Robotic Systems II

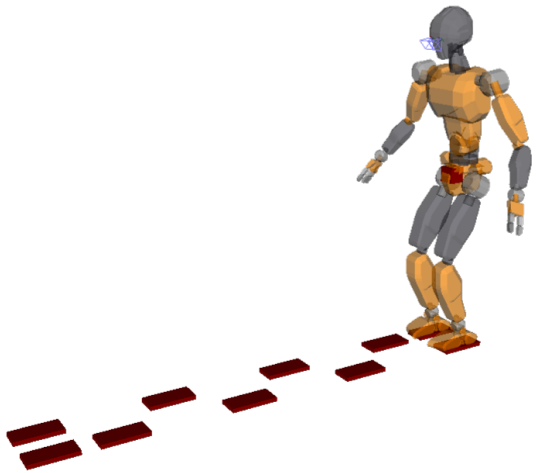
Lecture 9: Optimal Control for Robots with Multiple Limbs/End-Effectors

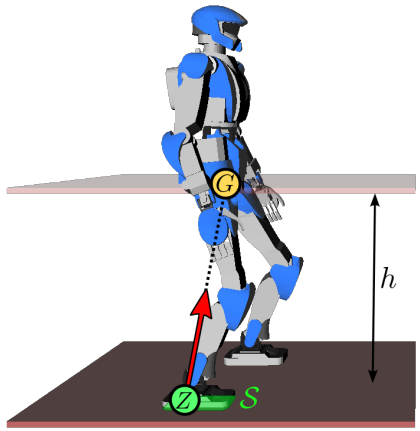
Konstantinos Chatzilygeroudis - costashatz@upatras.gr

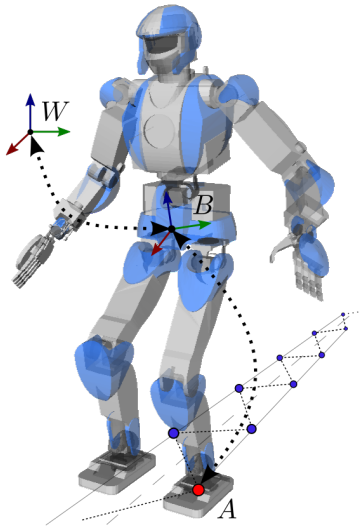
Department of Electrical and Computer Engineering
University of Patras

Template made by Panagiotis Papagiannopoulos

Control for Robots with Multiple Limbs







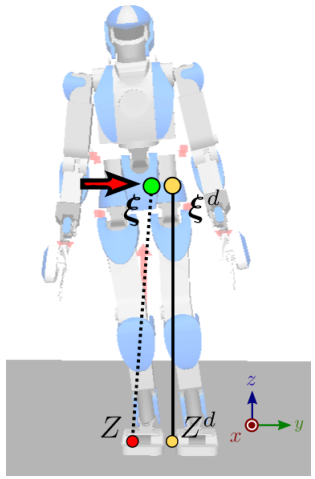
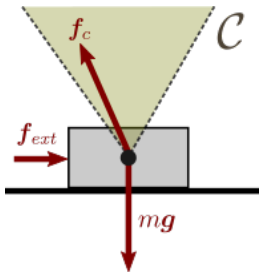


Image Credits: Stéphane Caron `https://scaron.info/`

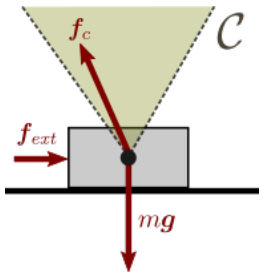
Contact Modelling

- Let's consider the 2D contact



From `https://scaron.info/`

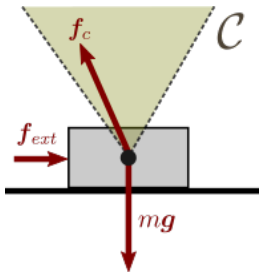
Contact Modelling



From `https://scaron.info/`

- Let's consider the 2D contact
- This contact is fixed while the force $\boldsymbol{f}_c = m\boldsymbol{g} - \boldsymbol{f}_{\text{ext}}$ remains inside the *friction cone* $\mathcal{C}$
- We say the contact is in *fixed mode*

Contact Modelling



From `https://scaron.info/`

- Let's consider the 2D contact
- This contact is fixed while the force $\boldsymbol{f}_c = m\boldsymbol{g} - \boldsymbol{f}_{\text{ext}}$ remains inside the *friction cone* $\mathcal{C}$
- We say the contact is in *fixed mode*
- When $\boldsymbol{f}_c \notin \mathcal{C}$, it switchs to the *sliding mode*
- *Rolling mode* when the body rotates over the ground/other body
- *Broken/free mode* when there is no contact force $\boldsymbol{f}_c$

Controlling with Contacts

- We usually want the contacts to be in **fixed mode**!
- Sliding contacts are hard to control
- Rolling contacts are also most of the time undesirable
- Free mode is what we already know!

Controlling with Contacts

- We usually want the contacts to be in **fixed mode**!
- Sliding contacts are hard to control
- Rolling contacts are also most of the time undesirable
- Free mode is what we already know!
- **How does this affect our formulations?!**

Friction Cones

- For each contact we have a force being applied at the contact position: $\boldsymbol{f}$
- We can split the force into the *normal* component, $\boldsymbol{f}_n$ and the *tangential* component, $\boldsymbol{f}_t$
- In 3D space we have 2 tangential components: $\boldsymbol{f}_t$ and $\boldsymbol{f}_b$
- For the contact to be fixed we need to have the following:

$$(\boldsymbol{f} \cdot \boldsymbol{n}) > 0, \text{ the force cannot pull/grasp!}$$
$$\|\boldsymbol{f} \cdot \boldsymbol{t}\|_2 \leqslant \mu(\boldsymbol{f} \cdot \boldsymbol{n})$$
$$\|\boldsymbol{f} \cdot \boldsymbol{b}\|_2 \leqslant \mu(\boldsymbol{f} \cdot \boldsymbol{n})$$

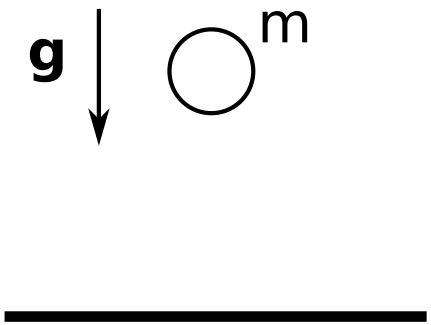
where μ is the Coulomb *static friction coefficient* at the contact, $\boldsymbol{n}$ is the normal direction and $\boldsymbol{t}, \boldsymbol{b}$ are the tangential directions of the contact.

Controlling with Contacts (2)

- We need additional constraints, i.e. to keep forces inside the friction cones!
- Discontinuous dynamics! **Why?**
- **How can we start solving these systems?**

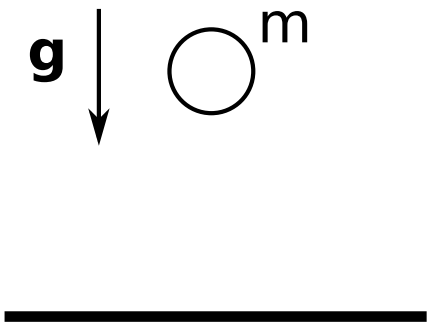
Ball Example

- Let's start by looking at a simple 2D example!
- How would the position and velocity over time graphs look like?



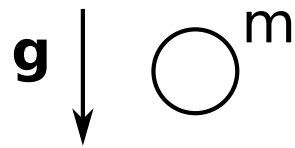
Ball Example

- Let's start by looking at a simple 2D example!
- How would the position and velocity over time graphs look like?
- In the air we have the nice Netwon equations $m\boldsymbol{a} = -m\boldsymbol{g}$
- What happens when the ball hits the ground?



Ball Example

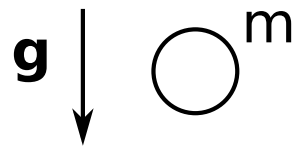
- Let's start by looking at a simple 2D example!
- How would the position and velocity over time graphs look like?
- In the air we have the nice Netwon equations $m\boldsymbol{a} = -m\boldsymbol{g}$



- What happens when the ball hits the ground?
- **Impact time is almost zero! Contact impacts are instantaneous!!**
- How would the position and velocity over time graphs look like now!?

Ball Example

- Let's start by looking at a simple 2D example!
- How would the position and velocity over time graphs look like?
- In the air we have the nice Netwon equations $m\boldsymbol{a} = -m\boldsymbol{g}$



- What happens when the ball hits the ground?
- **Impact time is almost zero! Contact impacts are instantaneous!!**
- How would the position and velocity over time graphs look like now!? Can't take derivatives of this!
- We cannot even simulate it numerically!

Contacts and Optimal Control

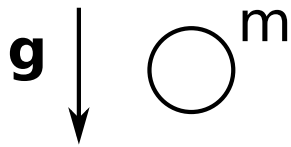
There are basically 2 ways of handling contacts:

1. **Hybrid/Event-based methods:**
   - We split the trajectory into segments
   - At the end of each segment, we have "contact event(s)"
   - Within the segments we have continuous regular dynamics
   - At the junctions (identified by a "guard function" or *pre-specified*), we define a **jump map** that "teleports" the state because of the discontinuous effects of the contact(s)
2. **Implicit/Time-stepping methods:**
   - We define a "signed distance function" $\phi(\boldsymbol{x}) \geqslant 0$ that describes the contacts (i.e. no penetration)
   - We solve a constrained optimization problem per time-step
   - We optimize contact forces explicitly with the rest of the state

Contacts and Optimal Control (2)

1. **Hybrid/Event-based methods:**
   - Can be very easily combined with non-linear trajectory optimization
   - Robust, works out of the box
   - Perfect for Locomotion
   - We need to prespecify the contact sequence!
   - Difficult to model grasping
2. **Implicit/Time-stepping methods:**
   - In theory, we automatically identify the optimal contact sequence
   - Very difficult optimization problems per time-step!

Let's go back to our ball example



1. **Hybrid/Event-based method:**

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \end{bmatrix}, \dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\boldsymbol{q}} \\ -\boldsymbol{g} \end{bmatrix}$$

$$g(\boldsymbol{x}) = q_z \geqslant 0 \quad \textit{"guard function"}$$

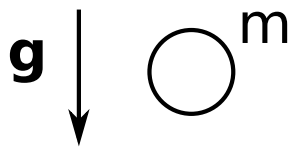
$$\boldsymbol{x}_{k+1} = j(\boldsymbol{x}_k) = \begin{bmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}}_x \\ -e\dot{\boldsymbol{q}}_z \end{bmatrix} \quad \textit{"jump map"}$$

e is the coefficient of restitution.

Let's go back to our ball example (2)

2 **Implicit/Time-stepping method:**

$$m\dot{\boldsymbol{q}} = -m\boldsymbol{g} + \boldsymbol{J}^T\boldsymbol{\lambda}$$
$$\phi(\boldsymbol{q}) = \boldsymbol{J}\boldsymbol{q}$$



We now do Backward Euler:

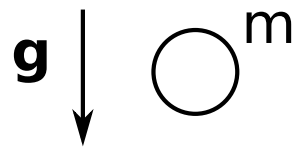
$$m\left(\frac{\dot{\boldsymbol{q}}_{k+1} - \dot{\boldsymbol{q}}_k}{dt}\right) = -m\boldsymbol{g} + \boldsymbol{J}^T\boldsymbol{\lambda}_k$$
$$\boldsymbol{q}_{k+1} = \boldsymbol{q}_k + \dot{\boldsymbol{q}}_{k+1}dt$$

And?

Let's go back to our ball example (2)

2 **Implicit/Time-stepping method:**

$$m\dot{\boldsymbol{q}} = -m\boldsymbol{g} + \boldsymbol{J}^T\boldsymbol{\lambda}$$
$$\phi(\boldsymbol{q}) = \boldsymbol{J}\boldsymbol{q}$$



We now do Backward Euler:

$$m\left(\frac{\dot{\boldsymbol{q}}_{k+1} - \dot{\boldsymbol{q}}_k}{dt}\right) = -m\boldsymbol{g} + \boldsymbol{J}^T\boldsymbol{\lambda}_k$$
$$\boldsymbol{q}_{k+1} = \boldsymbol{q}_k + \dot{\boldsymbol{q}}_{k+1}dt$$

And?

$$\phi(\boldsymbol{q}_{k+1}) \geqslant 0 \quad \textit{"signed distance function"}$$
$$\boldsymbol{\lambda} \geqslant 0 \quad \textit{"force pushes!"}$$
$$\phi(\boldsymbol{q}_{k+1})\boldsymbol{\lambda}_k = \boldsymbol{0} \quad \textit{"force only at contact"}$$

Ball with Unilateral Contact as a QP

State and dynamics. Let $\boldsymbol{q} = [x \;\; z]^\top$, $v = \dot{q}$, and $\boldsymbol{M} = \begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix}$. Free-flight acceleration is

$$\dot{\boldsymbol{q}} = \boldsymbol{v}, \qquad \dot{\boldsymbol{v}} = \boldsymbol{g} = \begin{bmatrix} 0 \\ -g \end{bmatrix}.$$

Backward Euler free-flight prediction. Given (q_k, v_k) and step dt,

$$\boldsymbol{v}_{k+1}^{\text{free}} = \boldsymbol{v}_k + dt\, \boldsymbol{g}, \qquad \phi_k = z_k, \qquad \phi_{k+1}^{\text{free}} = \phi_k + dt\, v_n^{\text{free}},$$

where the contact Jacobian is $\boldsymbol{J} = [0 \;\; 1]$ and $v_n = \boldsymbol{J}\boldsymbol{v}$ is the normal (vertical) velocity.

Impulse-based contact update. When contact is active, we solve for $(\boldsymbol{v}_{k+1}, \gamma)$:

$$\boldsymbol{M}\boldsymbol{v}_{k+1} - \boldsymbol{J}^\top \gamma = \boldsymbol{M}\boldsymbol{v}_{k+1}^{\text{free}},$$

so $\gamma \geqslant 0$ is an upward normal impulse.

Ball with Unilateral Contact as a QP (2)

Impulse-based contact update. When contact is active, we solve for $(\boldsymbol{v}_{k+1}, \gamma)$:

$$\boldsymbol{M}\boldsymbol{v}_{k+1} - \boldsymbol{J}^{\top}\gamma = \boldsymbol{M}\boldsymbol{v}_{k+1}^{\text{free}},$$

so $\gamma \geqslant 0$ is an upward normal impulse.

QP formulation. We pick the post-step velocity closest to free-flight:

$$\min_{\boldsymbol{v}_{k+1},\,\gamma} \quad \frac{1}{2}(\boldsymbol{v}_{k+1} - \boldsymbol{v}_{k+1}^{\text{free}})^{\top}\boldsymbol{M}(\boldsymbol{v}_{k+1} - \boldsymbol{v}_{k+1}^{\text{free}})$$

subject to unilateral contact constraints:

$$\underbrace{\phi_k + dt\,\boldsymbol{J}\boldsymbol{v}_{k+1} \geqslant 0}_{\text{nonpenetration}}, \qquad \underbrace{\gamma \geqslant 0}_{\text{unilateral impulse}} \quad .$$

Restitution (impact only). If $v_{n,k} < 0$ and $\phi_{k+1}^{\text{free}} < 0$, enforce

$$\boldsymbol{J}\boldsymbol{v}_{k+1} = -e\,v_{n,k},$$

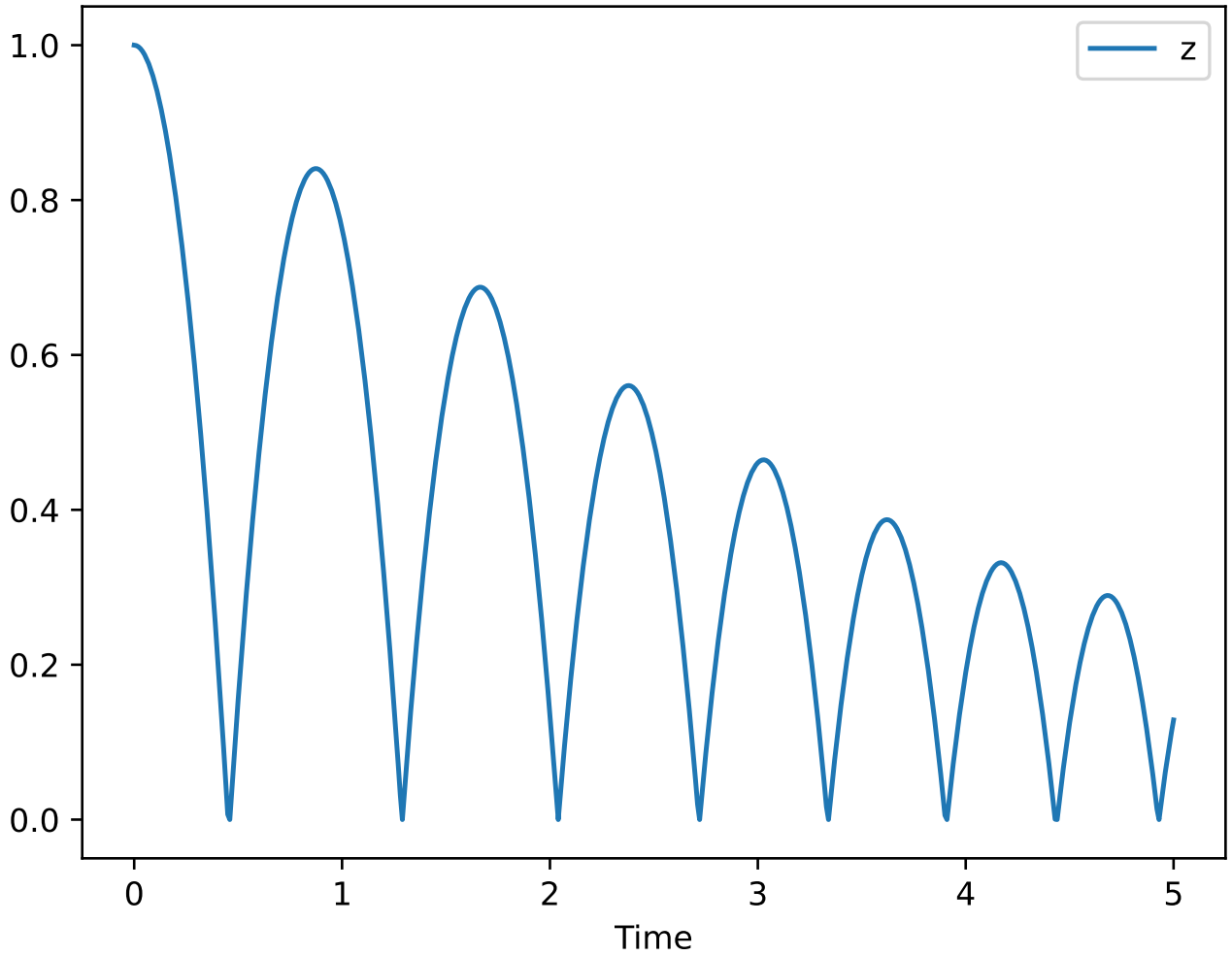
giving a bounce with coefficient e.

State update.

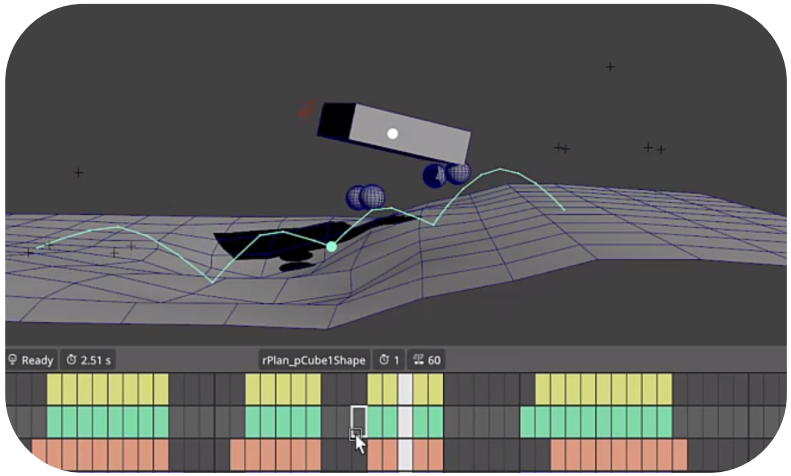
$$\boldsymbol{q}_{k+1} = \boldsymbol{q}_k + dt\,\boldsymbol{v}_{k+1}.$$

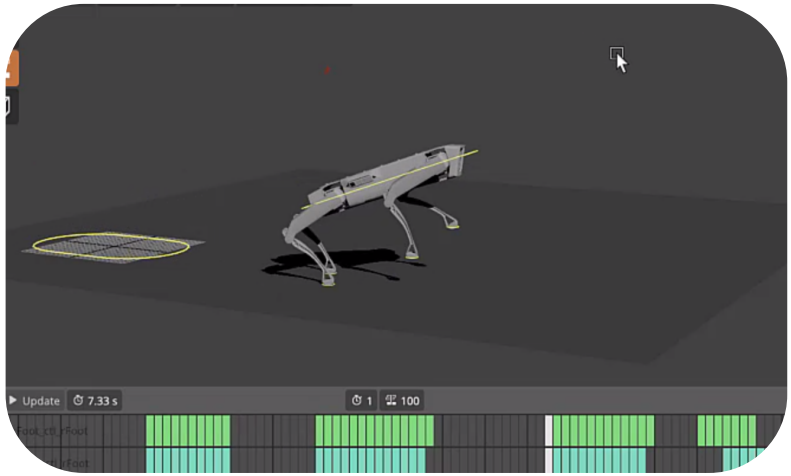
Optionally clamp to rest if $\phi_{k+1} \approx 0$ and $|v_{n,k+1}| \approx 0$.

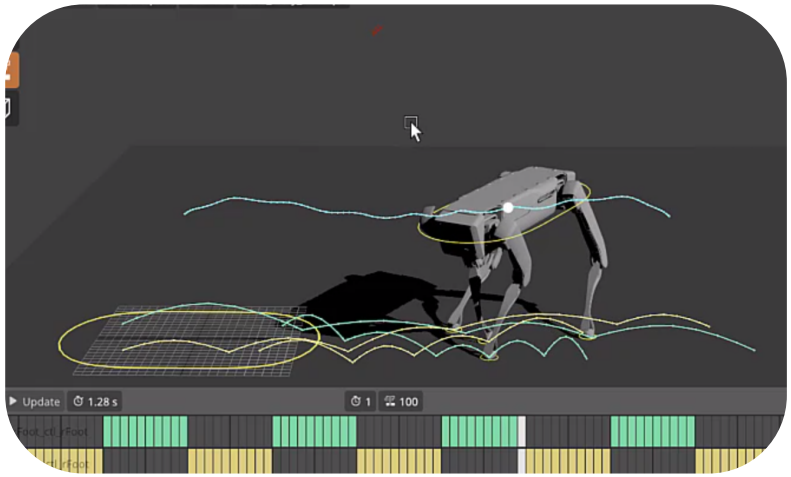
Falling Ball - Code Example

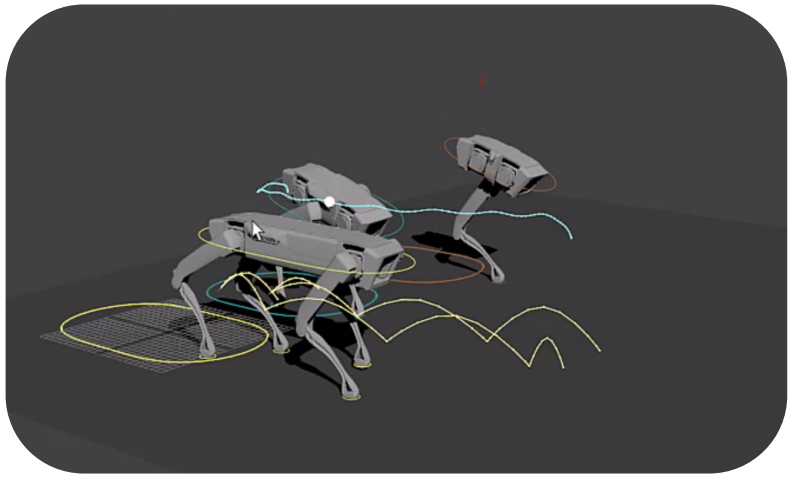


So how about this?



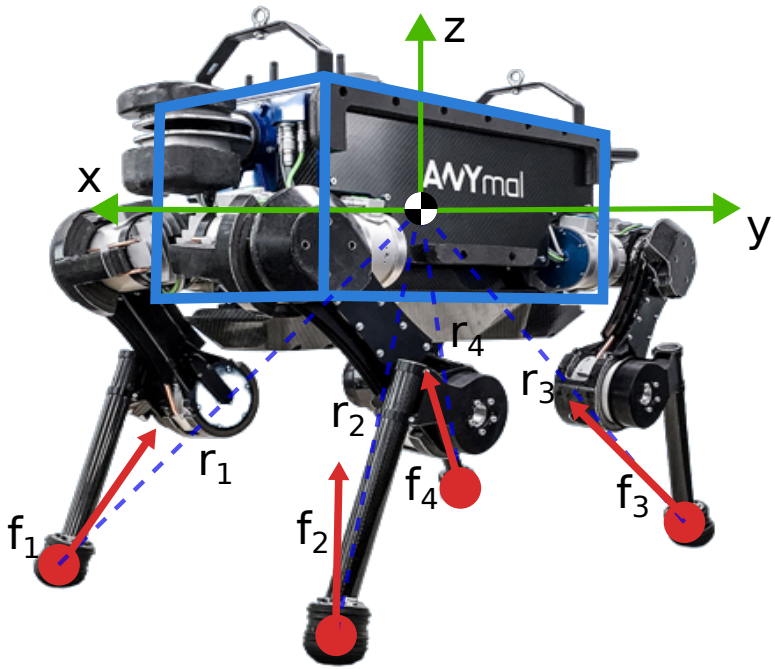




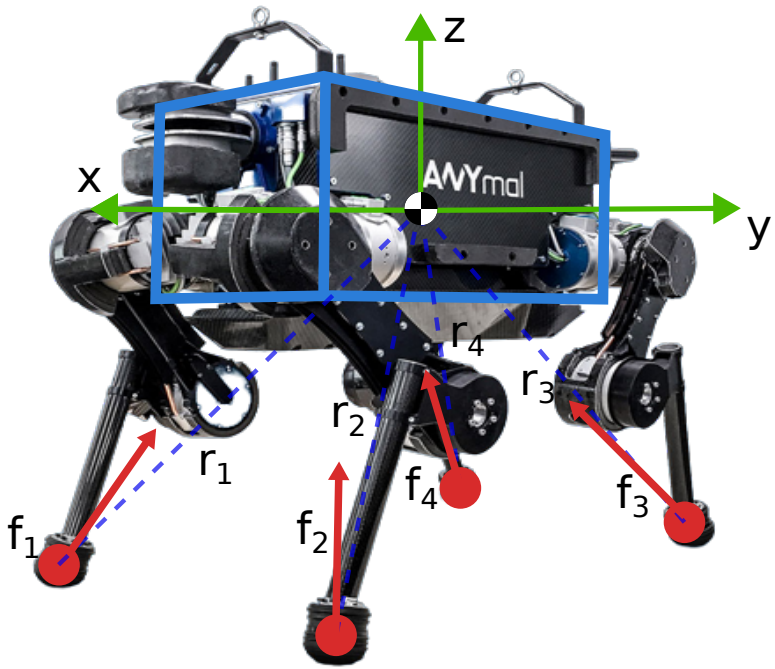


Credits: Ragdoll Dynamics (Imbalance Ltd)

Single Rigid Body Dynamics (SRBD) Model



Single Rigid Body Dynamics (SRBD) Model



Key Ideas of the SRBD model:

- We assume that the feet are massless, but can push on the ground!
- We only care about the movement of the root body!

SRBD Details

In the SRBD model:

- The rigid body has a mass $m \in \mathbb{R}^+$ and moment of inertia $\boldsymbol{I} \in \mathbb{R}^{3\times 3}$

$$\boldsymbol{x}_{\text{body}} = \begin{bmatrix} \boldsymbol{p}_w \in \mathbb{R}^3 \\ \dot{\boldsymbol{p}}_w \in \mathbb{R}^3 \\ \boldsymbol{R}_w \in \mathcal{SO}(3) \\ \boldsymbol{\omega}_b \in \mathbb{R}^3 \end{bmatrix}$$

- Each leg i has a position $\boldsymbol{p}_i \in \mathbb{R}^3$, it can generate contact forces $\boldsymbol{f}_i \in \mathbb{R}^3$

$$\ddot{\boldsymbol{p}}_w = \frac{\sum_i \boldsymbol{f}_i + m\,\boldsymbol{g}}{m}$$

$$\dot{\boldsymbol{\omega}}_b = \boldsymbol{I}^{-1}(\boldsymbol{R}_w^T(\sum_i(\boldsymbol{p}_i - \boldsymbol{p}_w) \times \boldsymbol{f}_i) - \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega})$$

SRBD Details (2)

Let's write it cleaner:

$$\ddot{\boldsymbol{p}}_w = \frac{\boldsymbol{f}_{\text{total}}}{m}$$
$$\dot{\boldsymbol{\omega}}_b = \boldsymbol{I}^{-1}(\boldsymbol{R}_w^T \boldsymbol{\tau}_{\text{total}} - \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega})$$

where

$$\boldsymbol{f}_{\text{total}} = \sum_i \boldsymbol{f}_i + m\,\boldsymbol{g}$$
$$\boldsymbol{\tau}_{\text{total}} = \sum_i \boldsymbol{r}_i \times \boldsymbol{f}_i$$
$$\boldsymbol{r}_i = \boldsymbol{p}_i - \boldsymbol{p}_w$$

Now we can integrate via RK4, Semi-Implicit Euler or any integrator that we like!

SRBD Details (3)

Remember: We need to have constraints for the forces!

$$(\boldsymbol{f} \cdot \boldsymbol{n}) > 0, \text{ the force cannot pull/grasp!}$$

$$\|\boldsymbol{f} \cdot \boldsymbol{t}\|_2 \leqslant \mu(\boldsymbol{f} \cdot \boldsymbol{n})$$

$$\|\boldsymbol{f} \cdot \boldsymbol{b}\|_2 \leqslant \mu(\boldsymbol{f} \cdot \boldsymbol{n})$$

These are "Conic" constraints!

SRBD Details (3)

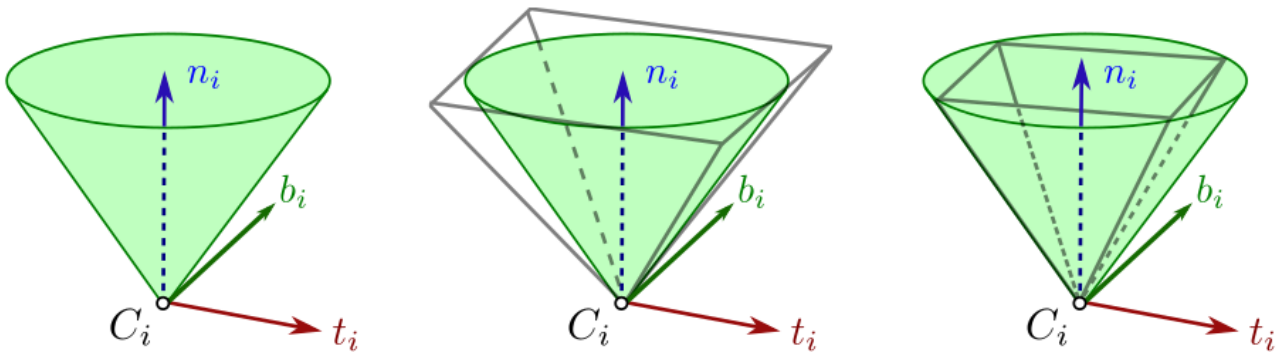
Remember: We need to have constraints for the forces!

$$(\boldsymbol{f} \cdot \boldsymbol{n}) > 0, \text{ the force cannot pull/grasp!}$$

$$\|\boldsymbol{f} \cdot \boldsymbol{t}\|_2 \leqslant \mu(\boldsymbol{f} \cdot \boldsymbol{n})$$

$$\|\boldsymbol{f} \cdot \boldsymbol{b}\|_2 \leqslant \mu(\boldsymbol{f} \cdot \boldsymbol{n})$$

These are "Conic" constraints! We usually linearize them to help the solvers:



From `https://scaron.info/`

$$(\boldsymbol{f} \cdot \boldsymbol{n}) > 0, \text{ the force cannot pull/grasp!}$$

$$|\boldsymbol{f} \cdot \boldsymbol{t}| \leqslant \mu(\boldsymbol{f} \cdot \boldsymbol{n}) \text{ or } |\boldsymbol{f} \cdot \boldsymbol{t}| \leqslant \frac{\mu}{\sqrt{2}}(\boldsymbol{f} \cdot \boldsymbol{n})$$

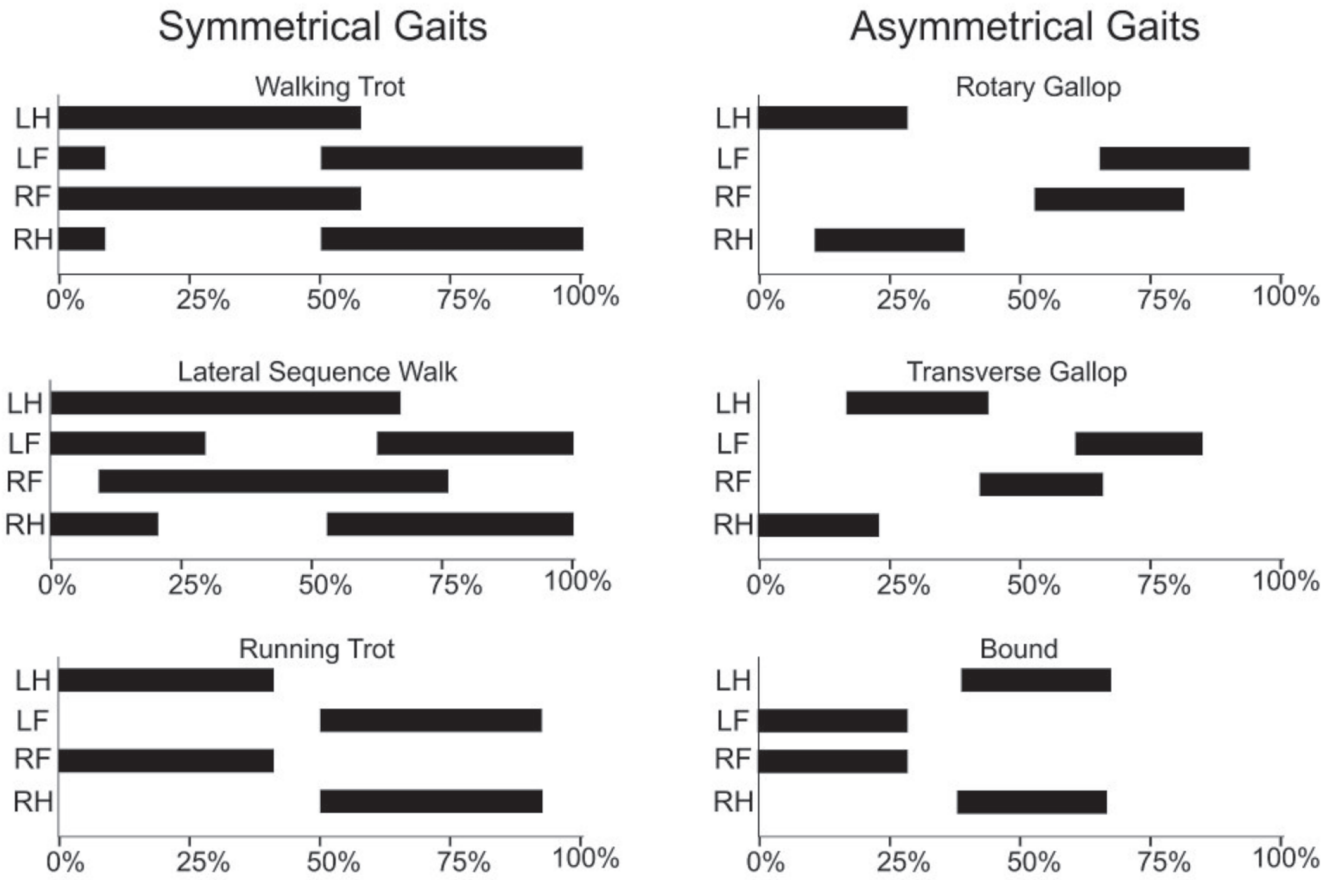
$$|\boldsymbol{f} \cdot \boldsymbol{b}| \leqslant \mu(\boldsymbol{f} \cdot \boldsymbol{n}) \text{ or } |\boldsymbol{f} \cdot \boldsymbol{b}| \leqslant \frac{\mu}{\sqrt{2}}(\boldsymbol{f} \cdot \boldsymbol{n})$$

Gait Sequence/Scheduling

But the feet are not moving!

Gait Sequence/Scheduling

But the feet are not moving!



Trajectory Optimization with the SRBD model

- We have K knot points
- For each knot point k and for each foot i, we **select beforehand** if the foot is in contact or not!
- We have the following optimization variables:

$$\boldsymbol{\theta}_k = \begin{bmatrix} \boldsymbol{p}_k \\ \dot{\boldsymbol{p}}_k \\ \boldsymbol{R}_k \\ \boldsymbol{\omega}_k \\ \boldsymbol{p}_{ik} \\ \boldsymbol{f}_{ik} \end{bmatrix}$$

- We interpolate each foot position $\boldsymbol{p}_i$ when in air! Or we can constrain their movement with respect to the body!
- How can we optimize/represent for each $\boldsymbol{R}_k$?

Trajectory Optimization with the SRBD model

- We have K knot points
- For each knot point k and for each foot i, we **select beforehand** if the foot is in contact or not!
- We have the following optimization variables:

$$\boldsymbol{\theta}_k = \begin{bmatrix} \boldsymbol{p}_k \\ \dot{\boldsymbol{p}}_k \\ \boldsymbol{R}_k \\ \boldsymbol{\omega}_k \\ \boldsymbol{p}_{ik} \\ \boldsymbol{f}_{ik} \end{bmatrix}$$

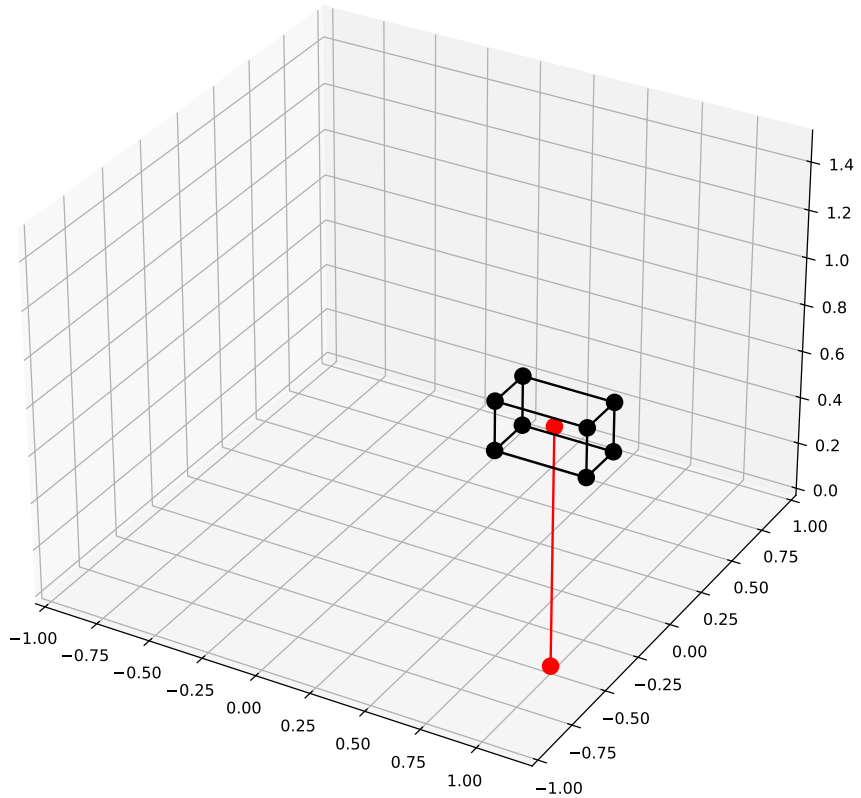
- We interpolate each foot position $\boldsymbol{p}_i$ when in air! Or we can constrain their movement with respect to the body!
- How can we optimize/represent for each $\boldsymbol{R}_k$? **Next lecture!**

Monopod Example without Orientation

Let's start simple:

- One foot (Monopod!)
- No orientation
- Euler Integration (implicit)
- Total time: $2.5\,s$
- Gait Sequence:
  $\texttt{Contact}(0.5\,s)$, $\texttt{Air}(0.5\,s)$, $\texttt{Contact}(0.5\,s)$, $\texttt{Air}(0.5\,s)$, $\texttt{Contact}(0.5\,s)$
- We can choose K as we like!

Monopod - Code Example



Thank you

- **Any Questions?**
- **Office Hours**:
  - **Tue & Thu** *(09:00-11:00)*
  - 24/7 by email (`costashatz@upatras.gr`, subject: *ECE_RSII_AM*)
- **Material and Announcements**

Laboratory of Automation & Robotics