbol{p}_i$ are not all on the same plane, we can introduce a CoP for each contact surface, but we cannot introduce a unique CoP for all contact forces as we did previously. Approximations, and generalizations to limited multiple contact situations have been proposed but haven't been widely adopted [48.11–14]. In the general case, the Newton and Euler equations (48.3) and (48.4) have to be considered explicitly together with the unilaterality condition (48.11) $f_i^z \geq 0$ in order to check which motion is feasible or not [48.15–17]. The problem we have to solve then relates very closely to the *force closure* problem discussed in Chap. 38 on grasping. Different ways to solve it as quickly as possible have been proposed [48.15, 18], and used mostly so far to measure offline the stability robustness of a given motion [48.19, 20], and only recently for motion planning, once again offline [48.21]. Refinements to curved contact surfaces have also been proposed [48.22].

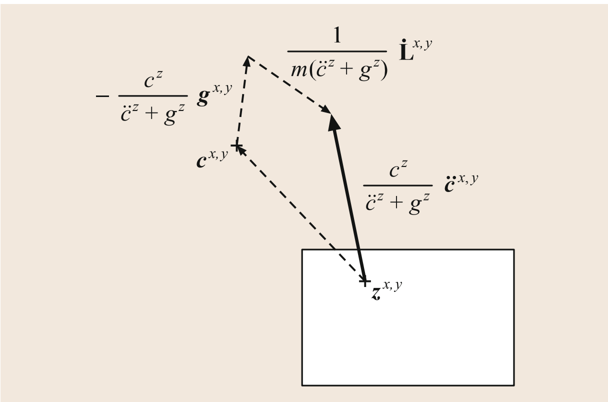


Fig. 48.3 The horizontal acceleration $\ddot{\boldsymbol{c}}^{x,y}$ of the COM of the robot is the sum of a force pushing the COM $\boldsymbol{c}^{x,y}$ away from the CoP $\boldsymbol{z}^{x,y}$ the effect of gravity $-\boldsymbol{g}^{x,y}$ and variations $\dot{\mathbf{L}}^{x,y}$ of the angular momentum

48.2.3 Contact Models

The structure of the Lagrangian dynamics makes it clear that contact forces are central to the modeling and control of legged robots. But note that the only characteristics of these forces that we have introduced so far is their unilaterality (48.11). As a result, the previous analysis applies to various contact situations: walking and running motions, but also sliding situations such as when skiing or skating, or even rolling situations [48.23, 24].

Let us briefly discuss now standard contact models (more details can be found in Chap. 37 on contact modeling and manipulation). Concerning motion and forces tangential to the contact surfaces, a simple Coulomb friction model is usually considered. Concerning motion and forces orthogonal to the contact surfaces, two options are generally considered: a compliant or a rigid model, introducing impacts and other nonsmooth behaviors.

Coulomb Friction

When a contact point $\boldsymbol{p}_i$ is sliding on its contact surface, the corresponding tangential contact force $\boldsymbol{f}_i^{x,y}$ is proportional to the normal force f_i^z in a direction opposite to the sliding motion

sliding

$$\boldsymbol{f}_i^{x,y} = -\mu_0 f_i^z \frac{\dot{\boldsymbol{p}}_i^{x,y}}{\|\dot{\boldsymbol{p}}_i^{x,y}\|} \quad \text{if } \dot{\boldsymbol{p}}_i^{x,y} \neq 0 , \tag{48.16}$$

with $\mu_0 > 0$ the friction coefficient. When the contact point is sticking and not sliding, the norm of the tangential force is simply bounded, with the same friction coefficient

sticking

$$\|\boldsymbol{f}_i^{x,y}\| \leq \mu_0 f_i^z \quad \text{if } \dot{\boldsymbol{p}}_i^{x,y} = 0 . \tag{48.17}$$

This is typically referred to as the *friction cone*. Note that this friction model directly implies the unilaterality condition (48.11).