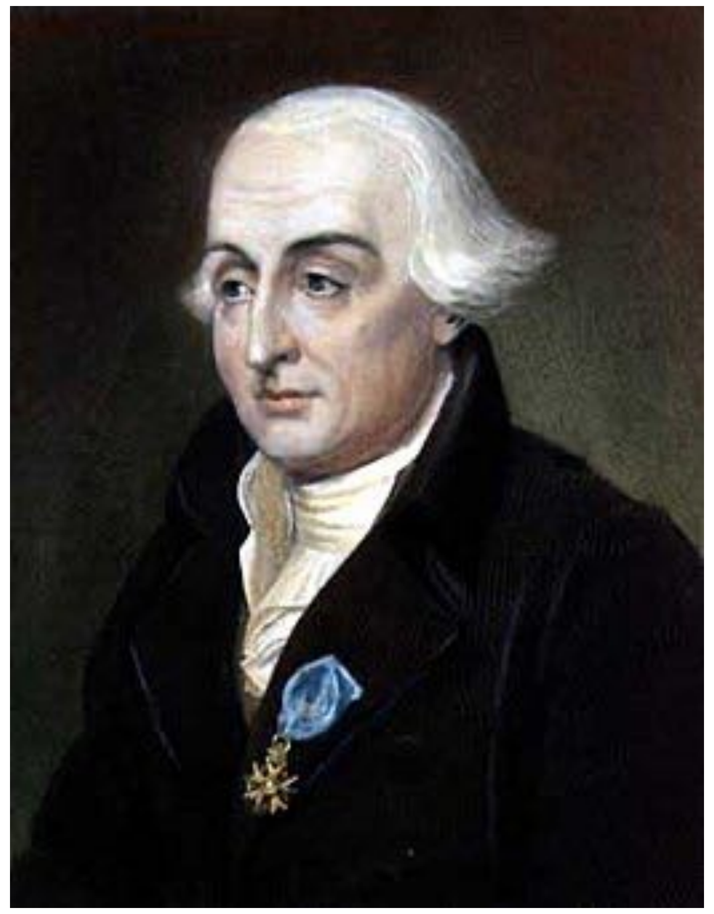
Constrained Dynamics

A proximal and sparse approach

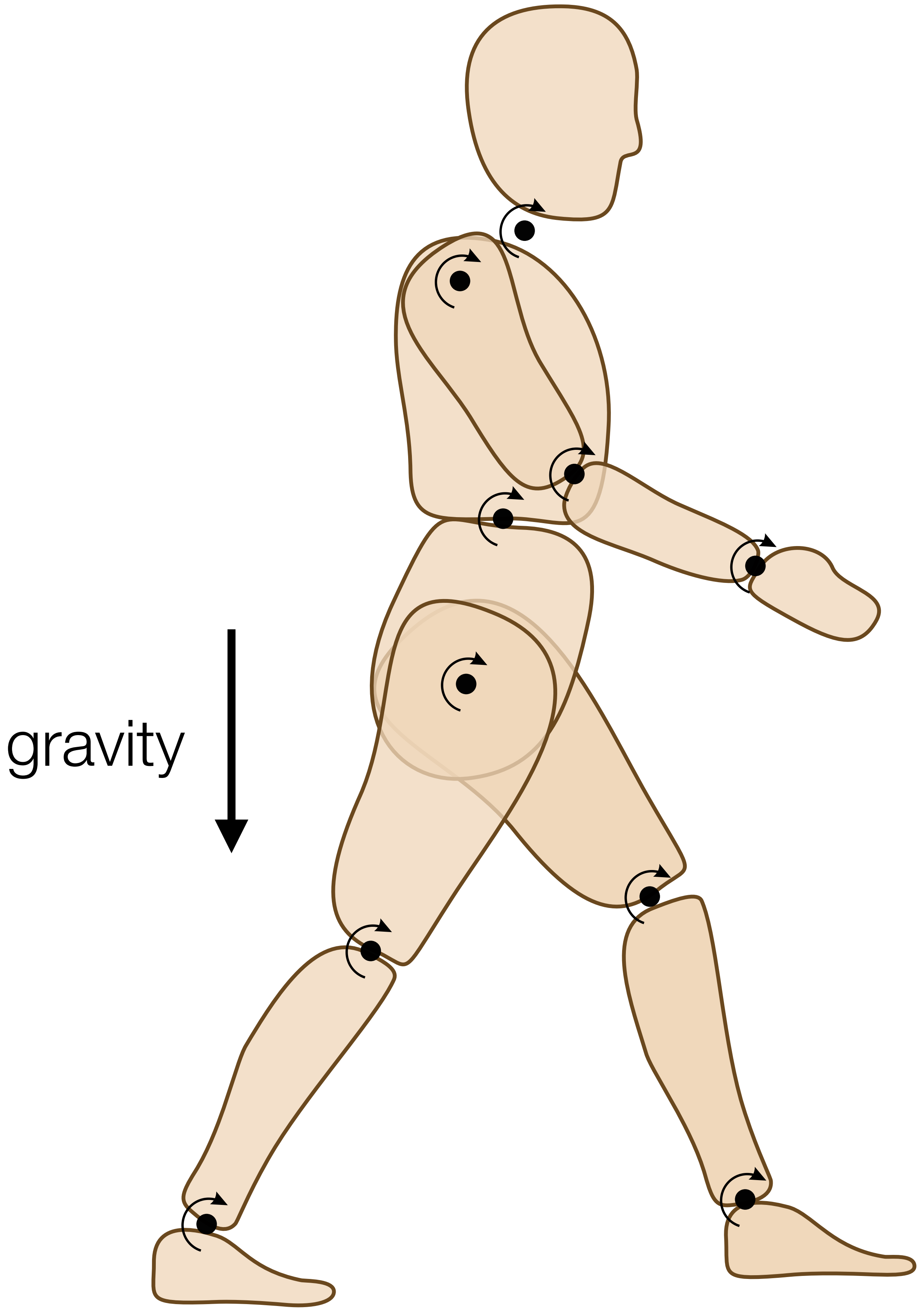
Justin Carpentier

Researcher, INRIA and ENS, Paris

Contact: the Physical Problem



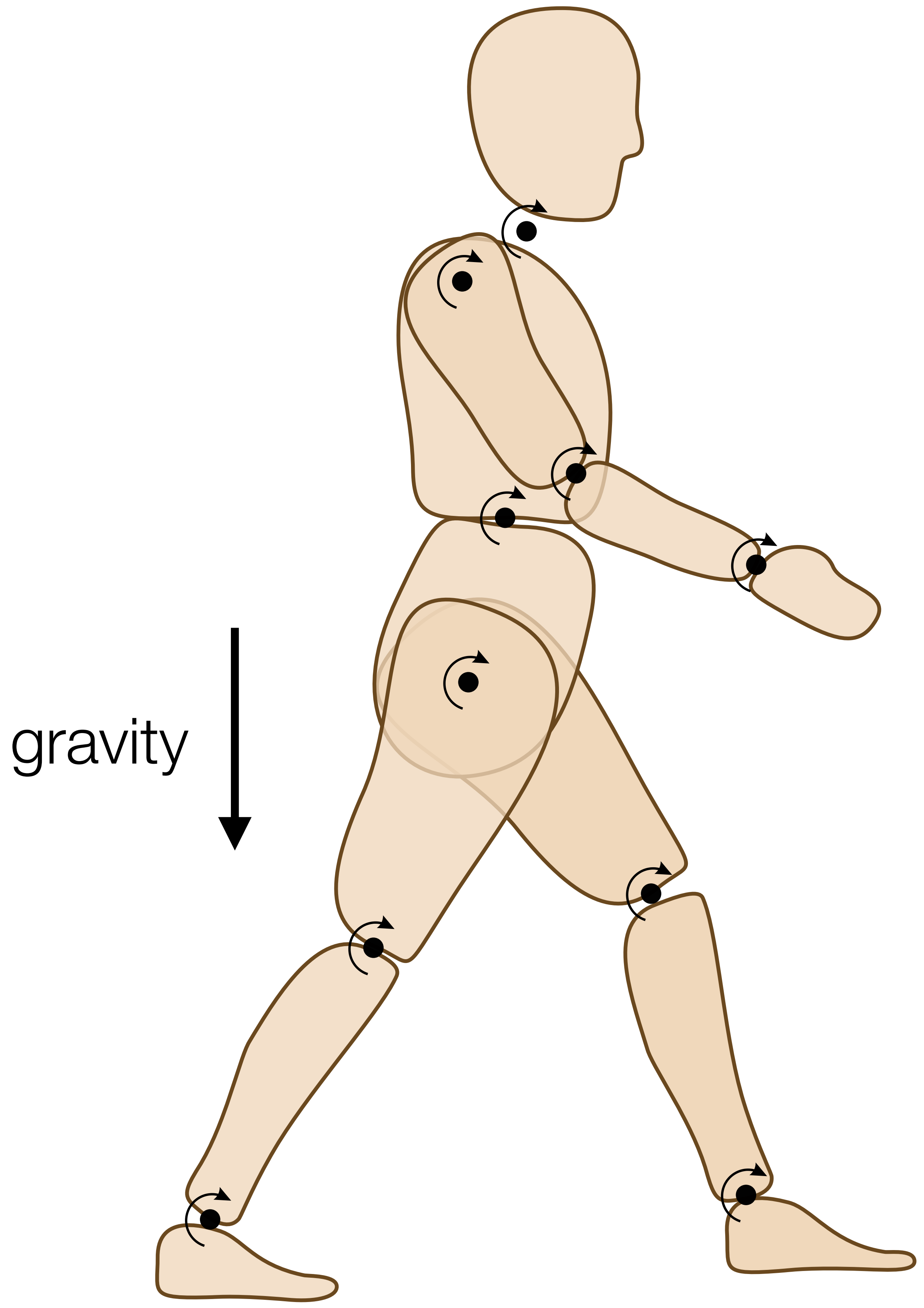
Joseph-Louis Lagrange

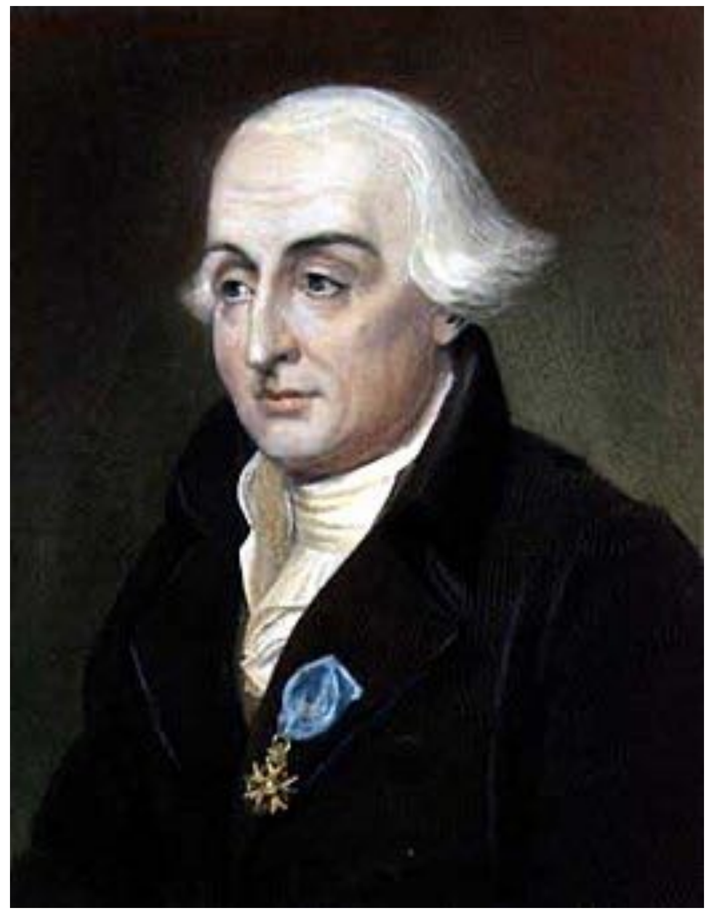


The poly-articulated system dynamics
is driven by the so-called **Lagrangian** dynamics:

$$\underbrace{M(q)\ddot{q}}_{\text{Mass Matrix}} + \underbrace{C(q,\dot{q})}_{\text{Coriolis centrifugal}} + \underbrace{G(q)}_{\text{Gravity}} = \underbrace{\tau}_{\text{Motor torque}}$$

Contact: the Physical Problem



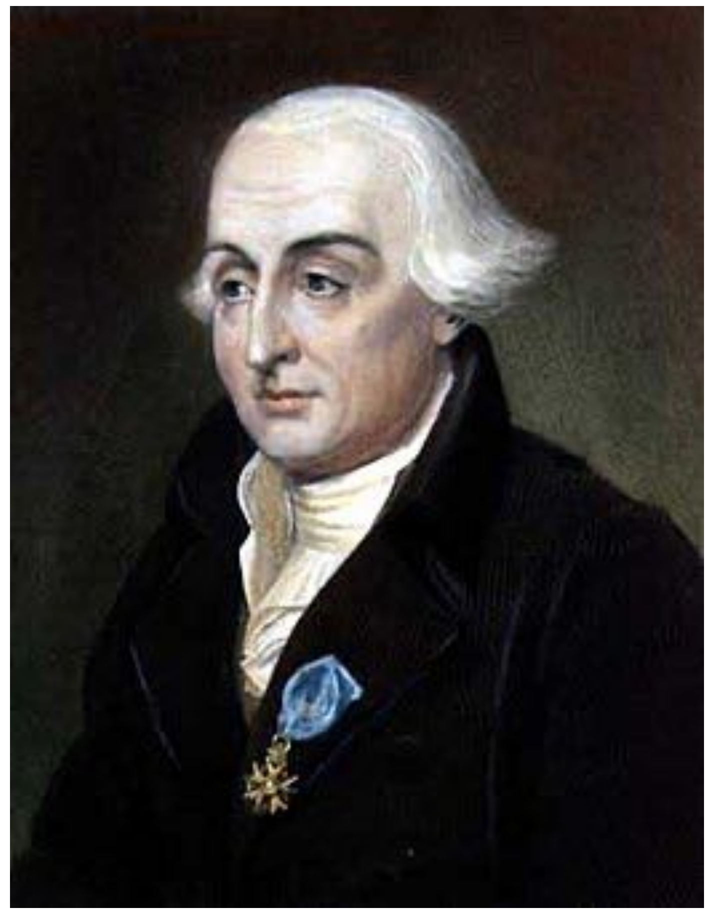


Joseph-Louis Lagrange

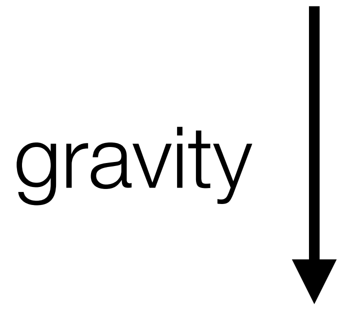
The poly-articulated system dynamics
is driven by the so-called **Lagrangian** dynamics:

$$\underbrace{M(q)\ddot{q}}_{\text{Mass Matrix}} + \underbrace{C(q,\dot{q})}_{\text{Coriolis centrifugal}} + \underbrace{G(q)}_{\text{Gravity}} = \underbrace{\tau}_{\text{Motor torque}}$$

Contact: the Physical Problem


Joseph-Louis Lagrange

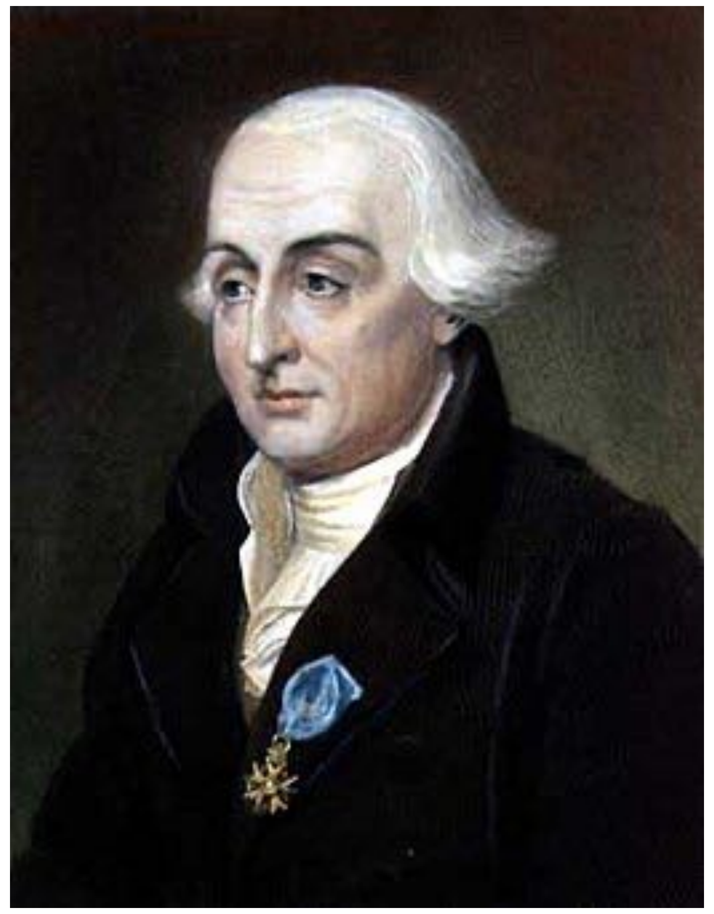
The poly-articulated system dynamics
is driven by the so-called **Lagrangian** dynamics:



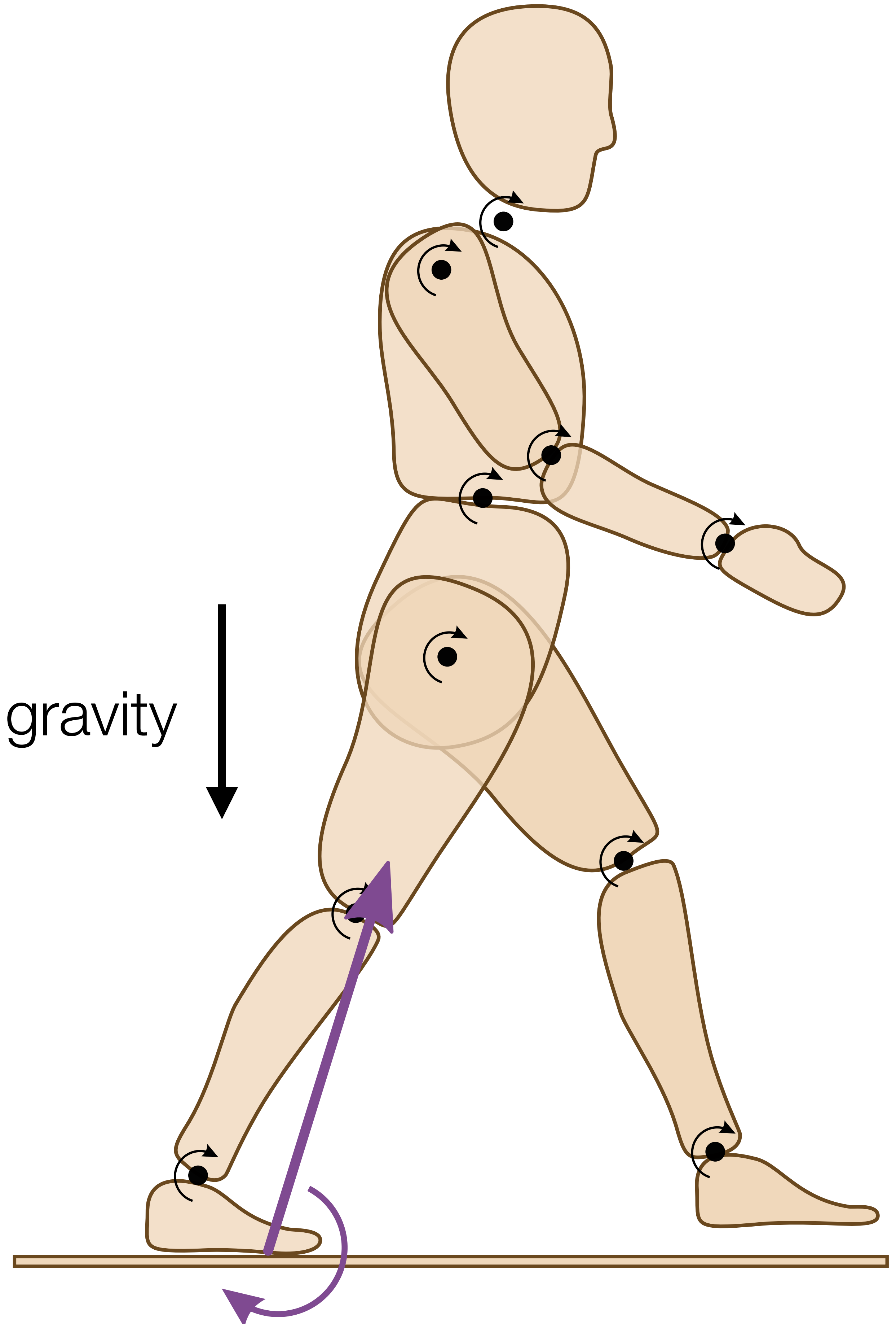
$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = \tau$$

Mass Matrix — Coriolis centrifugal — Gravity — Motor torque

Contact: the Physical Problem



Joseph-Louis Lagrange

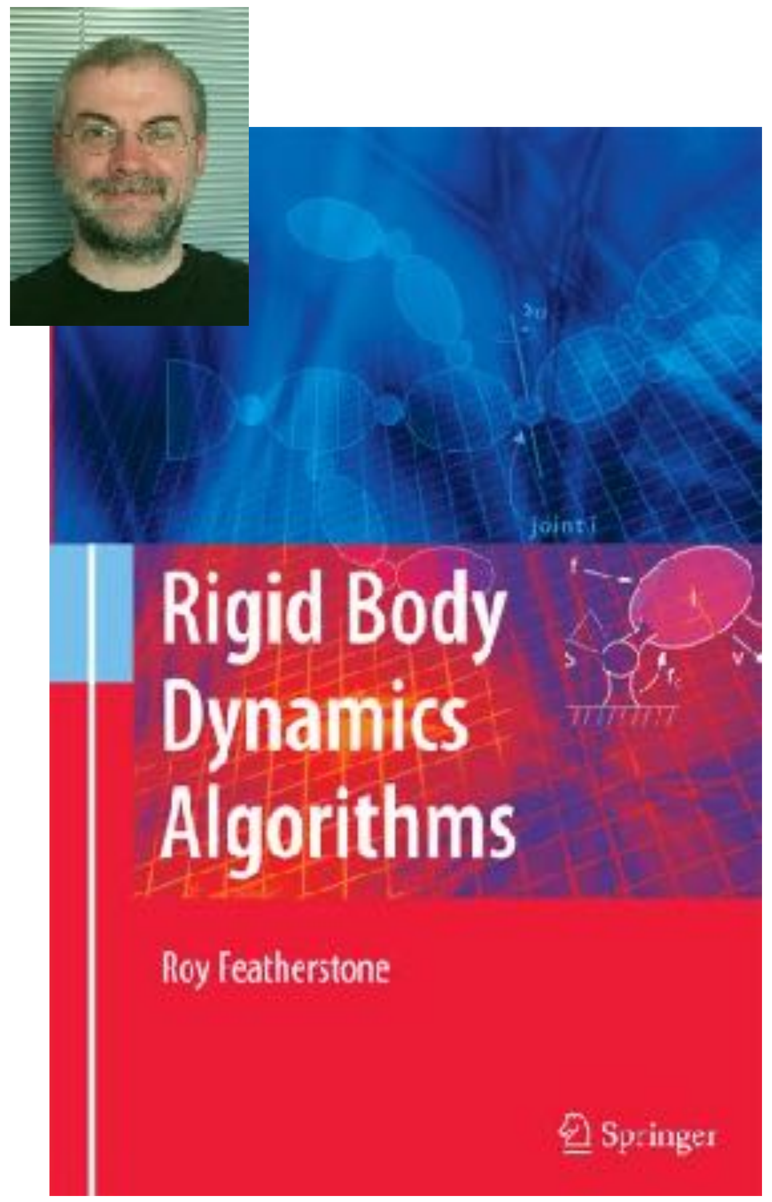


The poly-articulated system dynamics
is driven by the so-called **Lagrangian** dynamics:

$$\underbrace{M(q)\ddot{q}}_{\text{Mass Matrix}} + \underbrace{C(q,\dot{q})}_{\text{Coriolis centrifugal}} + \underbrace{G(q)}_{\text{Gravity}} = \underbrace{\tau}_{\text{Motor torque}} + \underbrace{J_c^\top(q)\lambda_c}_{\text{External forces}}$$

The Rigid Body Dynamics Algorithms

Goal: exploit at best the **sparsity** induced by the kinematic tree



Roy Featherstone

The Articulated Body Algorithm

$$\boxed{\ddot{q} = \textbf{ForwardDynamics}\left(q, \dot{q}, \tau, \lambda_c\right)}$$

Simulation

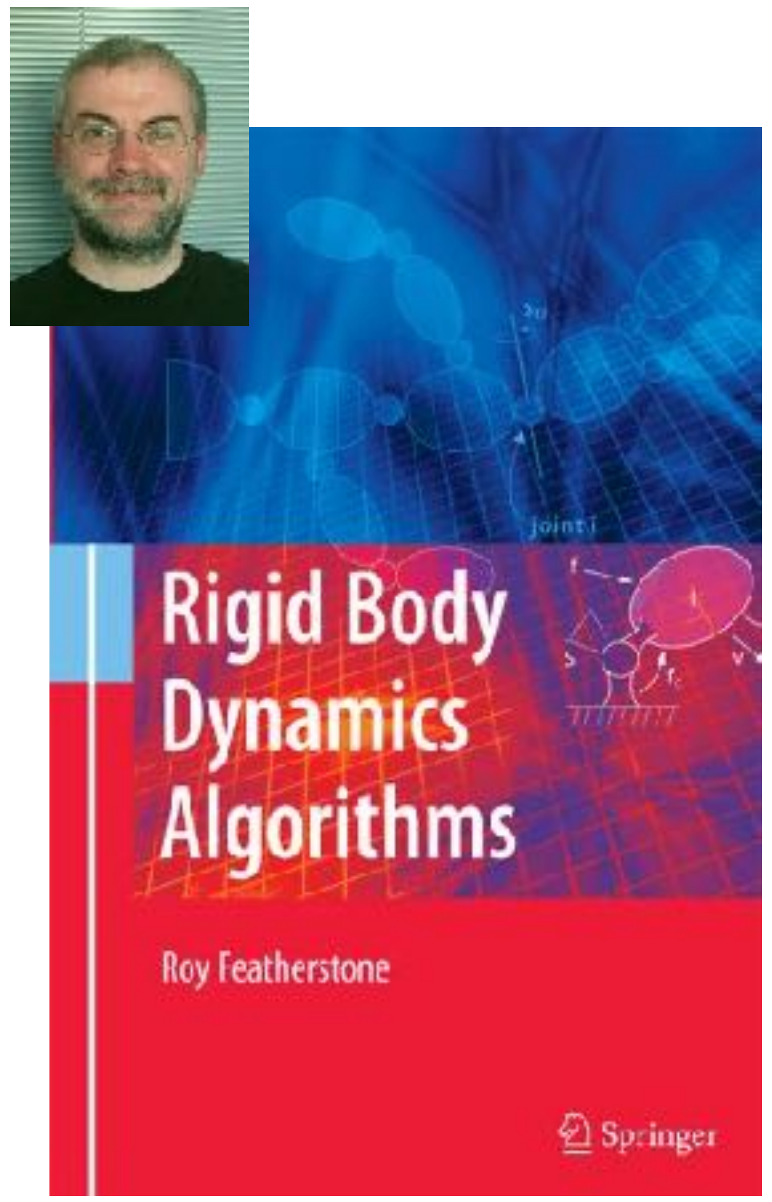
Control

$$\boxed{\tau = \textbf{InverseDynamics}\left(q, \dot{q}, \ddot{q}, \lambda_c\right)}$$

The Recursive Newton-Euler Algorithm

$$\underbrace{M(q)\ddot{q}}_{\text{Mass Matrix}} + \underbrace{C(q, \dot{q})}_{\text{Coriolis centrifugal}} + \underbrace{G(q)}_{\text{Gravity}} = \underbrace{\tau}_{\text{Motor torque}} + \underbrace{J_c^\top(q)\lambda_c}_{\text{External forces}}$$

The Rigid Body Dynamics Algorithms



Roy Featherstone

<u>Goal:</u> exploit at best the **sparsity** induced by the kinematic tree

The Articulated Body Algorithm

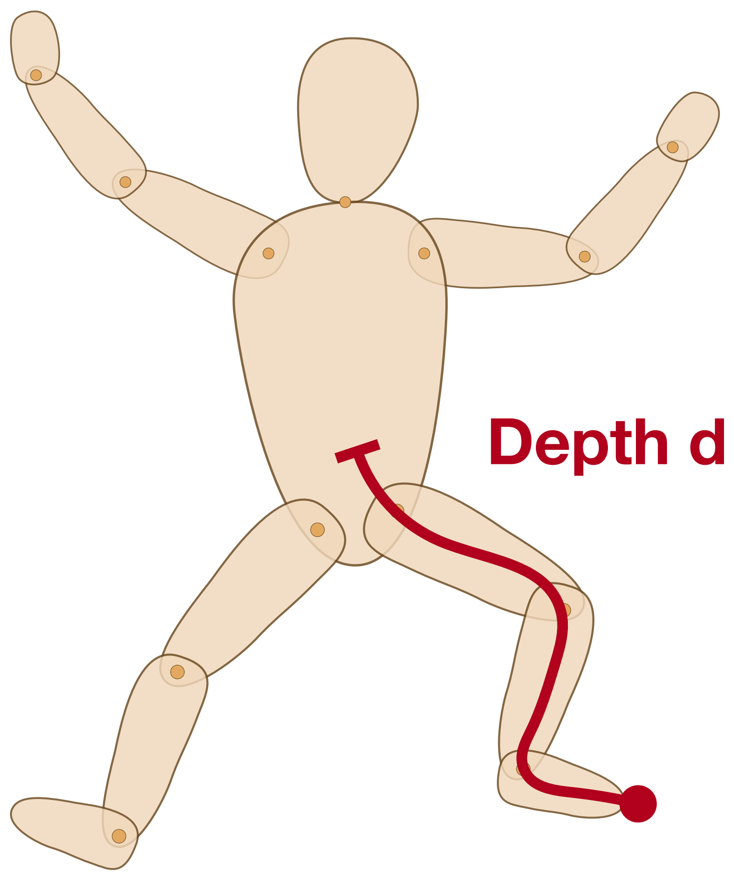
$$\boxed{\ddot{q} = \textbf{ForwardDynamics}\left(q, \dot{q}, \tau, \lambda_c\right)}$$

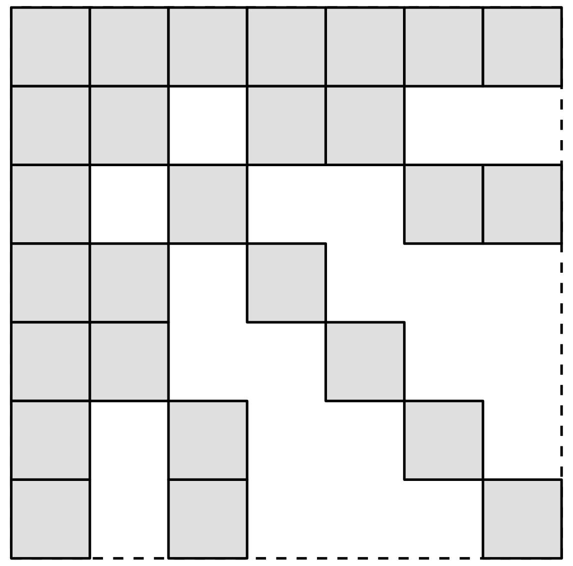
Simulation

Control

$$\boxed{\tau = \textbf{InverseDynamics}\left(q, \dot{q}, \ddot{q}, \lambda_c\right)}$$

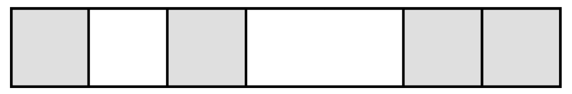
The Recursive Newton-Euler Algorithm





$$M(q)\ddot{q} \; + \; C(q, \dot{q}) \; + \; G(q) \; = \; \tau \; + \; J_c^{\top}(q)\lambda_c$$

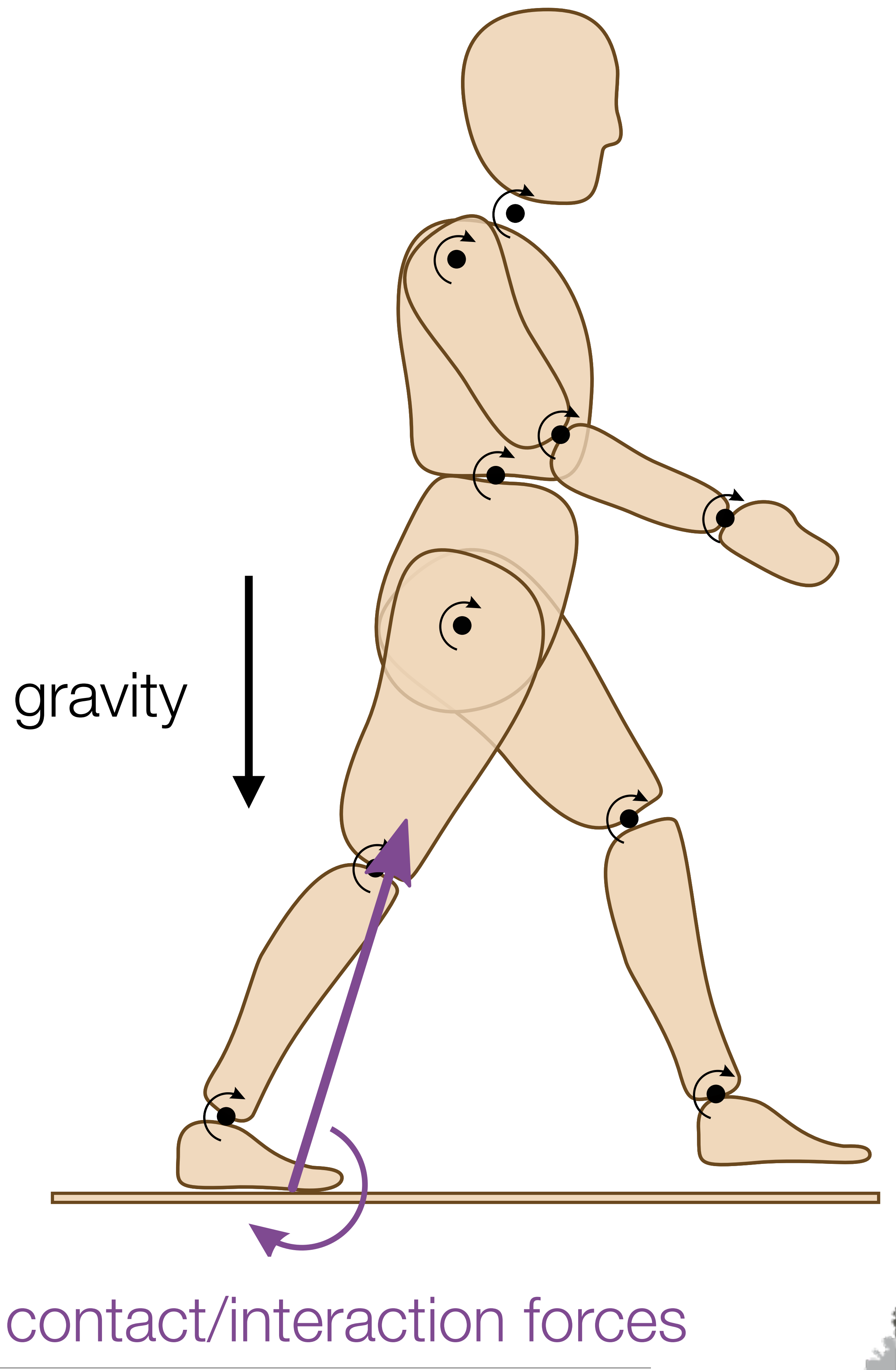
Mass Matrix | Coriolis centrifugal | Gravity | Motor torque | External forces



Gaol of this class

Understand the **various approaches** of the state of the art to compute λ_c in:

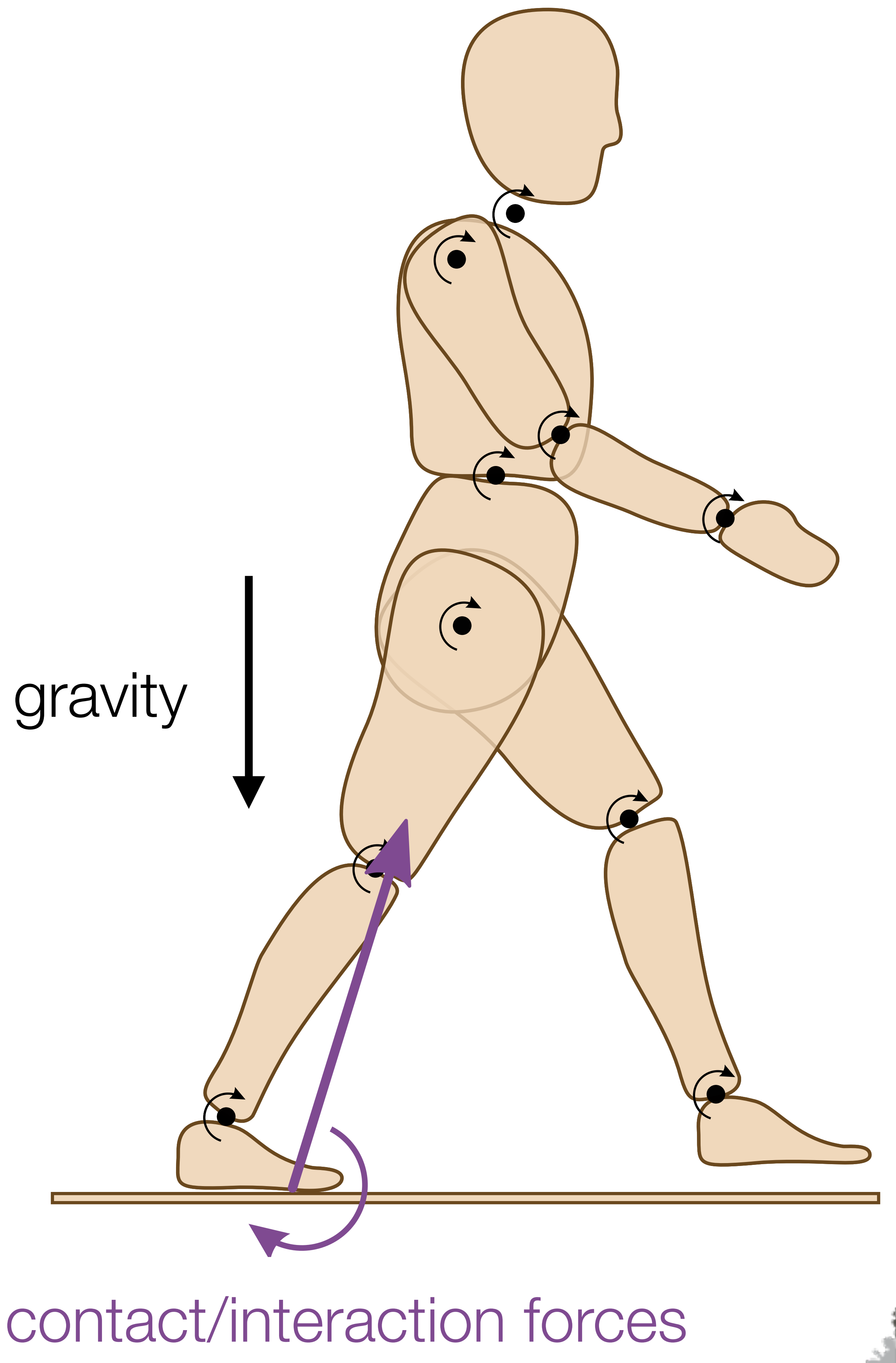
$$M(q)\ddot{q} \;+\; C(q,\dot{q}) \;+\; G(q) \;=\; \tau \;+\; J_c^{\top}(q)\,\lambda_c$$



Gaol of this class

Understand the **various approaches** of the state of the art to compute λ_c in:

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = \tau + J_c^\top(q)\lambda_c$$

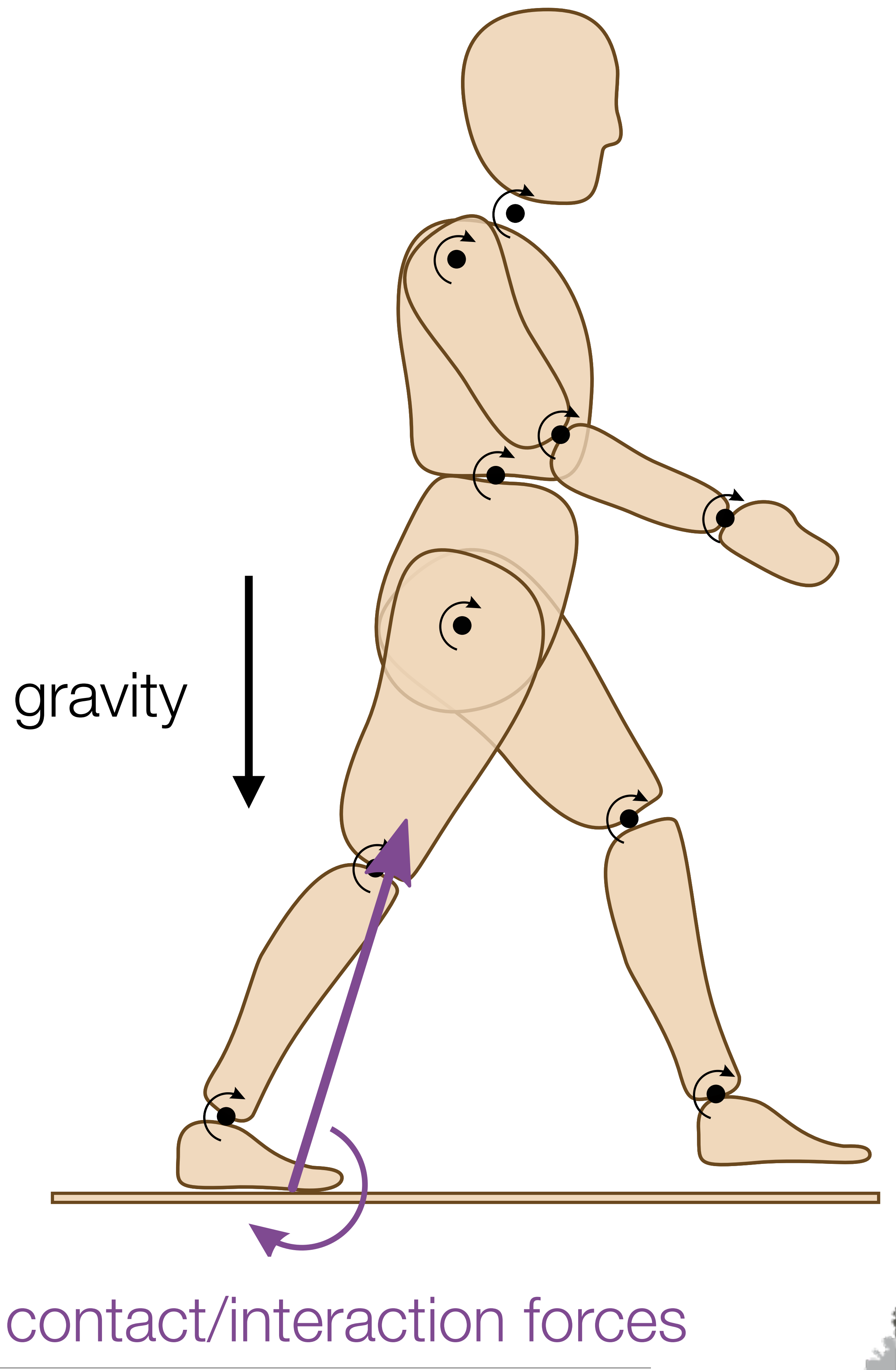


Gaol of this class

Understand the **various approaches** of the state of the art to compute λ_c in:

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = \tau + J_c^\top(q)\,\lambda_c$$

Soft contact | spring-damper model



Gaol of this class

Understand the **various approaches** of the state of the art to compute λ_c in:

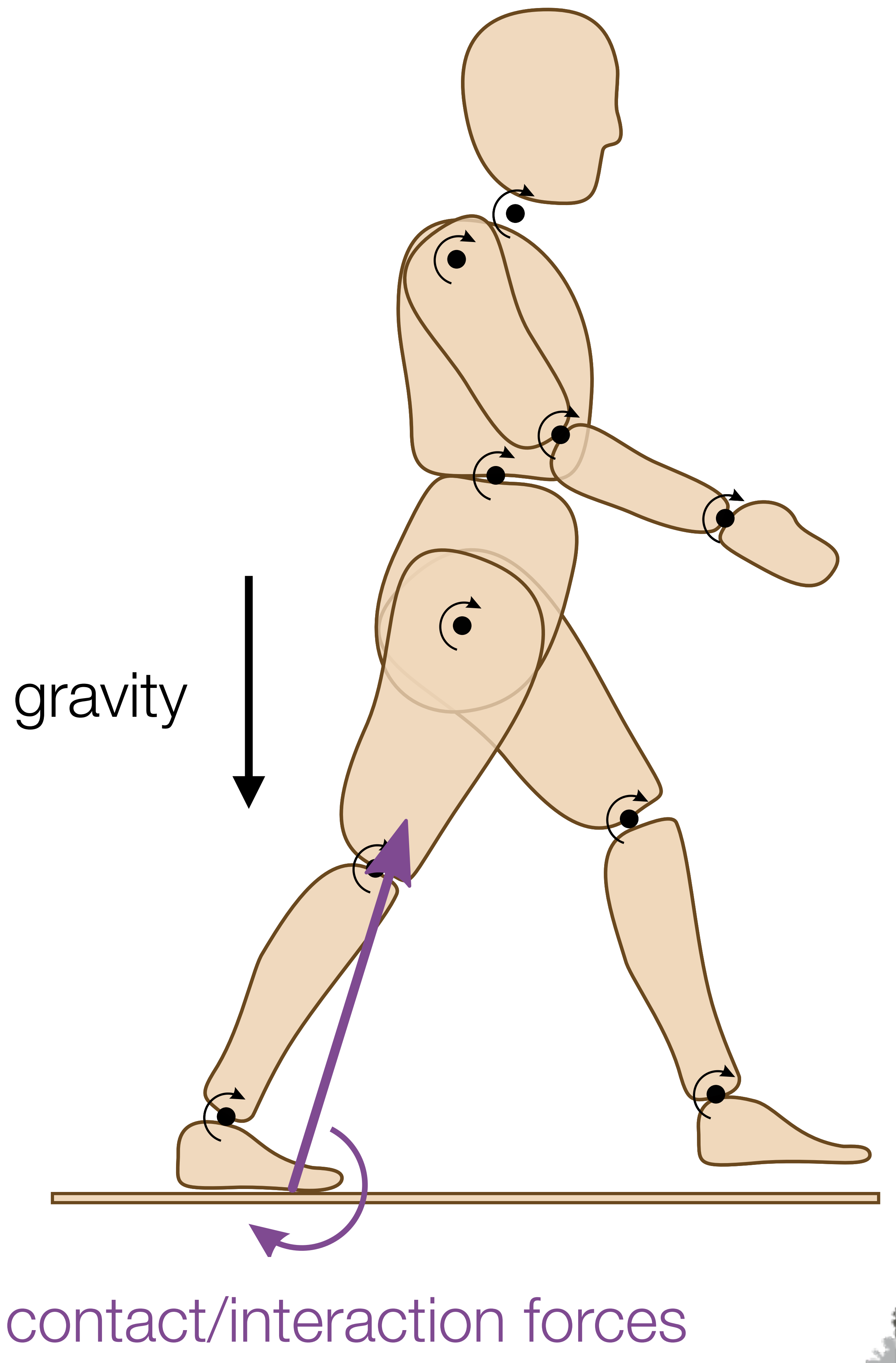
$$M(q)\ddot{q} \; + \; C(q,\dot{q}) \; + \; G(q) \; = \; \tau \; + \; J_c^\top(q)\,\lambda_c$$

Soft contact

- spring-damper model

Rigid contact

- bilateral contact model
- unilateral contact model



Gaol of this class

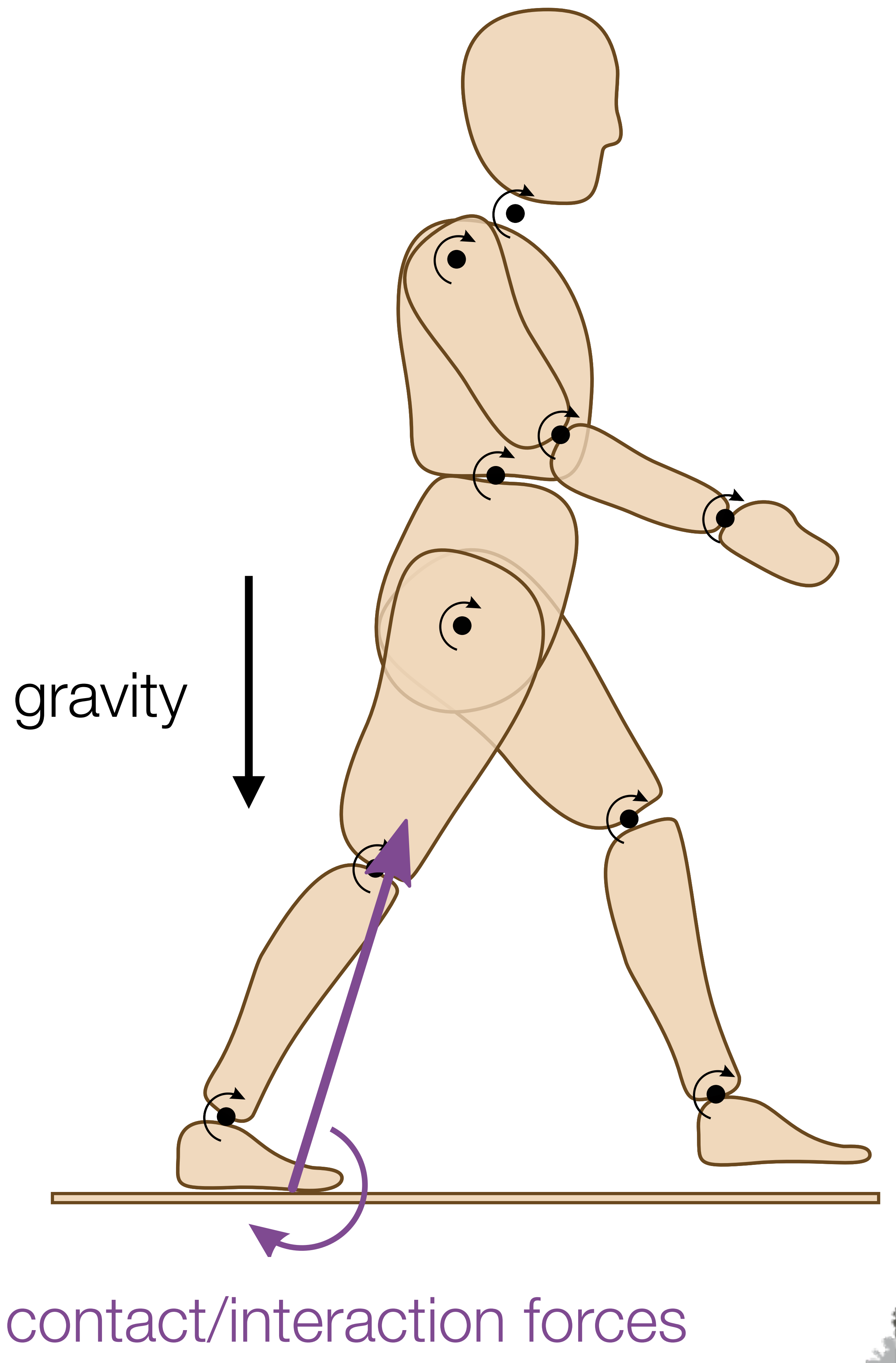
Understand the **various approaches** of the state of the art to compute λ_c in:

$$M(q)\ddot{q} + C(q,\dot{q}) + G(q) = \tau + J_c^\top(q)\,\lambda_c$$

Soft contact
- spring-damper model

Rigid contact
- bilateral contact model
- unilateral contact model

Mixed contact
- the relaxed contact model



The Soft Contact Problem

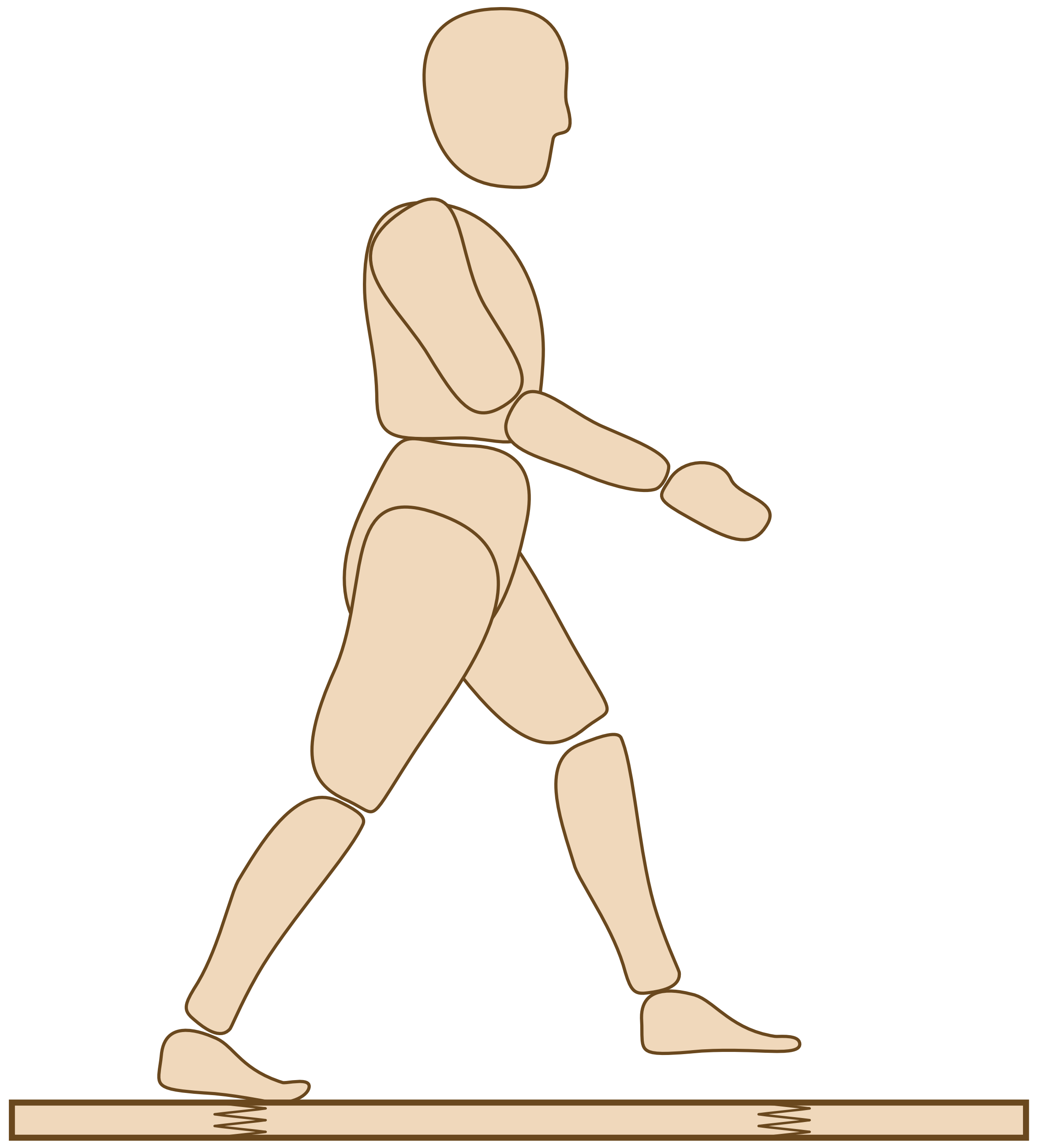

ARENA
SALTO


ARENA
SALTO

Soft contact: the spring-damper model

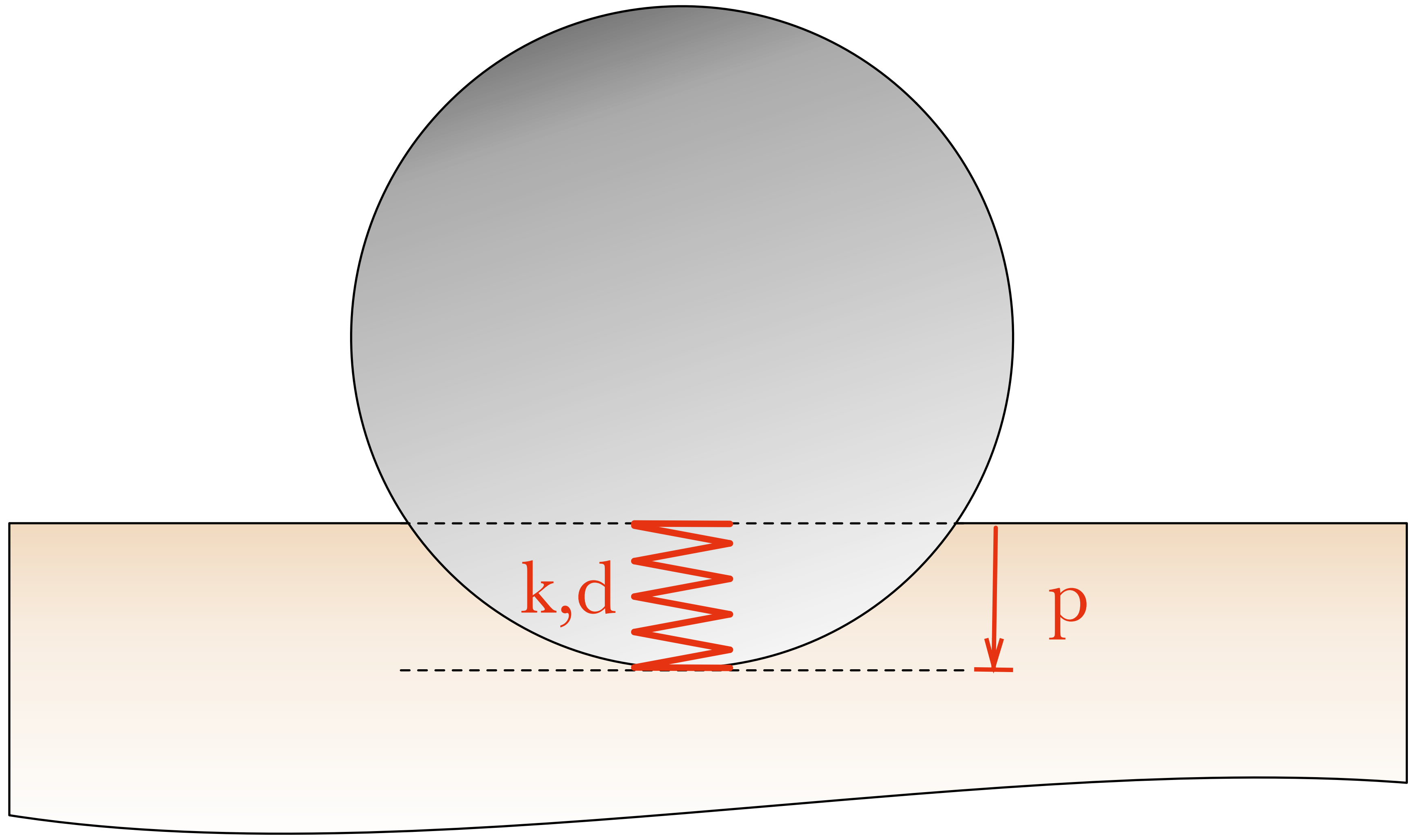
This is the **simplest** contact model, very **intuitive** and **straightforward** to implement

This contact model is defined by the spring k and the damper d quantities, reading:



$$\lambda_c^n = \max(-k\,.\,p - d\,.\,\dot{p}, 0)$$

the max function means:
the ground can ONLY push




Z
X
Y
Right
Left
General
1
40
80
120
160
200
240
280
320
360
400
440
480
520
565

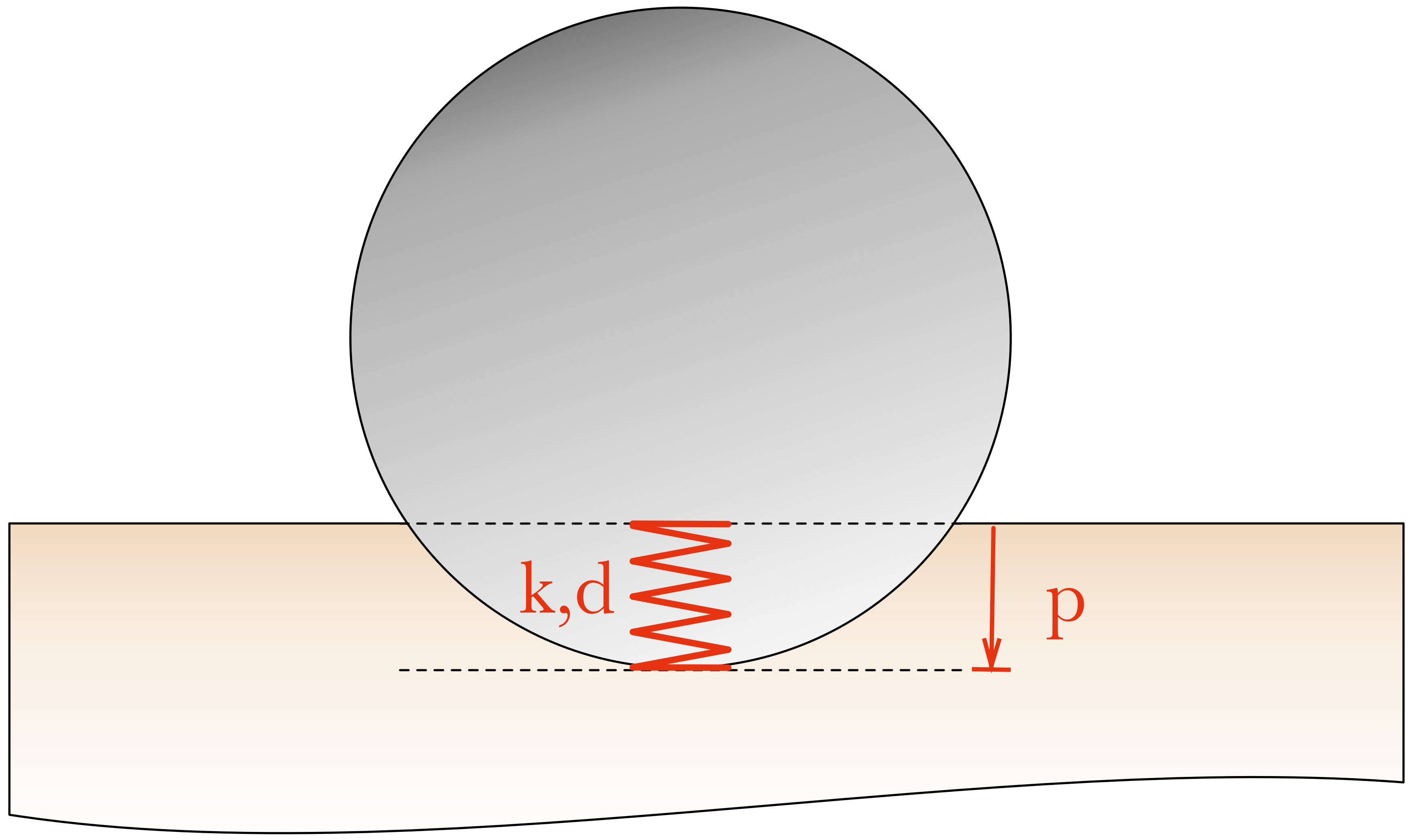
Z
X
Y
Right
Left
General
1
40
80
120
160
200
240
280
320
360
400
440
480
520
565

Soft contact: the spring-damper model

This is the **simplest** contact model, very **intuitive** and **straightforward** to implement

BUT

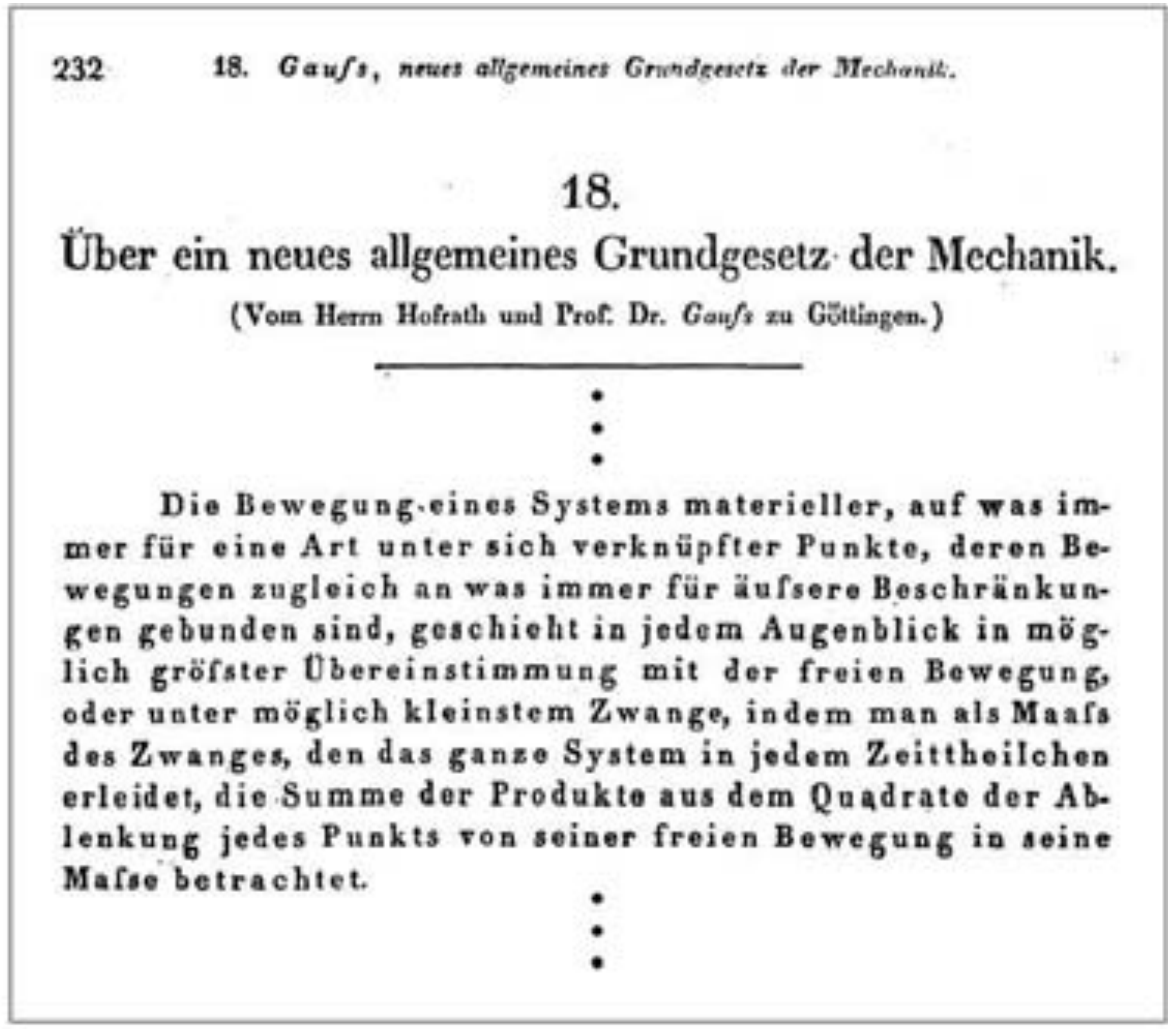
not **relevant** to model rigid interface ($k \to \infty$), requires **stable integrator** (stiff equation)



The Rigid Contact Problem

bilateral contacts

The Least-Constraint Principle



232 18. *Gauſs, neues allgemeines Grundgesetz der Mechanik.*

18.

Über ein neues allgemeines Grundgesetz der Mechanik.

(Vom Herrn Hofrath und Prof. Dr. *Gauſs* zu Göttingen.)

⋮

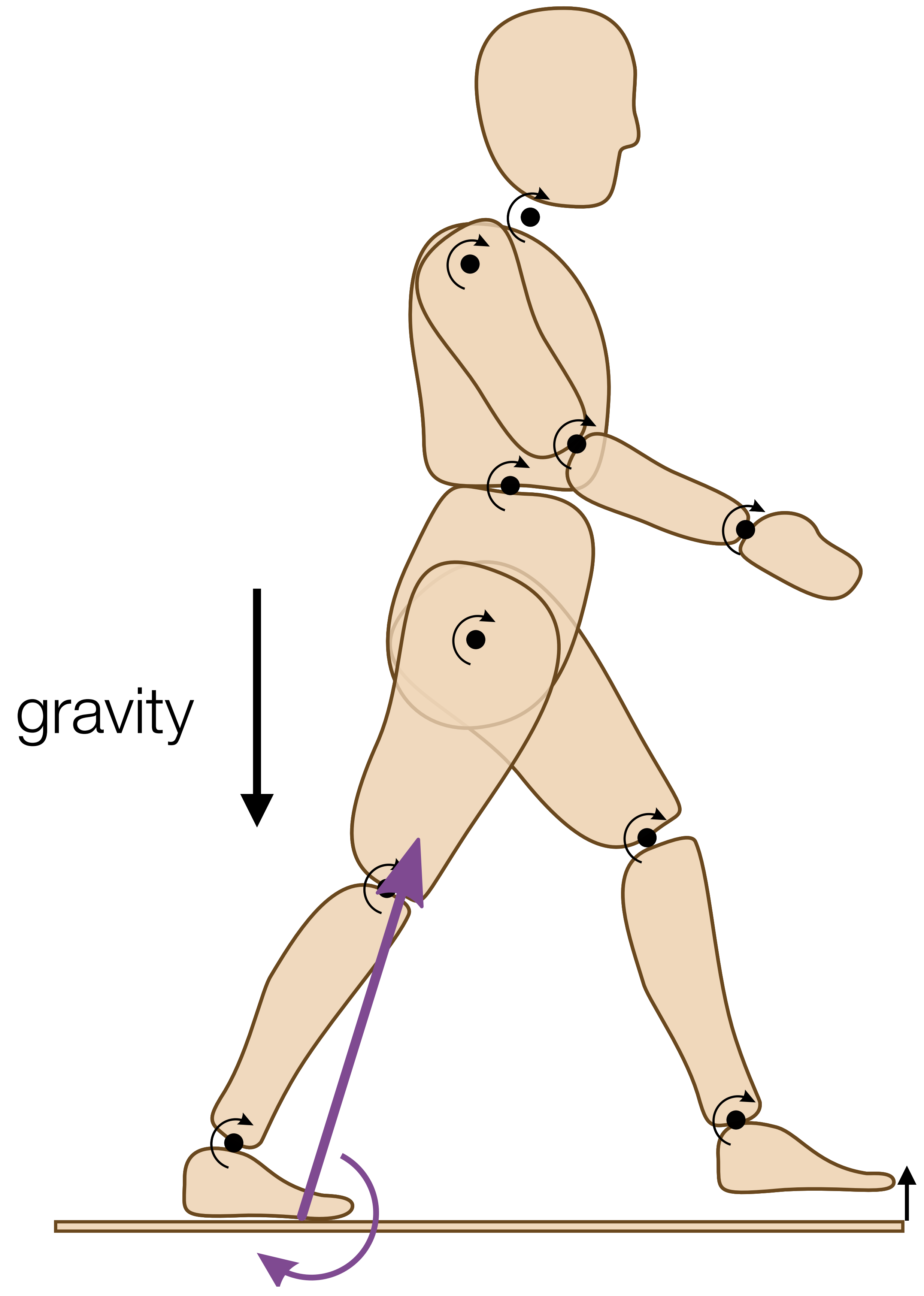
Die Bewegung eines Systems materieller, auf was immer für eine Art unter sich verknüpfter Punkte, deren Bewegungen zugleich an was immer für äuſsere Beschränkungen gebunden sind, geschieht in jedem Augenblick in möglich gröſster Übereinstimmung mit der freien Bewegung, oder unter möglich kleinstem Zwange, indem man als Maaſs des Zwanges, den das ganze System in jedem Zeittheilchen erleidet, die Summe der Produkte aus dem Quadrate der Ablenkung jedes Punkts von seiner freien Bewegung in seine Maſse betrachtet.

⋮

Carl Friedrich Gauss

"The motion of a system of material points. . . takes place in every moment ***in maximum accordance with the free movement or under least constraint****; [. . .] the measure of constraint, [. . .], is considered as the sum of products of mass and the square of the deviation to the free motion"*

The Least Constraint Principle as a classic QP



Problem: knowing q and $\dot{q}$, we aim at retrieving $\ddot{q}$ and λ_c

least distance w.r.t to the unconstrained acceleration

a metric induced by the kinetic energy

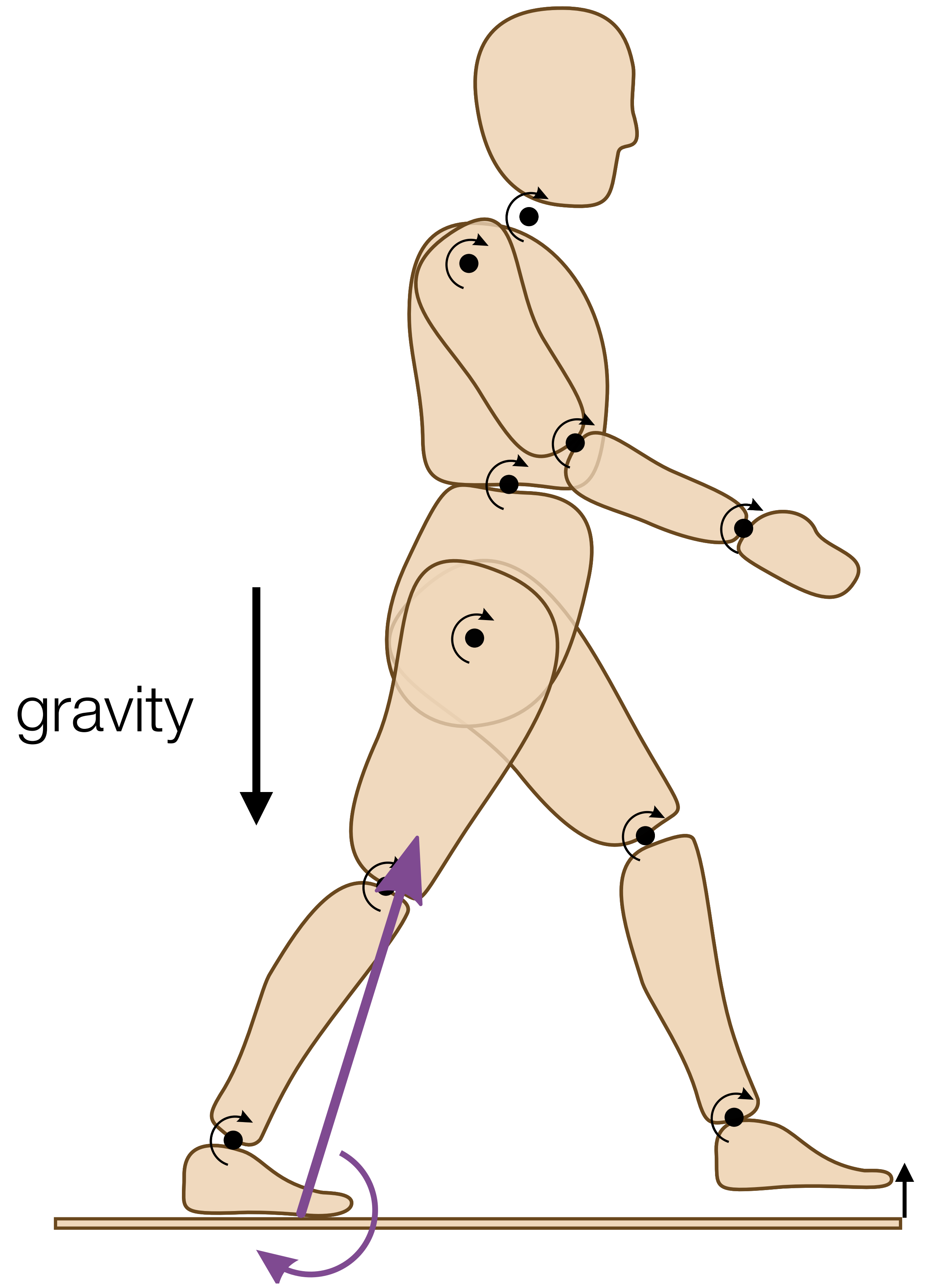
$$\min_{\ddot{q}} \quad \frac{1}{2} \| \ddot{q} - \ddot{q}_f \|^2_{M(q)}$$

$$c(q) = 0$$

gap between floor and foot

where $\ddot{q}_f \overset{\text{def}}{=} M^{-1}(q)\big(\tau - C(q,\dot{q}) - G(q)\big)$ is the so-called **free acceleration** (without constraint)

The Least Constraint Principle as a classic QP



Problem: knowing q and $\dot{q}$, we aim at retrieving $\ddot{q}$ and λ_c

$$\min_{\ddot{q}} \quad \frac{1}{2}\overbrace{\|\ddot{q} - \ddot{q}_f\|}^{\text{least distance w.r.t to the unconstrained acceleration}}{}^2_{M(q)}$$

a metric induced by the kinetic energy

$$c(q) = 0$$

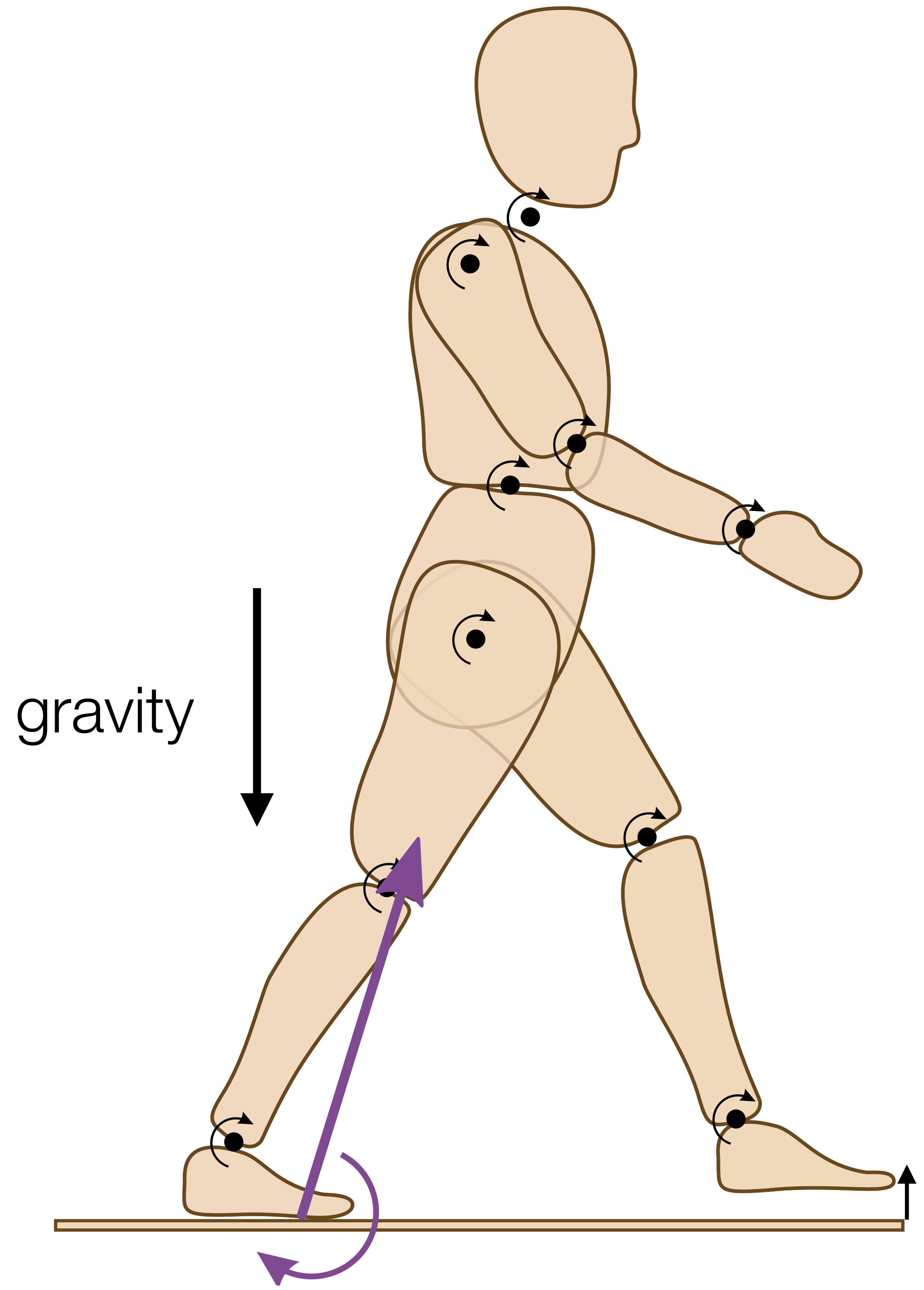
gap between floor and foot

$$\min_{\ddot{q}} \quad \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}$$

$$c(q) = 0$$

where $\ddot{q}_f \stackrel{\text{def}}{=} M^{-1}(q)\big(\tau - C(q,\dot{q}) - G(q)\big)$ is the so-called **free acceleration** (without constraint)

The Least Constraint Principle as a classic QP



<u>Problem:</u> knowing q and $\dot{q}$, we aim at retrieving $\ddot{q}$ and λ_c

least distance w.r.t to the unconstrained acceleration

a metric induced by the kinetic energy

$$\min_{\ddot{q}} \quad \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}$$

$$c(q) = 0$$

gap between floor and foot

$$\min_{\ddot{q}} \quad \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}$$

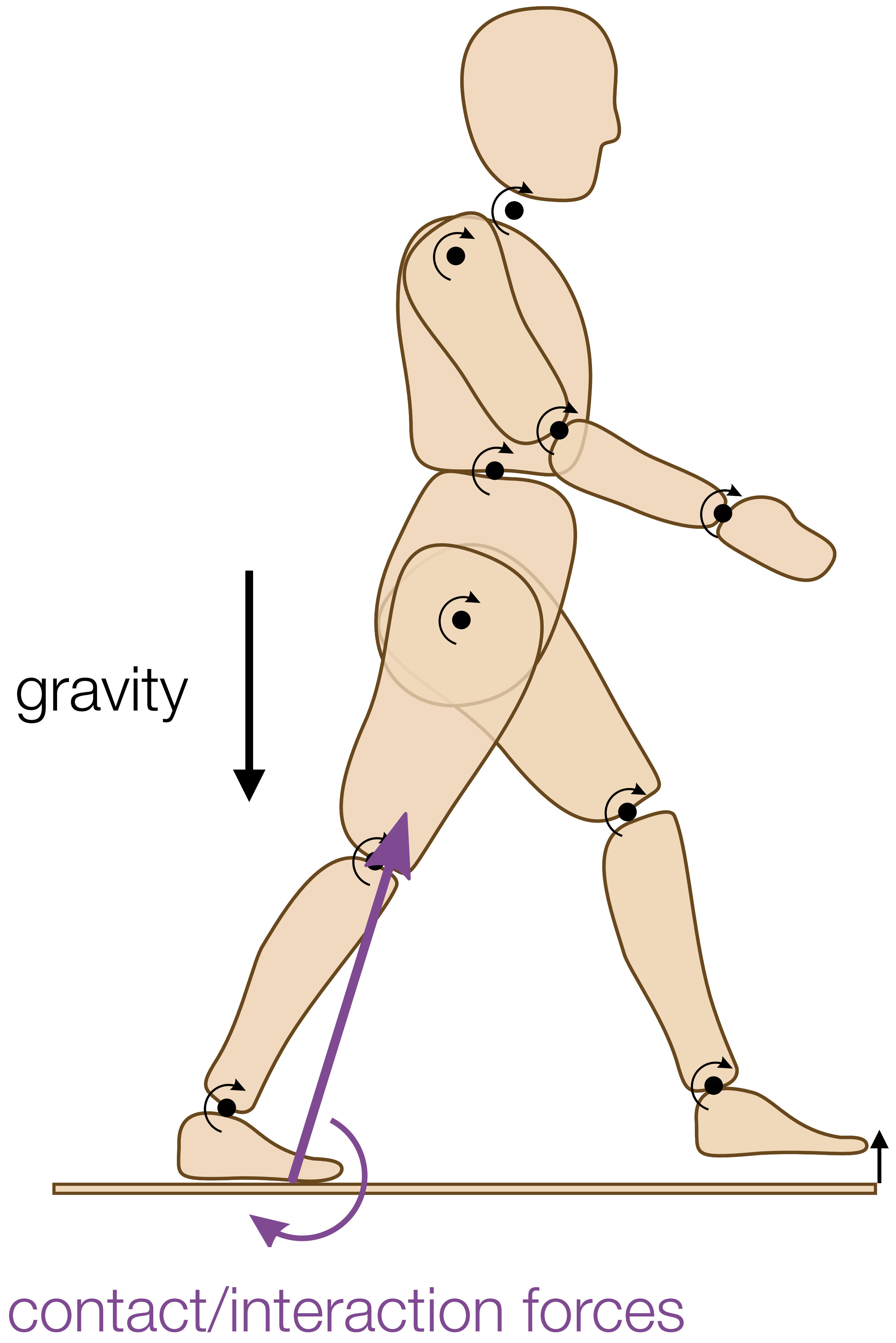
$$c(q) = 0$$

index reduction = time derivation

$$J_c(q)\,\dot{q} = 0$$

where $\ddot{q}_f \overset{\text{def}}{=} M^{-1}(q)\big(\tau - C(q,\dot{q}) - G(q)\big)$ is the so-called **free acceleration** (without constraint)

The Least Constraint Principle as a classic QP



<u>Problem:</u> knowing q and $\dot{q}$, we aim at retrieving $\ddot{q}$ and λ_c

least distance w.r.t to the unconstrained acceleration

a metric induced by the kinetic energy

$$\min_{\ddot{q}} \quad \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}$$

$$c(q) = 0$$

gap between floor and foot

$$\min_{\ddot{q}} \quad \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}$$

$$c(q) = 0$$

index reduction = time derivation

$$J_c(q)\,\dot{q} = 0$$

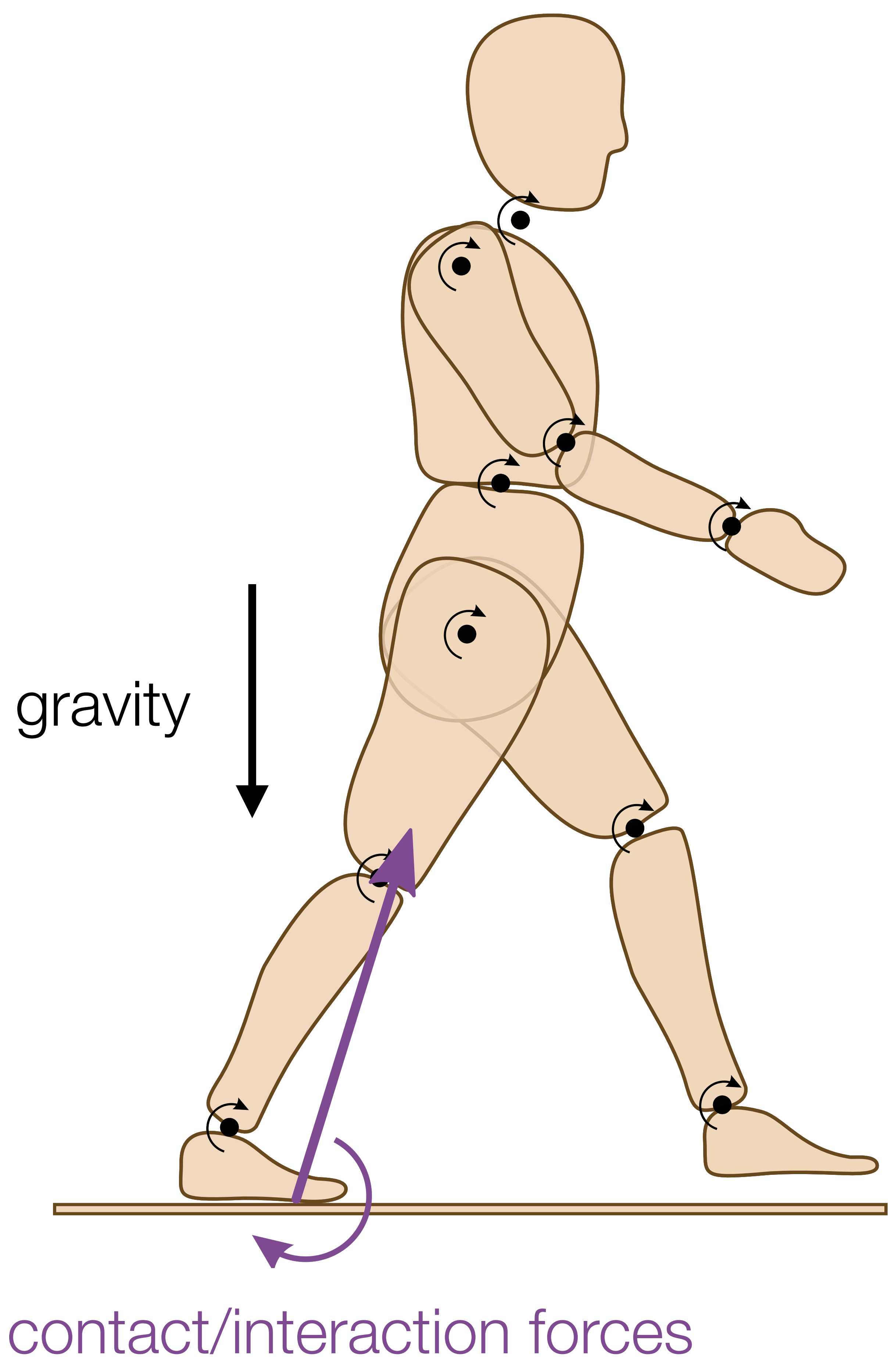
index reduction

$$\boxed{J_c(q)\,\ddot{q} + \underbrace{\dot{J}_c(q,\dot{q})\dot{q}}_{\gamma_c(q,\dot{q})} = 0}$$

the constraint differentiated twice w.r.t. time

where $\ddot{q}_f \stackrel{\text{def}}{=} M^{-1}(q)\big(\tau - C(q,\dot{q}) - G(q)\big)$ is the so-called **free acceleration** (without constraint)

The Least Action Principle as a classic QP

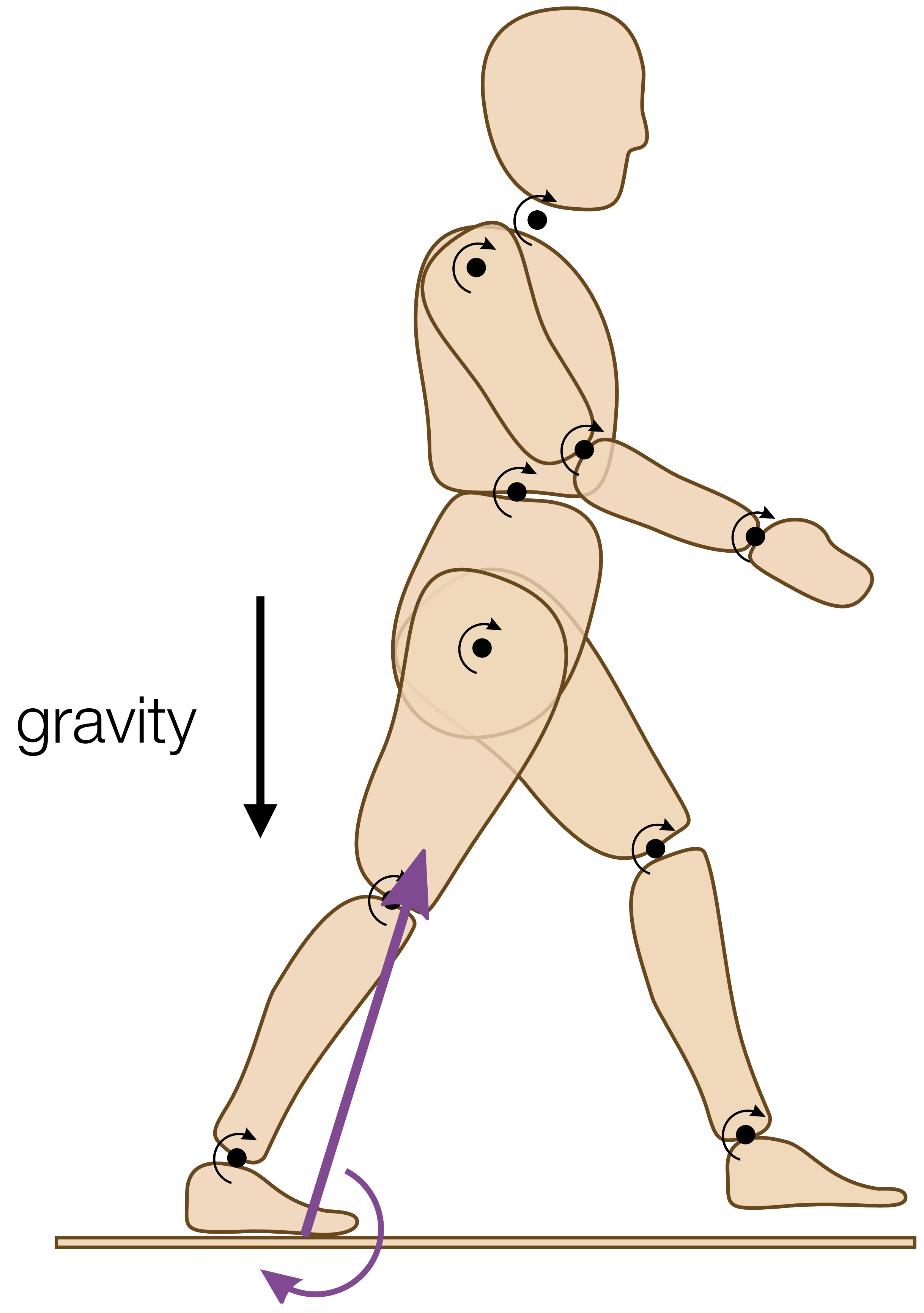


<u>Problem:</u> we have now formed an equality-constrained QP.

$$\min_{\ddot{q}} \quad \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}$$
$$J_c(q)\,\ddot{q} + \gamma_c(q,\dot{q}) = 0$$

How to solve it? Where do the contact forces lie?

The Least Action Principle as a classic QP



Problem: we have now formed an equality-constrained QP.

$$\min_{\ddot{q}} \quad \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}$$
$$J_c(q)\,\ddot{q} + \gamma_c(q,\dot{q}) = 0$$

How to solve it? Where do the contact forces lie?

The solution can be retrieved by **deriving the KKT conditions** of the QP problem via the so-called **Lagrangian:**

dual variable = contact forces

$$L(\ddot{q},\lambda_c) = \underbrace{\frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}}_{\text{cost function}} - \lambda_c^\top \underbrace{\left(J_c(q)\ddot{q} + \gamma_c(q,\dot{q})\right)}_{\text{equality constraint}}$$

Solving the Lagrangian contact problem

$$\min_{\ddot{q}} \quad \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}$$
$$J_c(q)\,\ddot{q} + \gamma_c(q,\dot{q}) = 0$$

$$\longrightarrow$$

$$L(\ddot{q}, \lambda_c) = \underbrace{\frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}}_{\text{cost function}} - \overbrace{\lambda_c^\top}^{\text{dual variable = contact forces}} \underbrace{\left(J_c(q)\ddot{q} + \gamma_c(q,\dot{q})\right)}_{\text{equality constraint}}$$

Solving the Lagrangian contact problem

$$\min_{\ddot{q}} \quad \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}$$
$$J_c(q)\,\ddot{q} + \gamma_c(q,\dot{q}) = 0$$

$$L(\ddot{q},\lambda_c) = \underbrace{\frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}}_{\text{cost function}} - \overbrace{\lambda_c^\top}^{\text{dual variable = contact forces}} \underbrace{\left(J_c(q)\ddot{q} + \gamma_c(q,\dot{q})\right)}_{\text{equality constraint}}$$

The **KKT conditions** of the QP problem are given by:

$$\nabla_{\ddot{q}} L = M(q)(\ddot{q} - \ddot{q}_f) - J_c(q)^\top \lambda_c = 0 \qquad \text{Joint space force propagation}$$
$$\nabla_{\lambda_c} L = J_c(q)\ddot{q} + \gamma_c(q,\dot{q}) = 0 \qquad \text{Contact acceleration constraint}$$

Solving the Lagrangian contact problem

$$\min_{\ddot{q}} \quad \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}$$
$$J_c(q)\,\ddot{q} + \gamma_c(q,\dot{q}) = 0$$

$$L(\ddot{q}, \lambda_c) = \underbrace{\frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}}_{\text{cost function}} - \overbrace{\lambda_c^\top}^{\text{dual variable = contact forces}} \underbrace{\left(J_c(q)\ddot{q} + \gamma_c(q,\dot{q})\right)}_{\text{equality constraint}}$$

The **KKT conditions** of the QP problem are given by:

$$\nabla_{\ddot{q}} L = M(q)(\ddot{q} - \ddot{q}_f) - J_c(q)^\top \lambda_c = 0$$ Joint space force propagation

$$\nabla_{\lambda_c} L = J_c(q)\ddot{q} + \gamma_c(q,\dot{q}) = 0$$ Contact acceleration constraint

leading to the so-called **KKT dynamics**:

$$\underbrace{\begin{bmatrix} M(q) & J_c^\top(q) \\ J_c(q) & 0 \end{bmatrix}}_{K(q)} \begin{bmatrix} \ddot{q} \\ -\lambda_c \end{bmatrix} = \begin{bmatrix} M(q)\ddot{q}_f \\ -\gamma_c(q,\dot{q}) \end{bmatrix}$$

Solving the Lagrangian contact problem

$$\min_{\ddot{q}} \quad \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}$$
$$J_c(q)\,\ddot{q} + \gamma_c(q,\dot{q}) = 0$$

$$L(\ddot{q}, \lambda_c) = \underbrace{\frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)}}_{\text{cost function}} - \overbrace{\lambda_c^\top}^{\text{dual variable = contact forces}} \underbrace{\big(J_c(q)\ddot{q} + \gamma_c(q,\dot{q})\big)}_{\text{equality constraint}}$$

The **KKT conditions** of the QP problem are given by:

$$\nabla_{\ddot{q}} L = M(q)(\ddot{q} - \ddot{q}_f) - J_c(q)^\top \lambda_c = 0$$ Joint space force propagation

$$\nabla_{\lambda_c} L = J_c(q)\ddot{q} + \gamma_c(q,\dot{q}) = 0$$ Contact acceleration constraint

leading to the so-called **KKT dynamics**:

$$\underbrace{\begin{bmatrix} M(q) & J_c^\top(q) \\ J_c(q) & 0 \end{bmatrix}}_{K(q)} \begin{bmatrix} \ddot{q} \\ -\lambda_c \end{bmatrix} = \begin{bmatrix} M(q)\ddot{q}_f \\ -\gamma_c(q,\dot{q}) \end{bmatrix}$$

BUT, there might be <u>one solution</u>, <u>redundant solutions</u> or <u>no solution</u> at all depending on **the rank of** $J_c(q)$.

Classic resolution

We can analytically inverse the system to obtain the solution in **3 main steps**:

$$\boxed{M(q)\ddot{q} - J_c(q)^\top \lambda_c = M(q)\ddot{q}_f}$$

$$\boxed{J_c(q)\ddot{q} + \gamma_c(q, \dot{q}) = 0}$$

Classic resolution

1 - Express $\ddot{q}$ as function of $\ddot{q}_f$ and λ_c

$$\boxed{\ddot{q} = \ddot{q}_f + M^{-1}(q)J_c(q)^\top \lambda_c}$$

We can analytically inverse the system to obtain the solution in **3 main steps**:

$$\boxed{M(q)\ddot{q} - J_c(q)^\top \lambda_c = M(q)\ddot{q}_f}$$

$$\boxed{J_c(q)\ddot{q} + \gamma_c(q, \dot{q}) = 0}$$

Classic resolution

We can analytically inverse the system to obtain the solution in **3 main steps**:

$$M(q)\ddot{q} - J_c(q)^\top \lambda_c = M(q)\ddot{q}_f$$

$$J_c(q)\ddot{q} + \gamma_c(q, \dot{q}) = 0$$

1 - Express $\ddot{q}$ as function of $\ddot{q}_f$ and λ_c

$$\ddot{q} = \ddot{q}_f + M^{-1}(q)J_c(q)^\top \lambda_c$$

2 - Replace $\ddot{q}$ and get an expression depending only on λ_c

$$\underbrace{J_c(q)M^{-1}(q)J_c(q)^\top}_{\Lambda_c^{-1}(q)} \lambda_c + \underbrace{J_c(q)\ddot{q}_f + \gamma_c(q, \dot{q})}_{a_{c,f}(q,\dot{q},\ddot{q}_f)} = 0$$

$\Lambda_c^{-1}(q)$: Delassus matrix
Inverse Operational Space Inertia Matrix

$a_{c,f}(q,\dot{q},\ddot{q}_f)$: Free contact acceleration

Classic resolution

We can analytically inverse the system to obtain the solution in **3 main steps**:

$$\boxed{M(q)\ddot{q} - J_c(q)^\top \lambda_c = M(q)\ddot{q}_f}$$

$$\boxed{J_c(q)\ddot{q} + \gamma_c(q, \dot{q}) = 0}$$

1 - Express $\ddot{q}$ as function of $\ddot{q}_f$ and λ_c

$$\boxed{\ddot{q} = \ddot{q}_f + M^{-1}(q)J_c(q)^\top \lambda_c}$$

2 - Replace $\ddot{q}$ and get an expression depending only on λ_c

$$\underbrace{J_c(q)M^{-1}(q)J_c(q)^\top}_{\Lambda_c^{-1}(q)} \lambda_c + \underbrace{J_c(q)\ddot{q}_f + \gamma_c(q, \dot{q})}_{a_{c,f}(q,\dot{q},\ddot{q}_f)} = 0$$

Delassus matrix
Inverse Operational Space Inertia Matrix

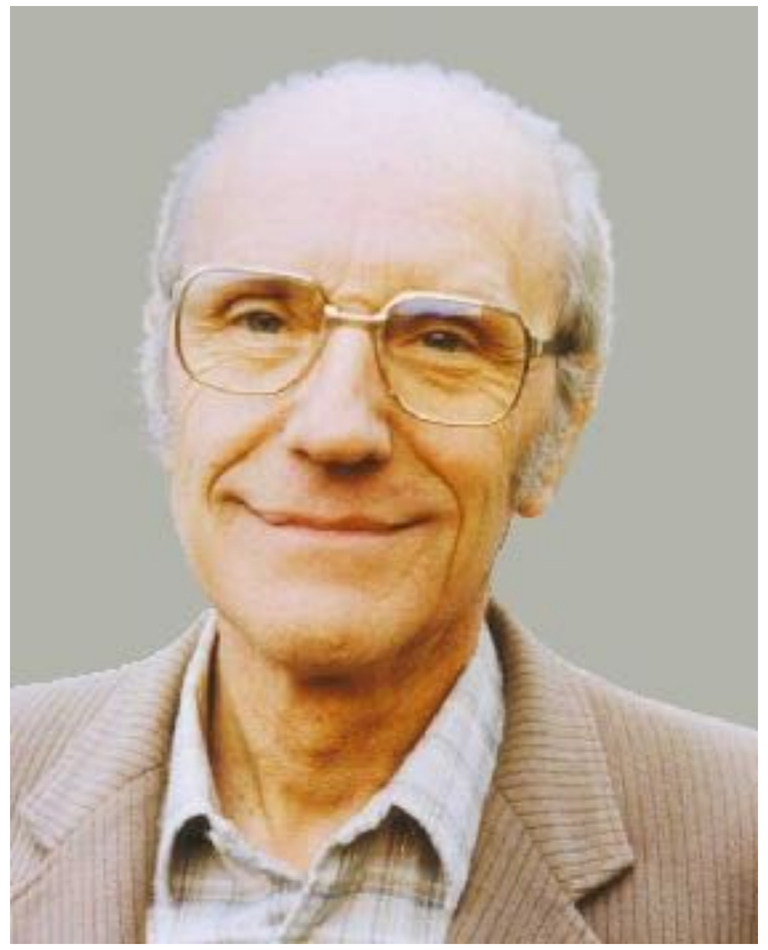
Free contact acceleration

3 - Inverse Λ_c^{-1} and find the optimal λ_c

$$\boxed{\lambda_c = -\Lambda_c(q)\, a_{c,f}(q, \dot{q}, \ddot{q}_f)}$$

The Proximal Rigid Contact Problem

bilateral contacts



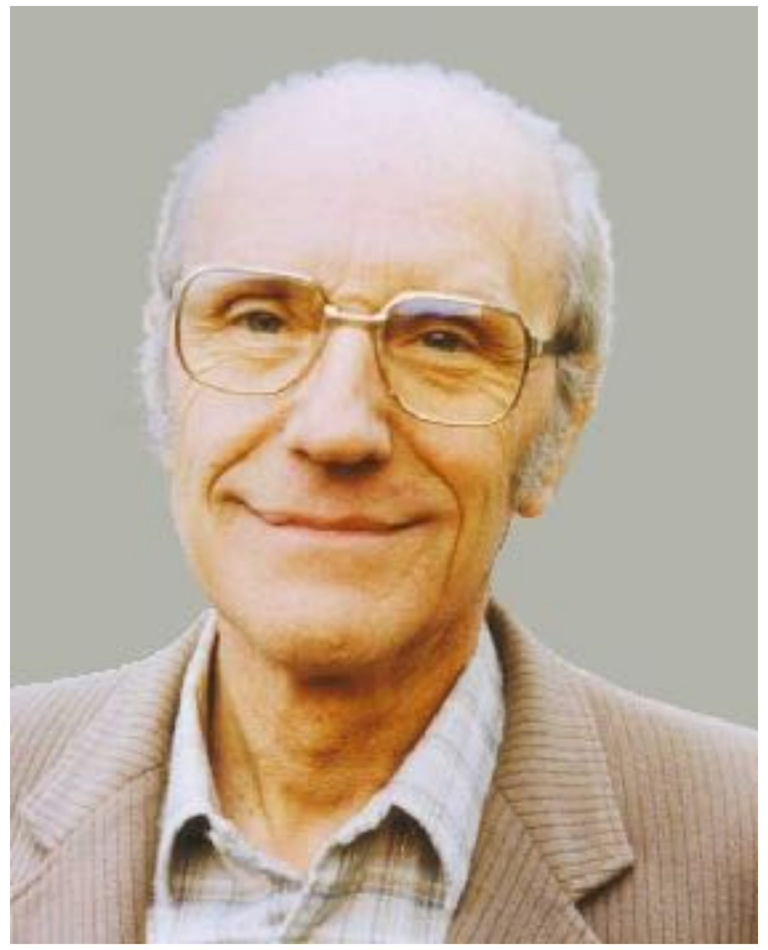
Jean-Jacques Moreau

The proximal settings

The **proximal operator** of a convex function $f(x)$ is given by:

$$\mathbf{prox}_{f,\alpha}(y) \stackrel{\mathbf{def}}{=} \mathbf{arg\,min}_{x \in \mathcal{X}}\, f(x) + \frac{\alpha}{2}\|x - y\|_2^2$$

where α can be assimilated to the inverse of a step size.



Jean-Jacques Moreau

The proximal settings

The **proximal operator** of a convex function $f(x)$ is given by:

$$\mathbf{prox}_{f,\alpha}(y) \stackrel{\mathbf{def}}{=} \underset{x \in \mathcal{X}}{\mathbf{arg\,min}}\; f(x) + \frac{\alpha}{2}\|x - y\|_2^2$$

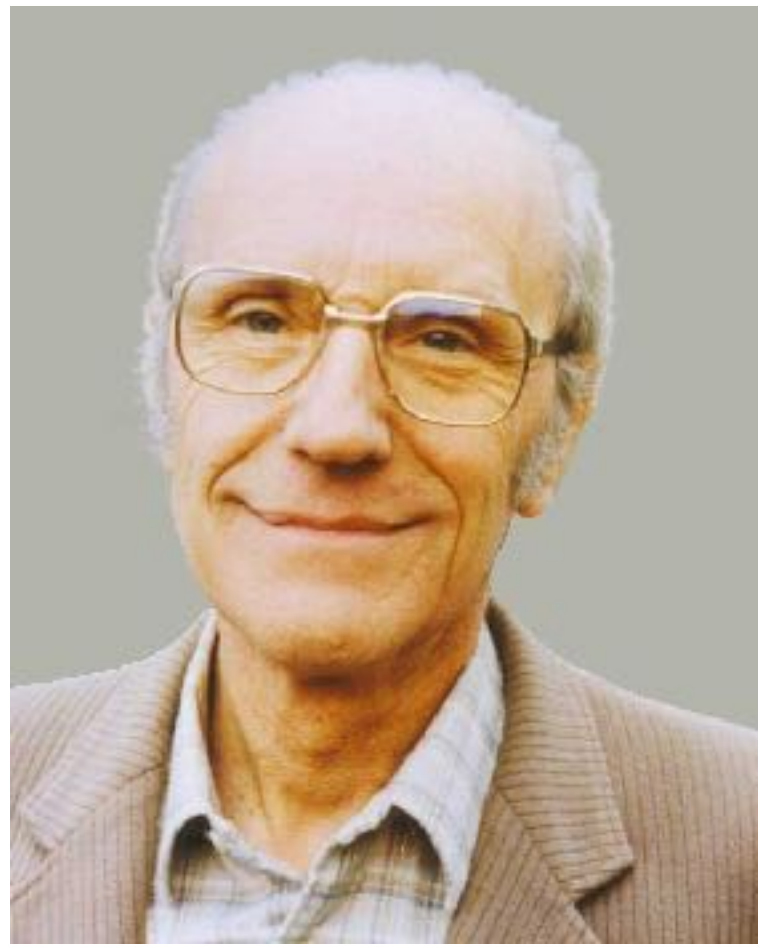
where α can be assimilated to the inverse of a step size.

Proximal algorithms typically iterate over the proximal operators, following the recursion:

$$x_{k+1} = \mathbf{prox}_{f,\alpha}(x_k)$$

In general, this results in a **cascade of simpler problems** to solve, at the price of <u>possibly requiring a large number of iterations</u> before converging to the solution of the original problem with a desired precision.

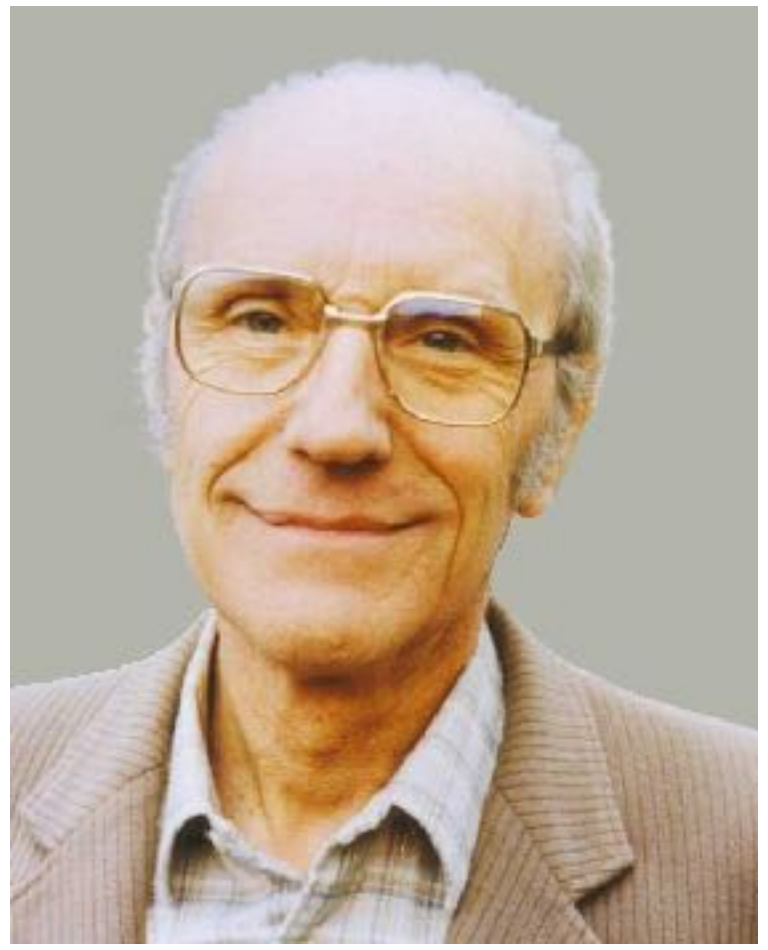
Smoothing the Lagrangian



Jean-Jacques Moreau

The solution is to add **an extra smoothing term** to the Lagrangian, similarly to proximal algorithms:

$$L_{\mu}(\ddot{q}, \lambda_c \,|\, \lambda_c^{-}) = \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)} + \lambda_c^{\top}\left(J_c(q)\ddot{q} + \gamma_c(q, \dot{q})\right) - \frac{\mu}{2}\|\lambda_c - \lambda_c^{-}\|_2^2$$



Jean-Jacques Moreau

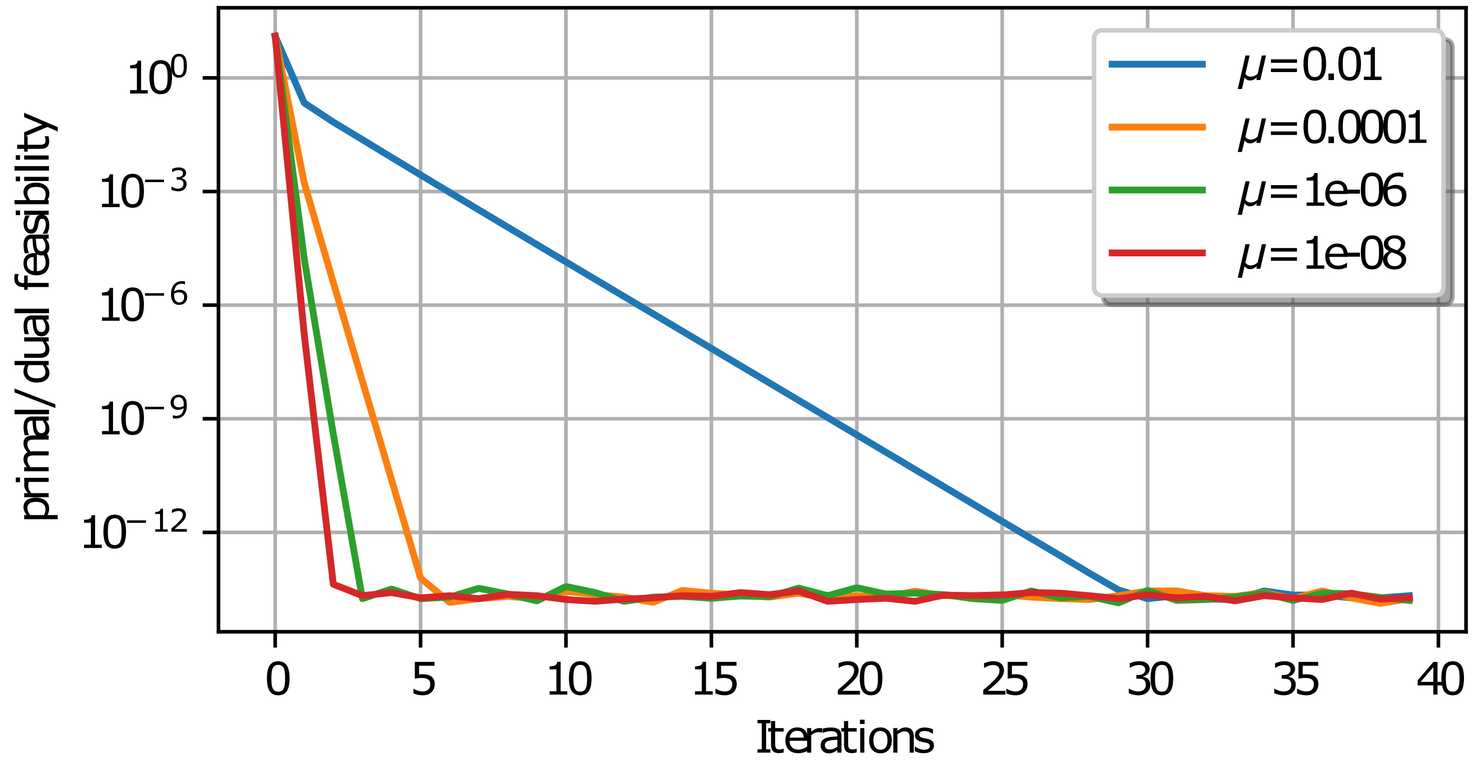
Smoothing the Lagrangian

The solution is to add **an extra smoothing term** to the Lagrangian, similarly to proximal algorithms:

$$L_\mu(\ddot{q}, \lambda_c \,|\, \lambda_c^-) = \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)} + \lambda_c^\top \left(J_c(q)\ddot{q} + \gamma_c(q,\dot{q})\right) - \frac{\mu}{2}\|\lambda_c - \lambda_c^-\|_2^2$$

which has the strong effect of making **KKT dynamics well posed**:

$$\underbrace{\begin{bmatrix} M(q) & J_c(q)^\top \\ J_c(q) & -\mu I \end{bmatrix}}_{K_\mu(q)} \begin{bmatrix} \ddot{q} \\ \lambda_c \end{bmatrix} = \begin{pmatrix} M(q)\ddot{q}_f \\ -\gamma_c(q,\dot{q}) - \mu\lambda_c^- \end{pmatrix}$$



converging to the least constraint solution if the **problem is not feasible**.

Explicit proximal contact solution

We can analytically inverse the system to obtain the solution in **3 main steps**:

$$\boxed{M(q)\ddot{q} - J_c(q)^\top \lambda_c = M(q)\ddot{q}_f}$$

$$\boxed{J_c(q)\ddot{q} + \gamma_c(q,\dot{q}) = -\mu(\lambda_c - \lambda_c^-)}$$

Explicit proximal contact solution

1 - Express $\ddot{q}$ as function of $\ddot{q}_f$ and λ_c

$$\ddot{q} = \ddot{q}_f + M^{-1}(q)J_c(q)^\top \lambda_c$$

We can analytically inverse the system to obtain the solution in **3 main steps**:

$$M(q)\ddot{q} - J_c(q)^\top \lambda_c = M(q)\ddot{q}_f$$

$$J_c(q)\ddot{q} + \gamma_c(q, \dot{q}) = -\mu(\lambda_c - \lambda_c^-)$$

Explicit proximal contact solution

We can analytically inverse the system to obtain the solution in **3 main steps**:

$$\boxed{M(q)\ddot{q} - J_c(q)^\top \lambda_c = M(q)\ddot{q}_f}$$

$$\boxed{J_c(q)\ddot{q} + \gamma_c(q, \dot{q}) = -\mu(\lambda_c - \lambda_c^-)}$$

1 - Express $\ddot{q}$ as function of $\ddot{q}_f$ and λ_c

$$\boxed{\ddot{q} = \ddot{q}_f + M^{-1}(q)J_c(q)^\top \lambda_c}$$

2 - Replace $\ddot{q}$ and get an expression depending only on λ_c

$$\underbrace{\left(J_c(q)M^{-1}(q)J_c(q)^\top + \mu I\right)}_{\Lambda^{-1}_{c,\mu}(q)} \lambda_c + \underbrace{J_c(q)\ddot{q}_f + \gamma_c(q, \dot{q})}_{a_{c,f}(q,\dot{q},\ddot{q}_f)} = \mu \lambda_c^-$$

damped Delassus' matrix

Explicit proximal contact solution

We can analytically inverse the system to obtain the solution in **3 main steps**:

$$\boxed{M(q)\ddot{q} - J_c(q)^\top \lambda_c = M(q)\ddot{q}_f}$$

$$\boxed{J_c(q)\ddot{q} + \gamma_c(q, \dot{q}) = -\mu(\lambda_c - \lambda_c^-)}$$

1 - Express $\ddot{q}$ as function of $\ddot{q}_f$ and λ_c

$$\boxed{\ddot{q} = \ddot{q}_f + M^{-1}(q) J_c(q)^\top \lambda_c}$$

2 - Replace $\ddot{q}$ and get an expression depending only on λ_c

$$\underbrace{\left(J_c(q) M^{-1}(q) J_c(q)^\top + \mu I\right)}_{\Lambda^{-1}_{c,\mu}(q)} \lambda_c + \underbrace{J_c(q)\ddot{q}_f + \gamma_c(q, \dot{q})}_{a_{c,f}(q,\dot{q},\ddot{q}_f)} = \mu \lambda_c^-$$

damped Delassus' matrix

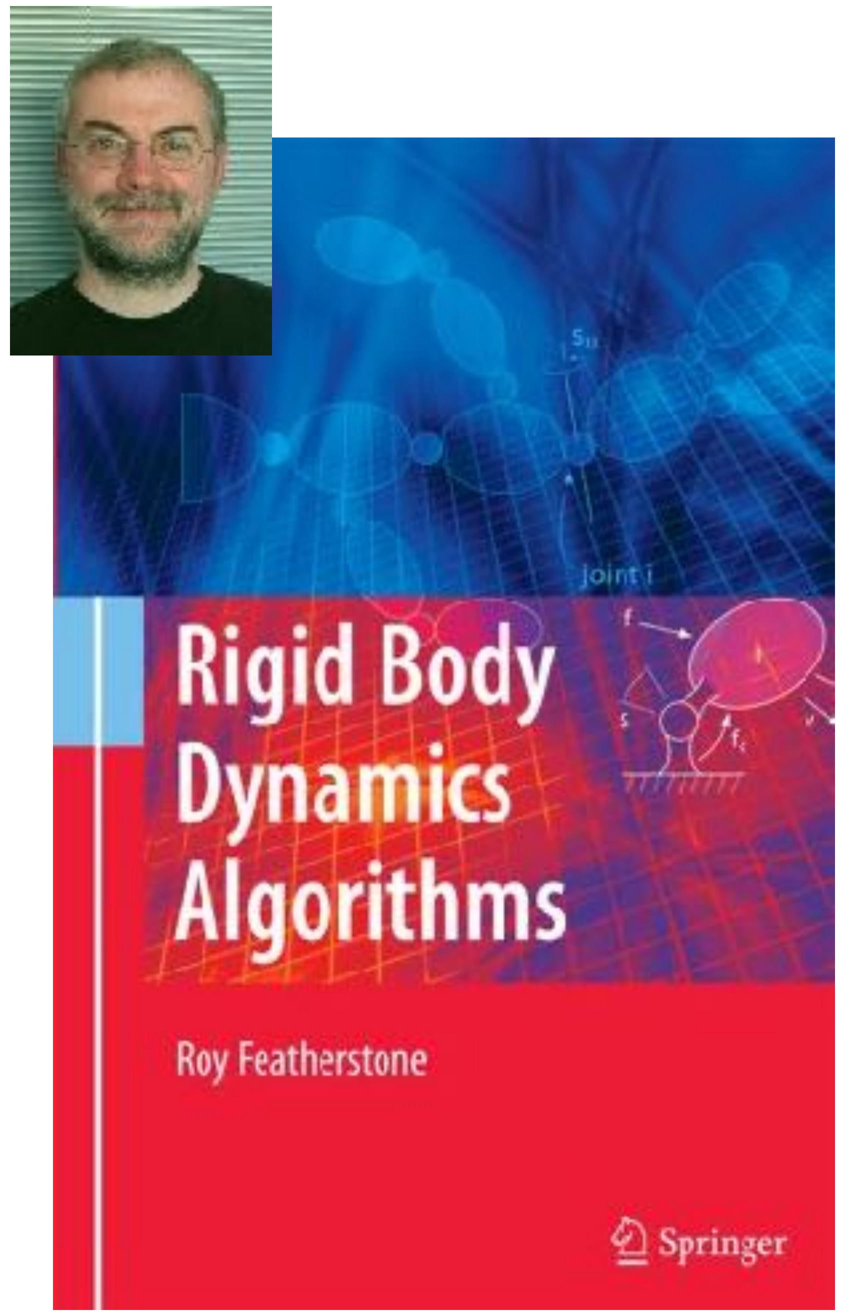
3 - Inverse $\Lambda^{-1}_{c,\mu}(q)$ and find the optimal λ_c

$$\boxed{\lambda_c = -\Lambda_{c,\mu}(q)\left(a_{c,f}(q, \dot{q}, \ddot{q}_f) - \mu \lambda_c^-\right)}$$

Sparse resolution of the Rigid Contact Problem

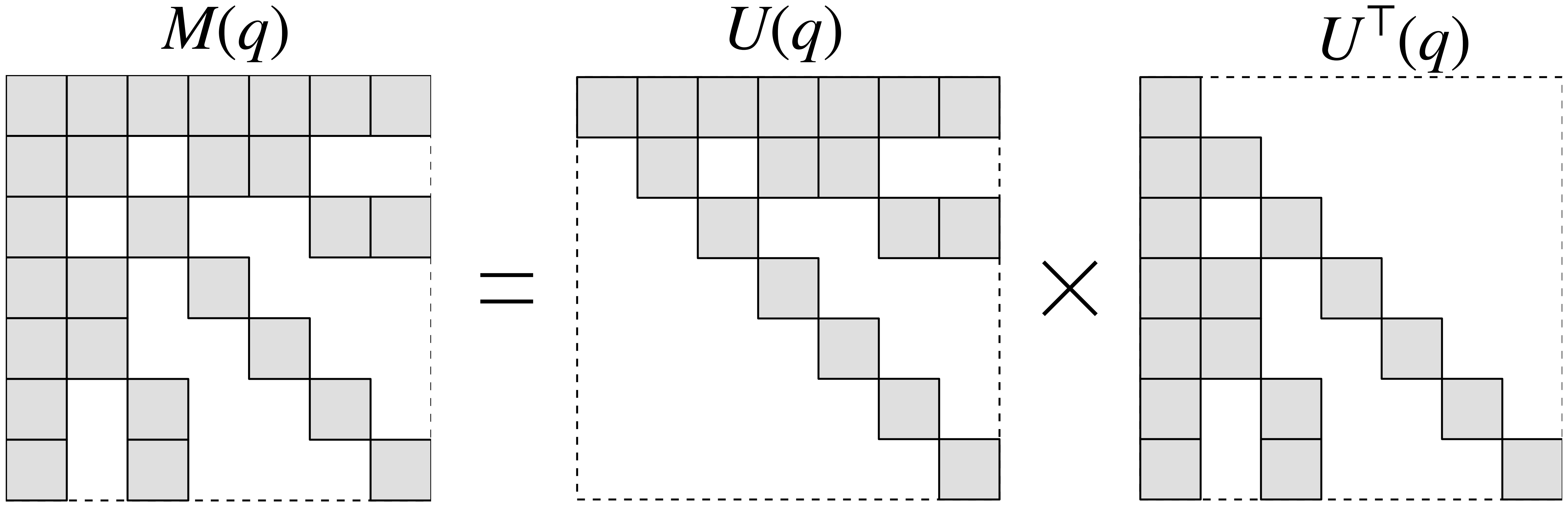
bilateral contacts

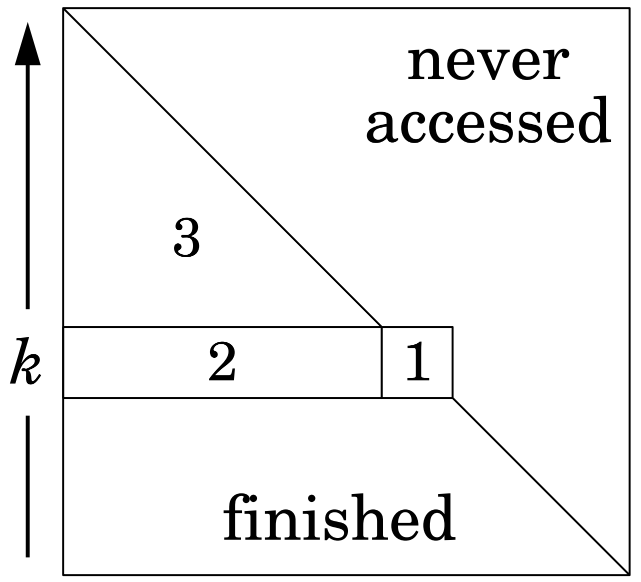
Mass Matrix: sparse Cholesky factorization



Goal: compute $\Lambda_c^{-1}(q) \stackrel{\text{def}}{=} J_c(q)M^{-1}(q)J_c^{\top}(q)$ without computing $M^{-1}(q)$

Solution: exploiting **the sparsity** in the Cholesky factorization of $M(q)$





Cholesky factorization

1. $U_{k,k} = \sqrt{M_{k,k}}$
2. $U_{k,i} = M_{k,i}/U_{k,k}$
3. $U_{i,j} = M_{i,j} - U_{k,i}\,U_{k,j}$

The total complexity is $O(N^2)$ instead of $O(N^3)$
when using a **dense** Cholesky decomposition

Sparse Contact Matrix Decomposition

The goal is to **exploit and reserve the sparsity** in the factorization of the KKT matrix $K_\mu(q)$

Instead of working with:

$$\begin{bmatrix} M(q) & J_c(q)^\top \\ J_c(q) & -\mu I \end{bmatrix} \longrightarrow$$

we gonna work with:

$$\begin{bmatrix} -\mu I & J_c(q) \\ J_c(q)^\top & M(q) \end{bmatrix}$$

Sparse Contact Matrix Decomposition

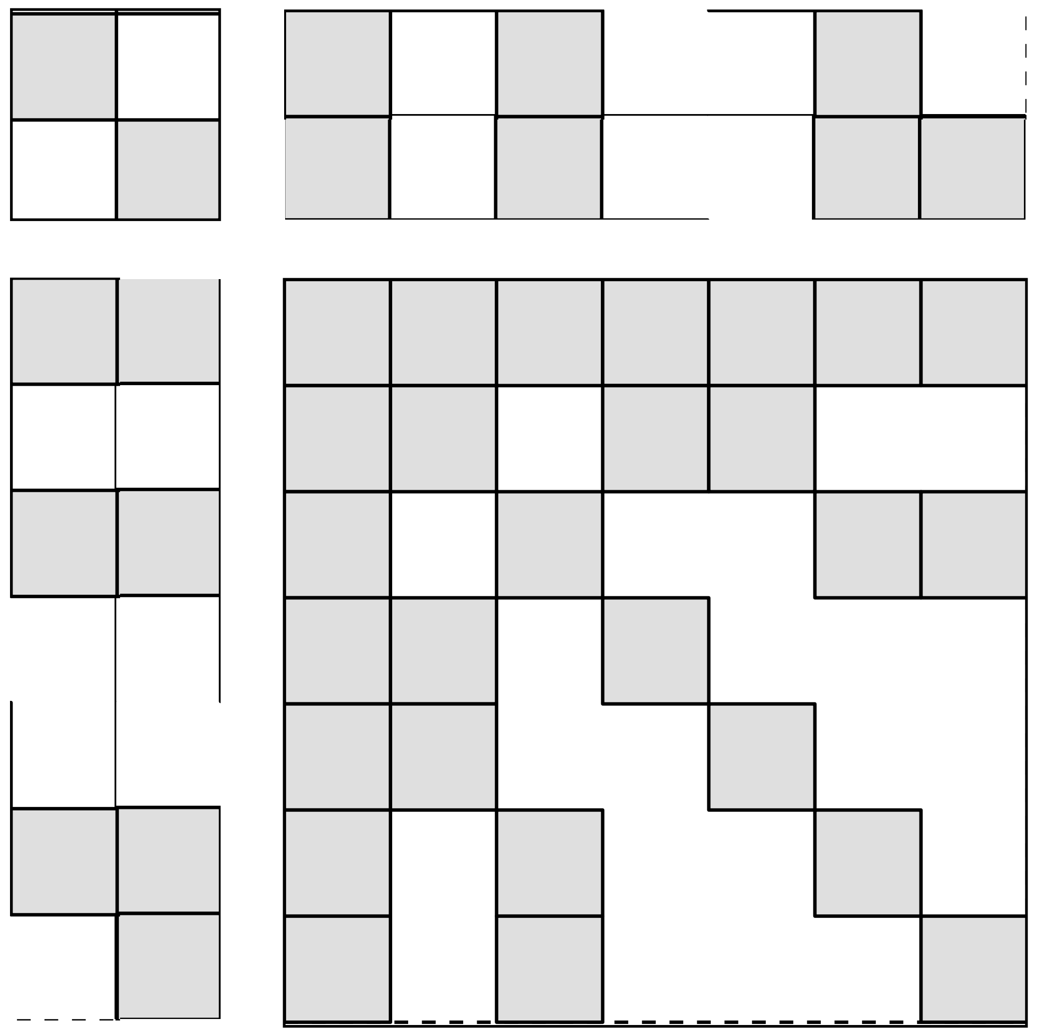
The goal is to **exploit and reserve the sparsity** in the factorization of the KKT matrix $K_{\mu}(q)$

Instead of working with:

$$\begin{bmatrix} M(q) & J_c(q)^{\top} \\ J_c(q) & -\mu I \end{bmatrix}$$

we gonna work with:

$$\begin{bmatrix} -\mu I & J_c(q) \\ J_c(q)^{\top} & M(q) \end{bmatrix}$$



The total complexity remains low in $O((N+N_c)^2)$ instead of $O((N+N_c)^3)$ when using dense Cholesky decomposition

Pseudo-code of the Cholesky factorization of A:

```
for k = N to 1 do
    i = p(k)
    while i > 0 do
        a = A_{i+m,k+m} / A_{k+m,k+m}
        j = i
        Pass over the joints
        while j > 0 do
            A_{j+m,i+m} = A_{j+m,i+m} - A_{j+m,k+m} a
            j = p(j)
        end
        Pass over the constraints
        for l = n_c to 1 do
            if i ∈ σ_l
                for j = n_i to 1 do
                    A_{j+i_l,i+m} = A_{j+i_l,i+m} - A_{j+i_l,k+m} a
                end
            end
        end
        A_{i+m,k+m} = a
        i = p(i)
    end
end
Dense factorization related to the OSIM
for l = n_c to 1 do
    for k̃ = n_i to 1 do
        k = i_l + k̃
        for i = k - 1 to 1 do
            a = A_{i,k} / A_{k,k}
            for j = i to 1 do
                A_{j,i} = A_{j,i} - A_{j,k} a
            end
            A_{i,k} = a
        end
    end
end
```

Looking at the KKT inverse

From the inverse of the KKT matrix, we can directly retrieve a lot of by-product and useful quantities:

$$K_\mu(q) = \begin{bmatrix} -\mu I & J_c(q) \\ J_c(q)^\top & M(q) \end{bmatrix} \longrightarrow K_\mu^{-1}(q) = \begin{bmatrix} -\overbrace{(J_c M^{-1} J_c^\top + \mu I)^{-1}}^{\Lambda_\mu} & \Lambda_\mu J_c M^{-1} \\ \left(\Lambda_\mu J_c M^{-1}\right)^\top & -M^{-1} - M^{-1} J_c^\top \Lambda_\mu J_c M^{-1} \end{bmatrix}$$

Cholesky decomposition

$$K_\mu = \underbrace{\begin{bmatrix} U_{\Lambda_\mu^{-1}} & J_c U_{\Lambda_M}^{-\top} D_M^{-1} \\ 0 & U_M \end{bmatrix}}_{U_{K_\mu}} \underbrace{\begin{bmatrix} -D_{\Lambda_\mu^{-1}} & 0 \\ 0 & D_M \end{bmatrix}}_{D_{K_\mu}} \underbrace{\begin{bmatrix} U_{\Lambda_\mu^{-1}}^\top & 0 \\ D_M^{-1} U_M^{-1} J_c^\top & U_M^\top \end{bmatrix}}_{U_{K_\mu}^\top}$$

Proximal and sparse resolution

Proximal and Sparse Resolution of Constrained Dynamic Equations

Author Names Omitted for Anonymous Review. Paper-ID 56

***Abstract*—Control of robots with kinematic constraints like loop-closure constraints or interactions with the environment require solving the underlying constrained dynamics equations of motion. Several approaches have been proposed so far in the literature to solve these constrained optimization problems, for instance by either taking advantage in part of the sparsity of the kinematic tree or by considering an explicit formulation of the constraints in the problem resolution. Yet, not all the constraints allow an explicit formulation and in general, approaches of the state of the art suffer from singularity issues, especially in the context of redundant or singular constraints. In this paper, we propose a unified approach to solve forward or inverse dynamics equations involving constraints in an efficient, generic and robust manner. To this aim, we first (i) propose a proximal formulation of the constrained dynamics which converges to an optimal solution in the least-square sense even in the presence of singularities. Based on this proximal formulation, we introduce (ii) a sparse Cholesky factorization of the underlying Karush–Kuhn–Tucker matrix related to the constrained dynamics, which exploits at best the sparsity of the kinematic structure of the robot. We also show (iii) that it is possible to extract from this factorization the Cholesky decomposition associated to the so-called Operational Space Inertia Matrix, inherent to task-based control frameworks or physic simulations. These new formulation and factorization are implemented and benchmark on various robotic platforms, ranging from classic robotic arms or quadrupeds to humanoid robots with closed kinematic chains, and show how they significantly outperform alternative solutions of the state of the art.**

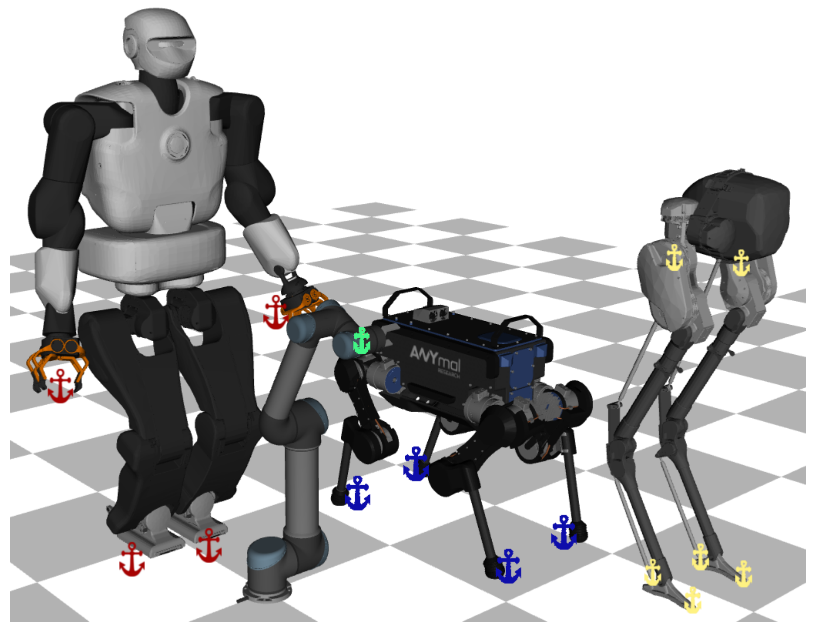


Fig. 1. Robotic systems may be subject to different types of constraints: point contact constraints (quadrupeds), flat foot constraints (humanoids), closed kinematic chains (parallel robots, here the 4-bar linkages of Cassie) or even contact with the end effectors (any robot). Each colored "anchor" here shows a possible kinematic constraint applied on the dynamics of the robot. In this paper, we introduce a generic approach to handle all these types of constraints, contacts and kinematic closures, in a unified and efficient manner, even in the context of ill-posed or singular cases.

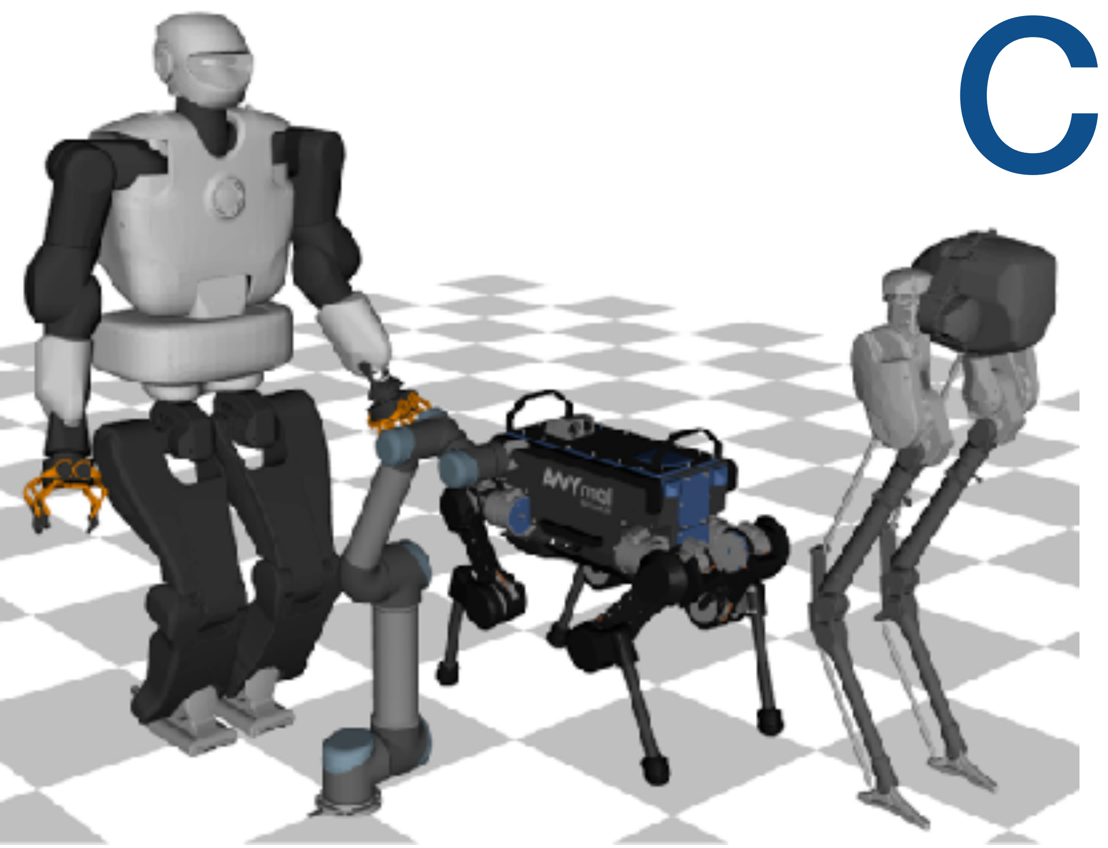
I. Introduction

As soon as a robot makes contacts with the world or is endowed with loop closures in its design, its dynamics is governed by the constrained equations of motion. From a phenomenological point of view, these equations of motion follow the so-called least-action principle, also known under the name of the Maupertuis principle which dates back to the 17^{th} century. This principle states that the motion of the system follows the closest possible acceleration to the free-falling acceleration (in the sense of the kinetic metric) which respects the constraints. In other words, solving the constrained equations of motion boils down to solving a constrained optimization problem where forces acts as the Lagrange multipliers of the motion constraints.

This fact has been has been recognized since [1], and our work largely takes inspiration from it. In this work, Baraff proposed to formulate the dynamics with maximal coordinates (i.e. each rigid body is represented by its 6 coordinates of motion) as a sparse constrained optimization problem, and proposed an algorithm to solve it in linear time. While maximal coordinates are interesting for their versatility and largely used in simulation [2], working directly in the configuration space with generalized coordinates presents several advantages [14] that we propose to exploit in this paper.

Some constraints can be put under an explicit form, i.e. there exists a reduced parametrization of the configuration that is free of constraints. This is often the case for classical kinematic closures [32, 14]. Yet explicit formulation is not always possible, and in particular is not possible for the common case of contact constraints [37]. We address here the more generic case where the constraints are written under an implicit form i.e. the configuration should nullify a set of equations, which makes it possible to handle any kind of design or contact constraints, or both together (see Fig. 1).

While recursive formulation exists and are predominant for unconstrained dynamics (both inverse [35, 31] and forward [11]) and for decomposing and inverting the joint-space inertia matrix (JSIM) [12], little effort seems to have been put in formulating a recursive algorithm in reduced coordinates for the constrained dynamics. Yet the constrained system is directly inverted in several trajectory-optimization implementation. In [6], the constrained dynamics is explicitly solved in the context of trajectory optimization. In [22], a hierarchical trajectory optimization is also inverting directly

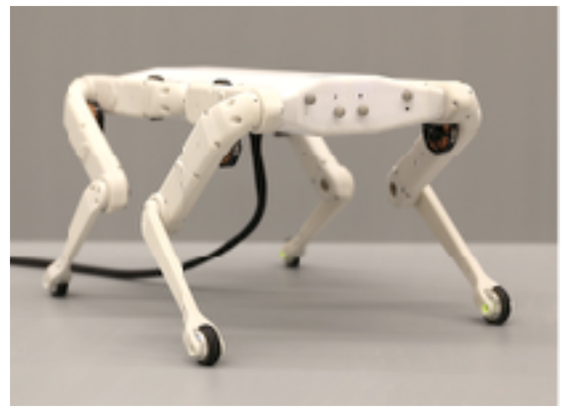


Cholesky decomposition timings

We benchmark the proposed Cholesky against classical approaches:

$$K_{\mu} = \underbrace{\begin{bmatrix} U_{\Lambda_{\mu}^{-1}} & J_c U_{\Lambda_M}^{-\top} D_M^{-1} \\ 0 & U_M \end{bmatrix}}_{U_{K_{\mu}}} \underbrace{\begin{bmatrix} -D_{\Lambda_{\mu}^{-1}} & 0 \\ 0 & D_M \end{bmatrix}}_{D_{K_{\mu}}} \underbrace{\begin{bmatrix} U_{\Lambda_{\mu}^{-1}}^{\top} & 0 \\ D_M^{-1} U_M^{-1} J_c^{\top} & U_M^{\top} \end{bmatrix}}_{U_{K_{\mu}}^{\top}}$$

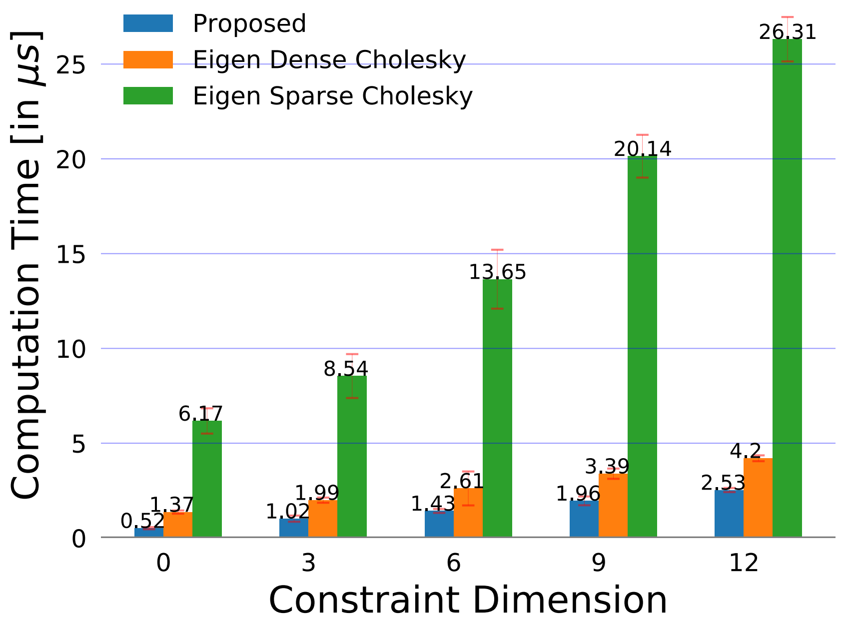
Solo 8

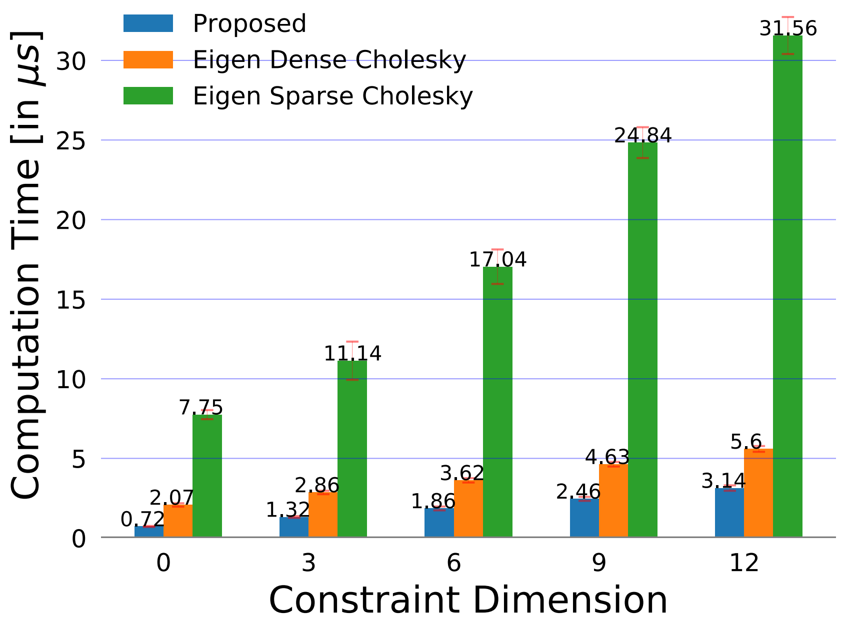


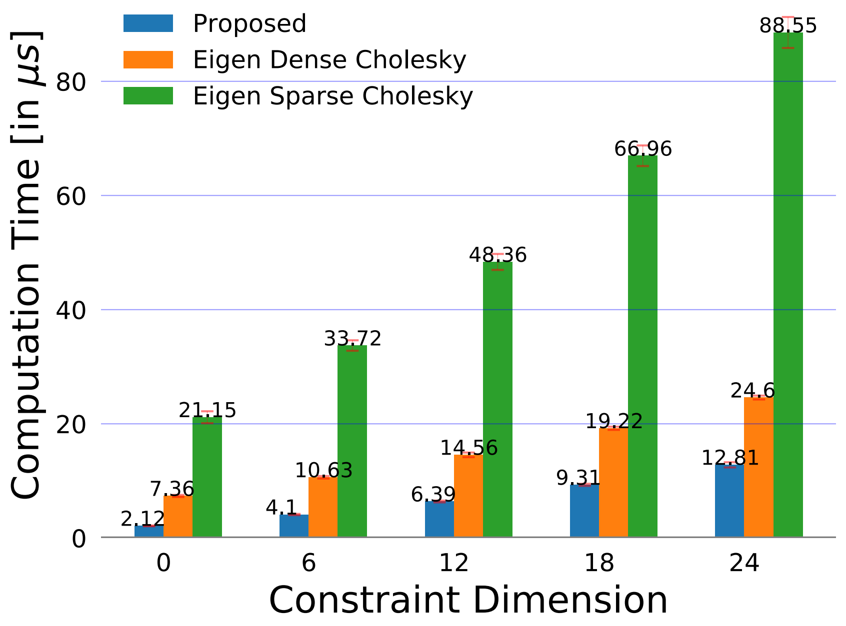
Anymal B

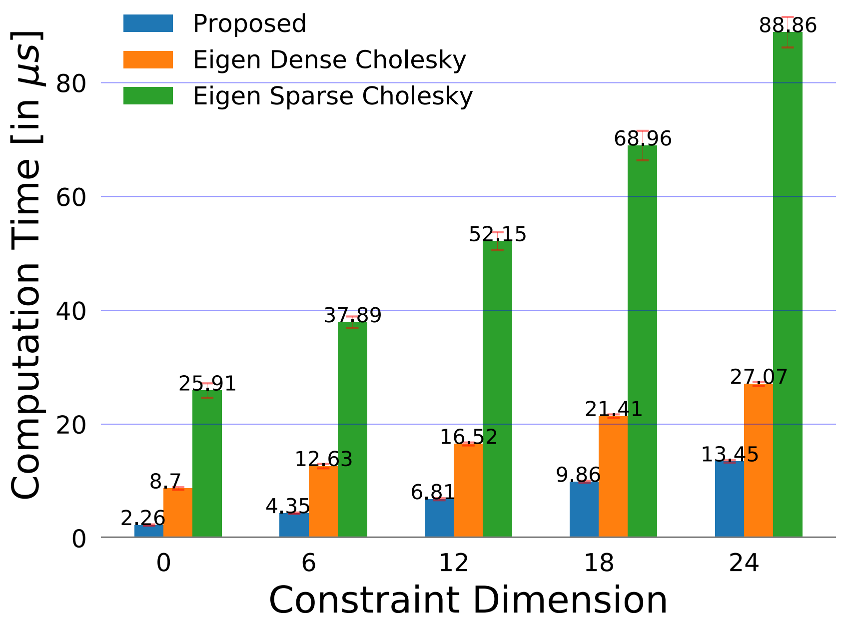
iCub

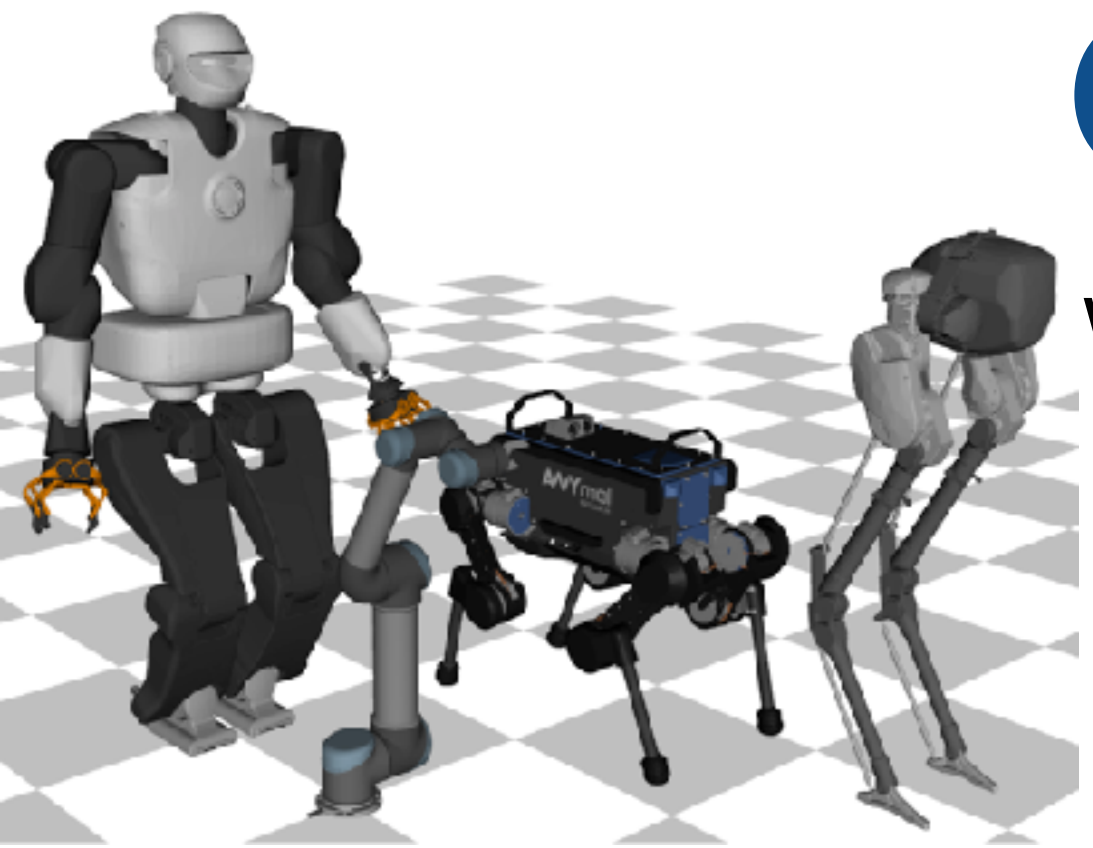
Talos











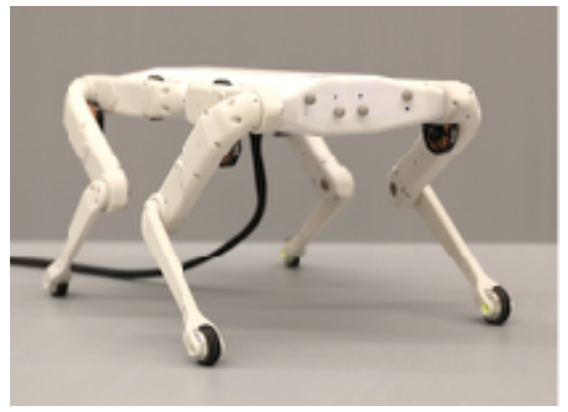
Constrained dynamics timings

We benchmark the constrained dynamics resolution against classical approaches:

$$\min_{\ddot{q}} \frac{1}{2}\|\ddot{q} - \ddot{q}_f\|^2_{M(q)} \qquad \text{where } \ddot{q}_f \stackrel{\text{def}}{=} M(q)^{-1}\big(\tau - C(q,\dot{q}) - G(q)\big)$$

$$a_c = J_c(q)\ddot{q} + \dot{J}_c(q,\dot{q})\dot{q}$$

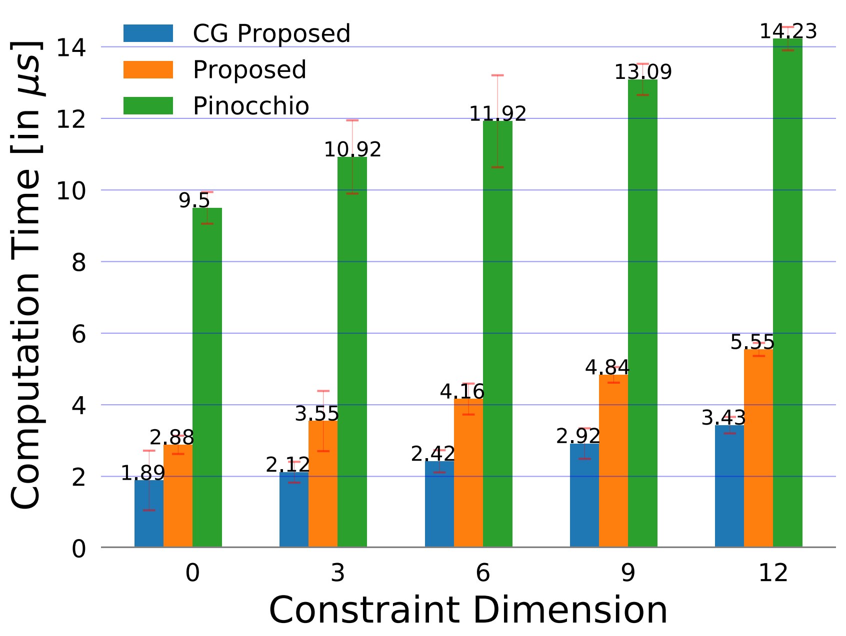
Solo 8

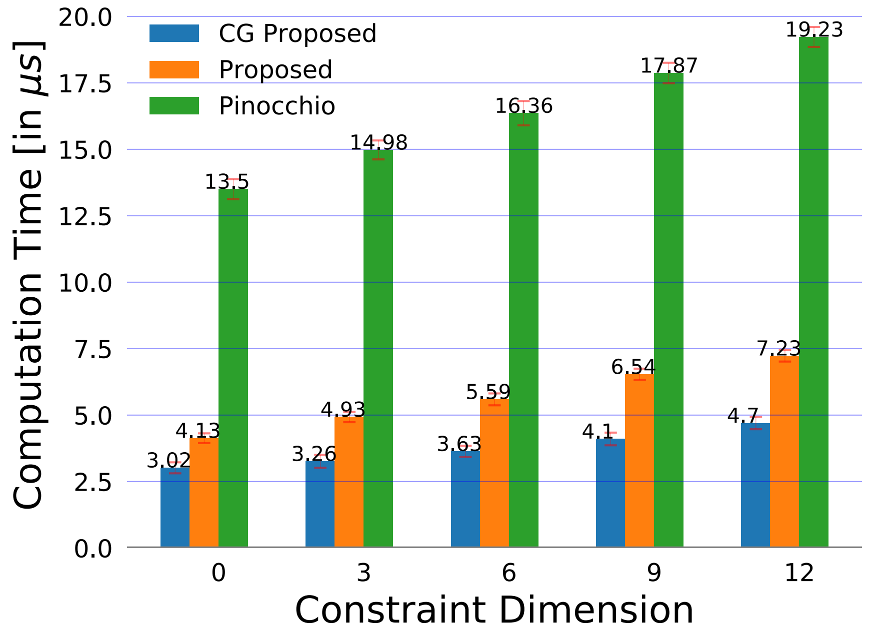


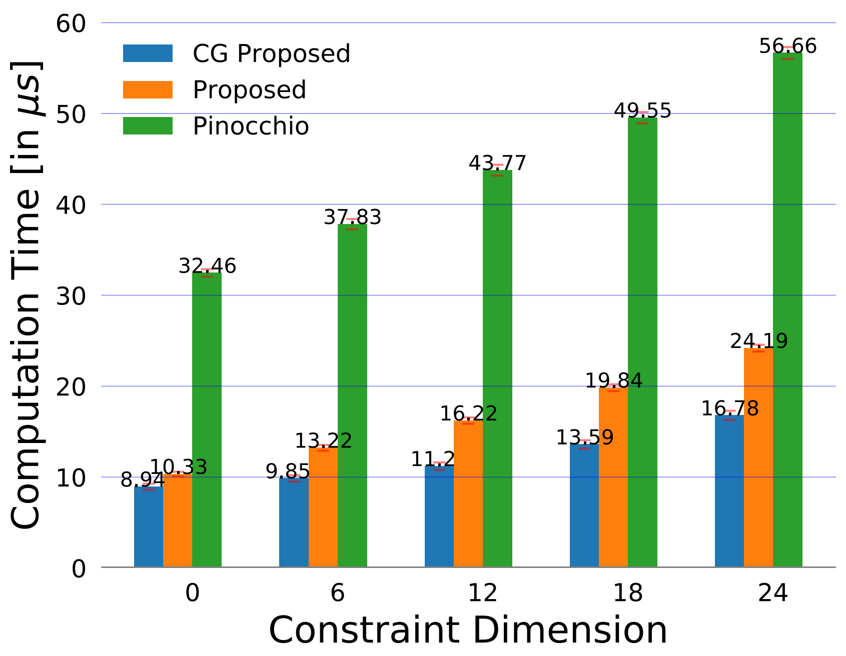
Anymal B

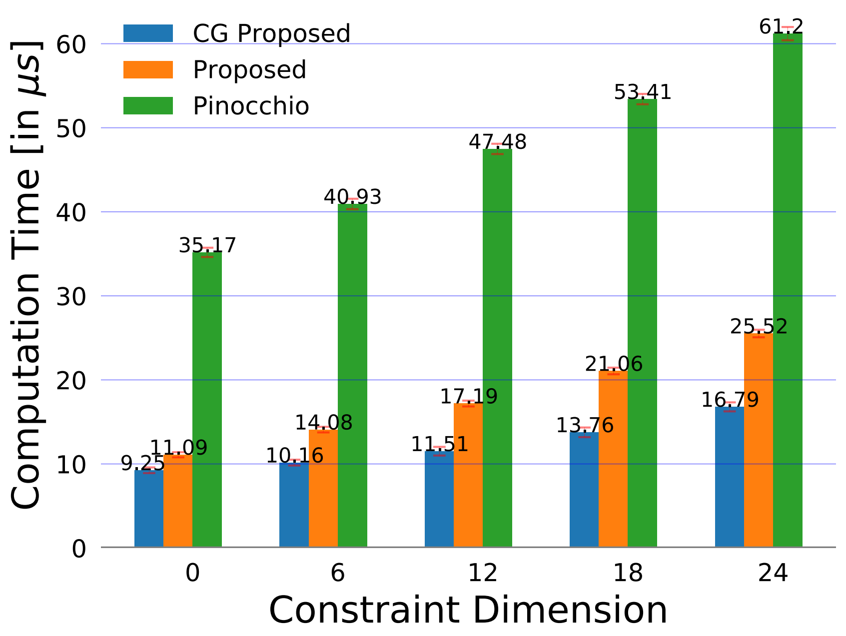
iCub

Talos









More to come during JNRH

- ✓ Presentation by **Louis Montaut** on Collision Detection and their differentiation
- ✓ Presentation by **Quentin Le Lidec** on Contact Models and Solvers in Robotics
- ✓ Presentation by **Bruce Wingo** on efficient inverse dynamics for closed-loop mechanisms