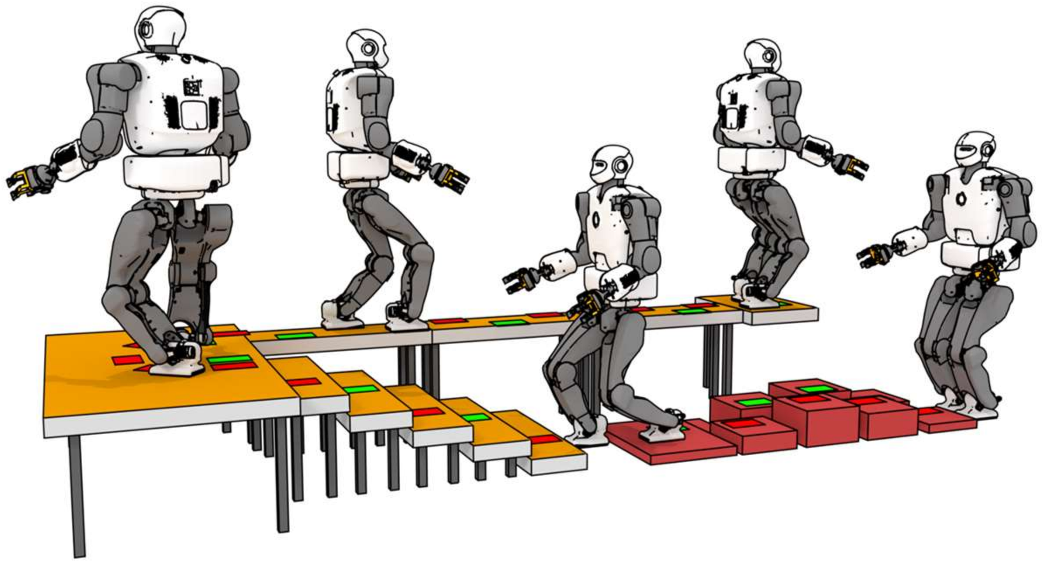
The "contact planning problem" for legged robots: A cardinality minimisation approach



Steve Tonneau

THE UNIVERSITY of EDINBURGH

R4
13/09/2021

Collaborators


Daeun Song   Young J. Kim

이화여자대학교
EWHA WOMANS UNIVERSITY

Andrea Del Prete

UNIVERSITY OF TRENTO - Italy


Nicolas Mansard   Pierre Fernbach

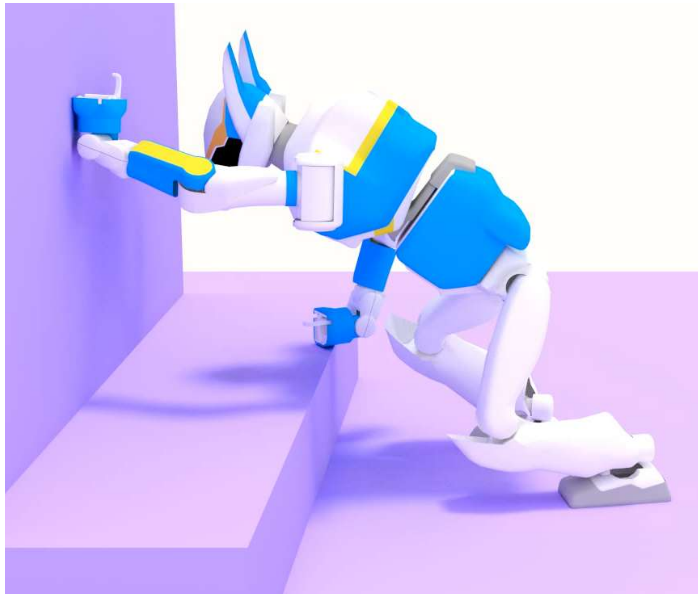
Gepetto
LAAS-CNRS

Solving Footstep Planning as a Feasibility Problem using L1-norm Minimization, *RAL 2021*

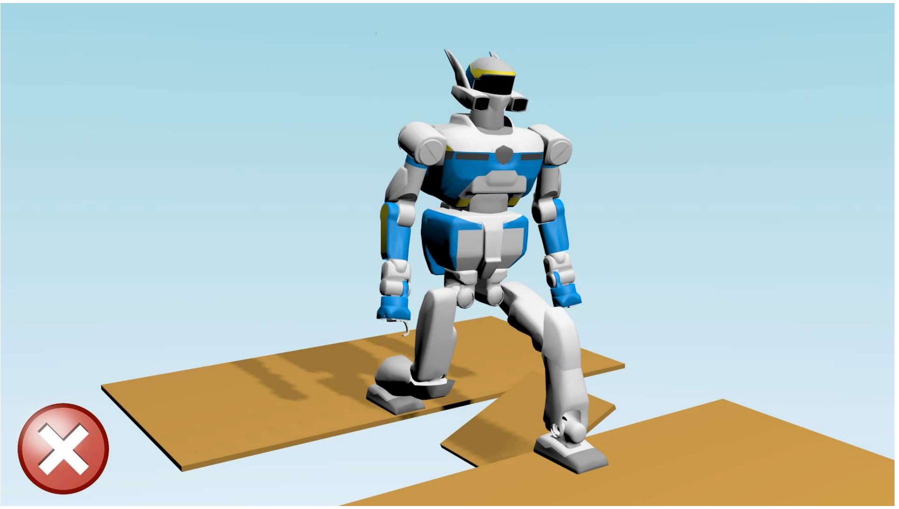
Research goal: autonomous legged locomotion

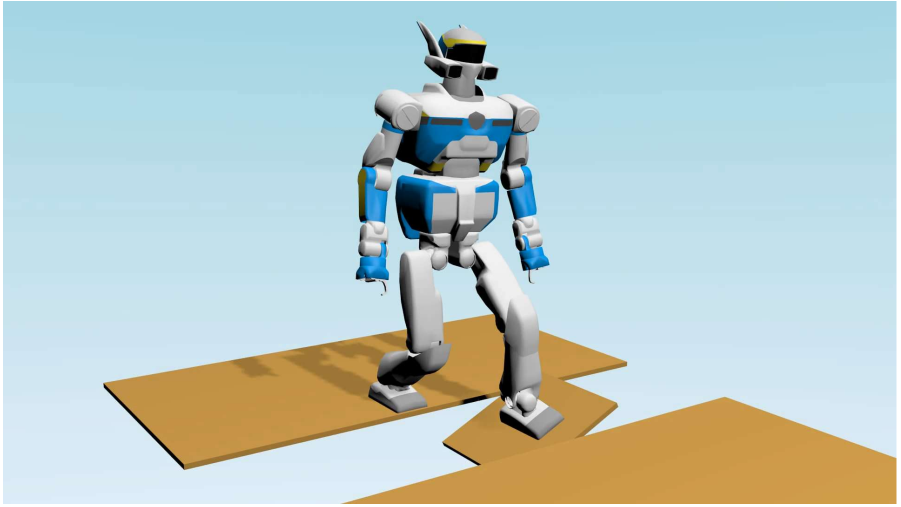
Carpentier, Tonneau, Naveau, Stasse et Mansard, ICRA 16

Contact postures in high-dimensional space ?

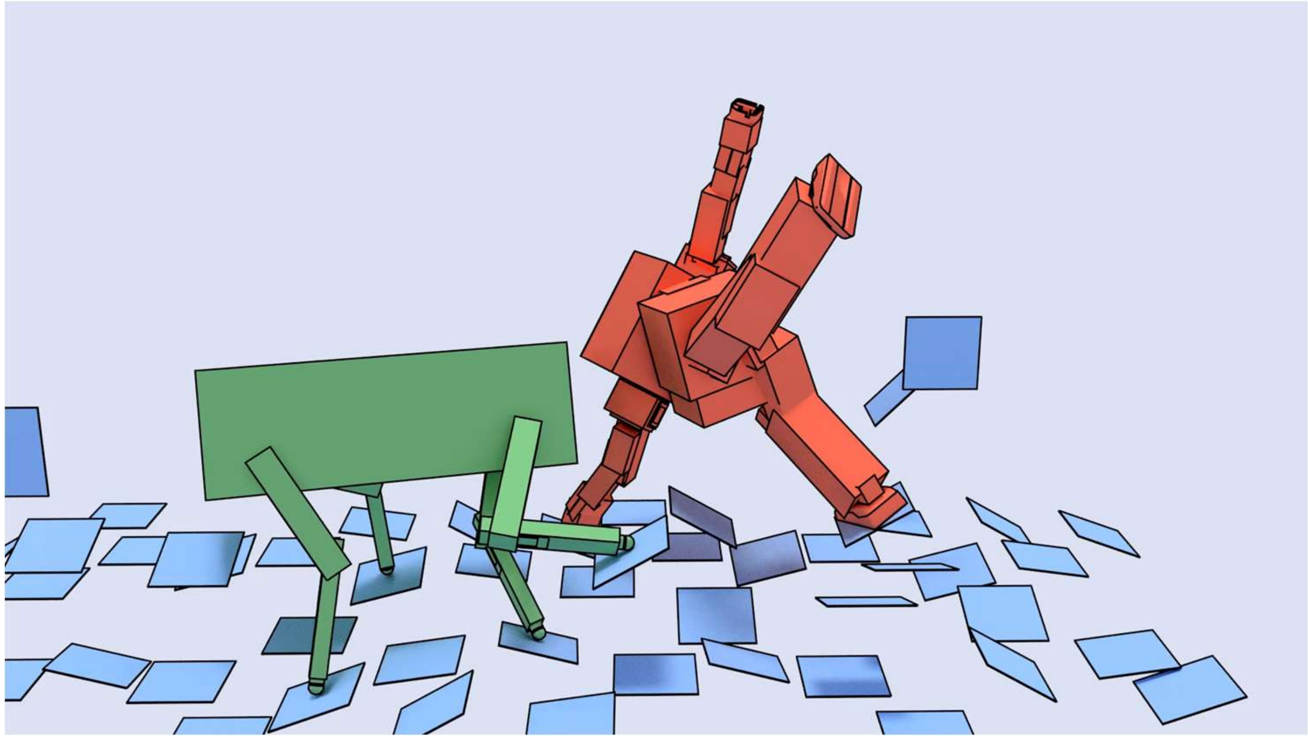


Contact-dependent, discontinuous, non-linear dynamics / geometric constraints





Contact interactions without collisions ?



Legged locomotion is too hard

We need to cheat !

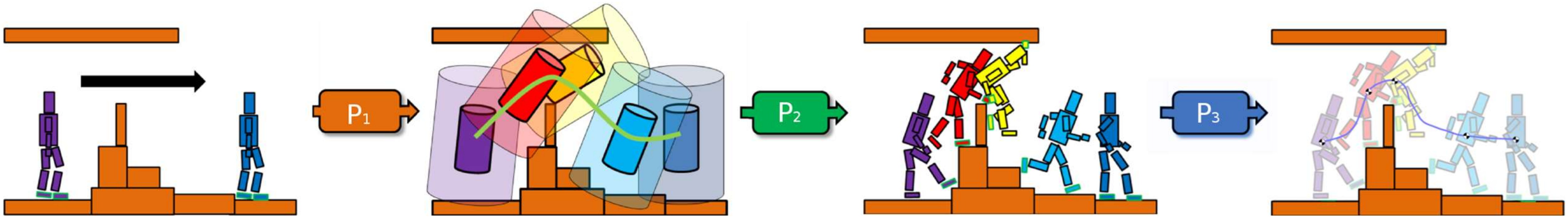
This has a cost … not discussed today

A divide and conquer approach [Tonneau et al. 15]

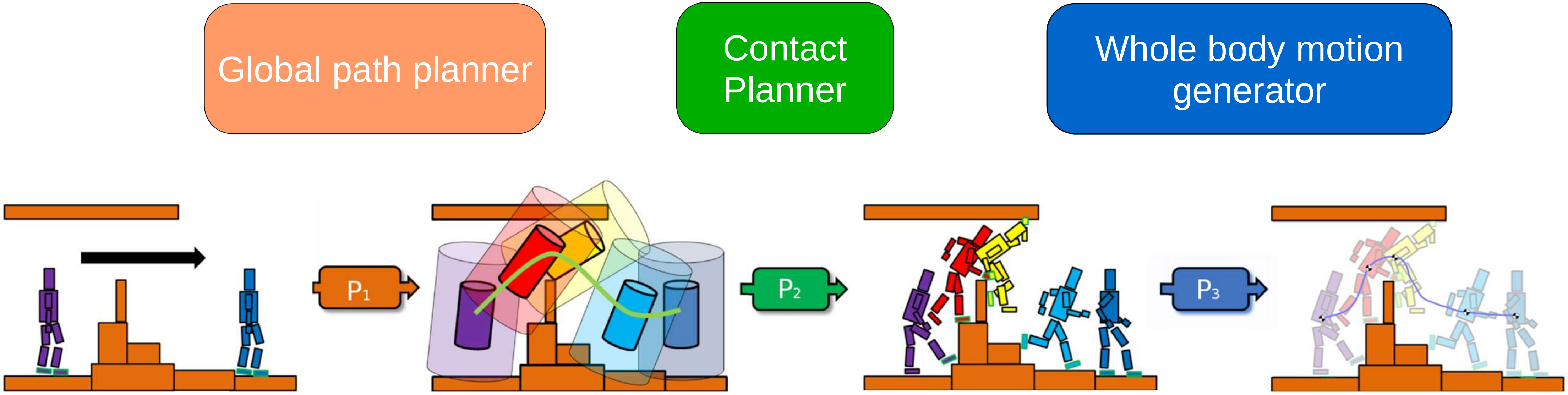
Global path planner

Contact Planner

Whole body motion generator

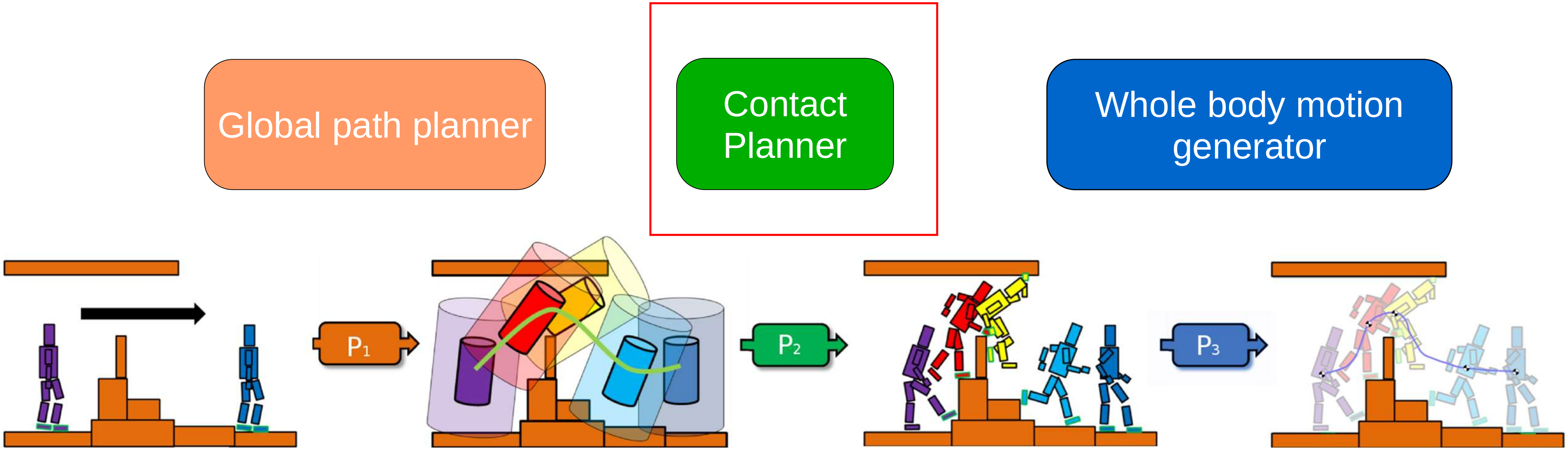


A divide and conquer approach [Tonneau et al. 15]



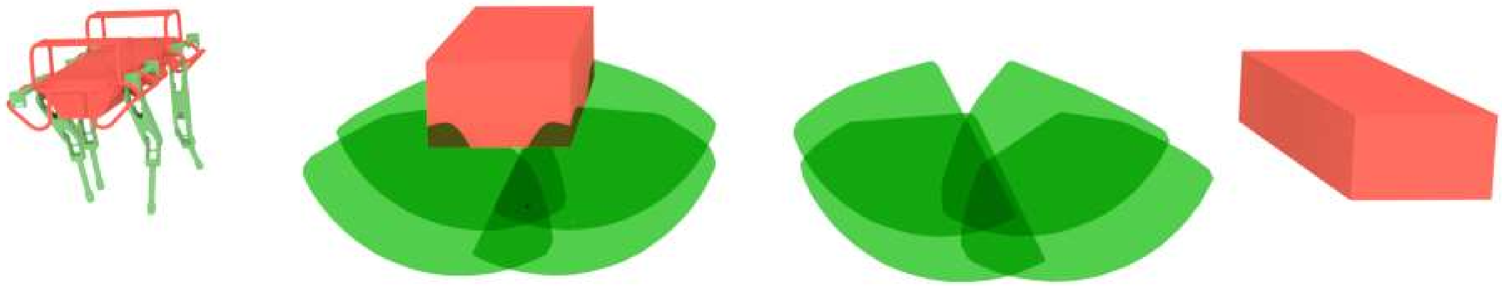
Solving $\mathcal{P}_i$ in the feasibility domain of $\mathcal{P}_j, j > i$?

A divide and conquer approach [Tonneau et al. 15]



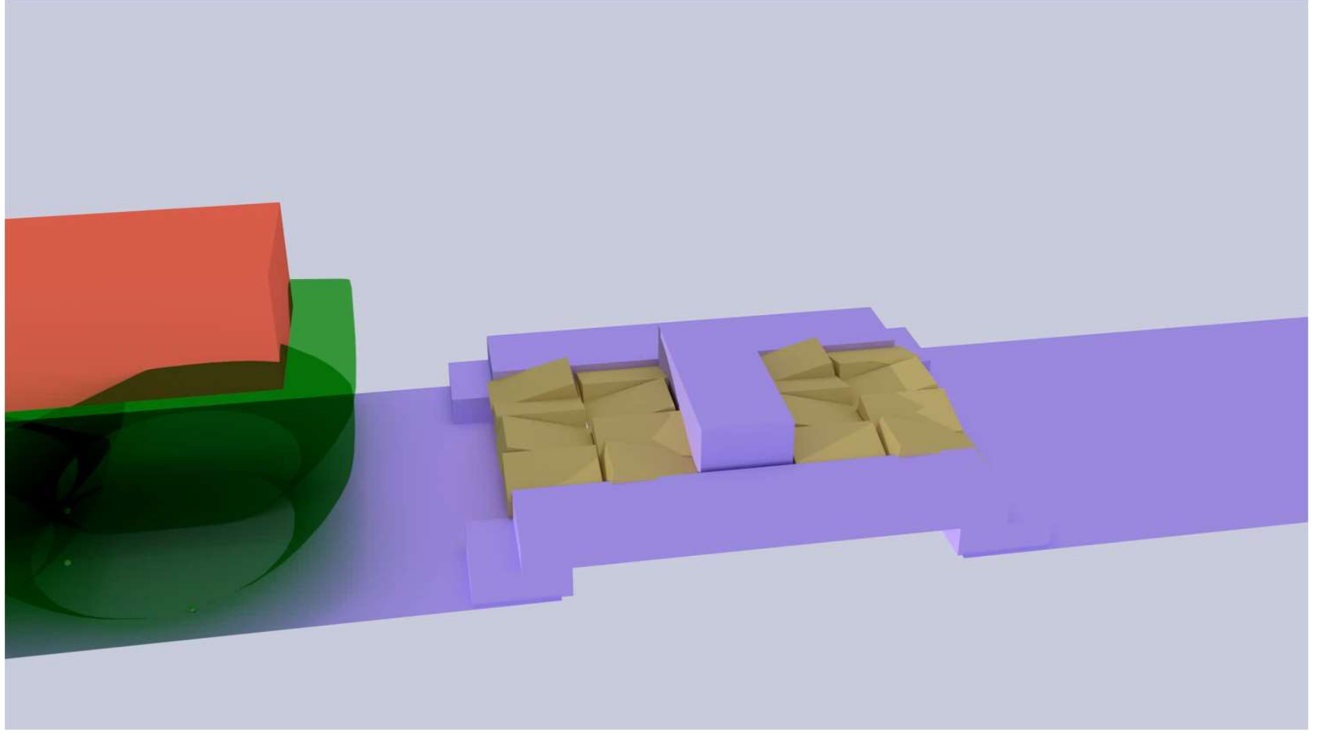
Solving $\mathcal{P}_i$ in the feasibility domain of $\mathcal{P}_j, j > i$?

Cheating assumption: low-dimensional space



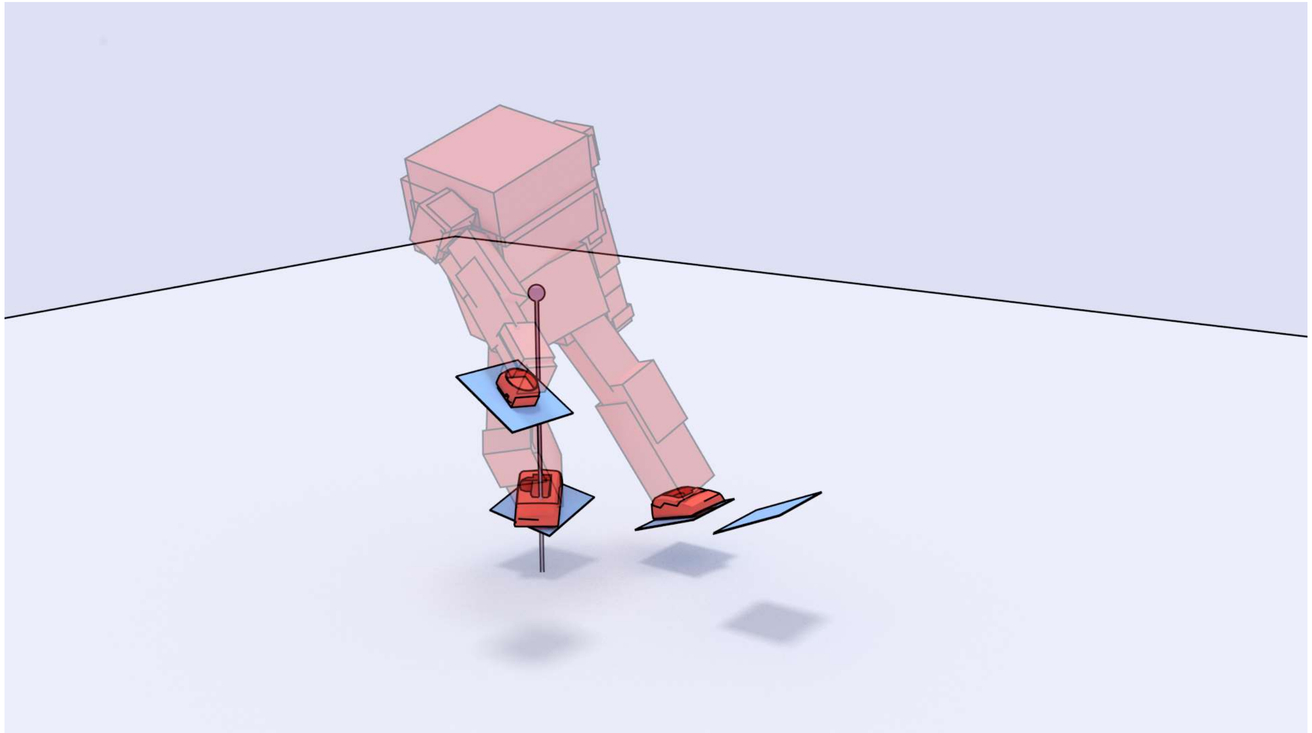
Dimensionality reduction for contacts / collision constraints [Tonneau el al. 2014]

Cheating assumption: low-dimensional space



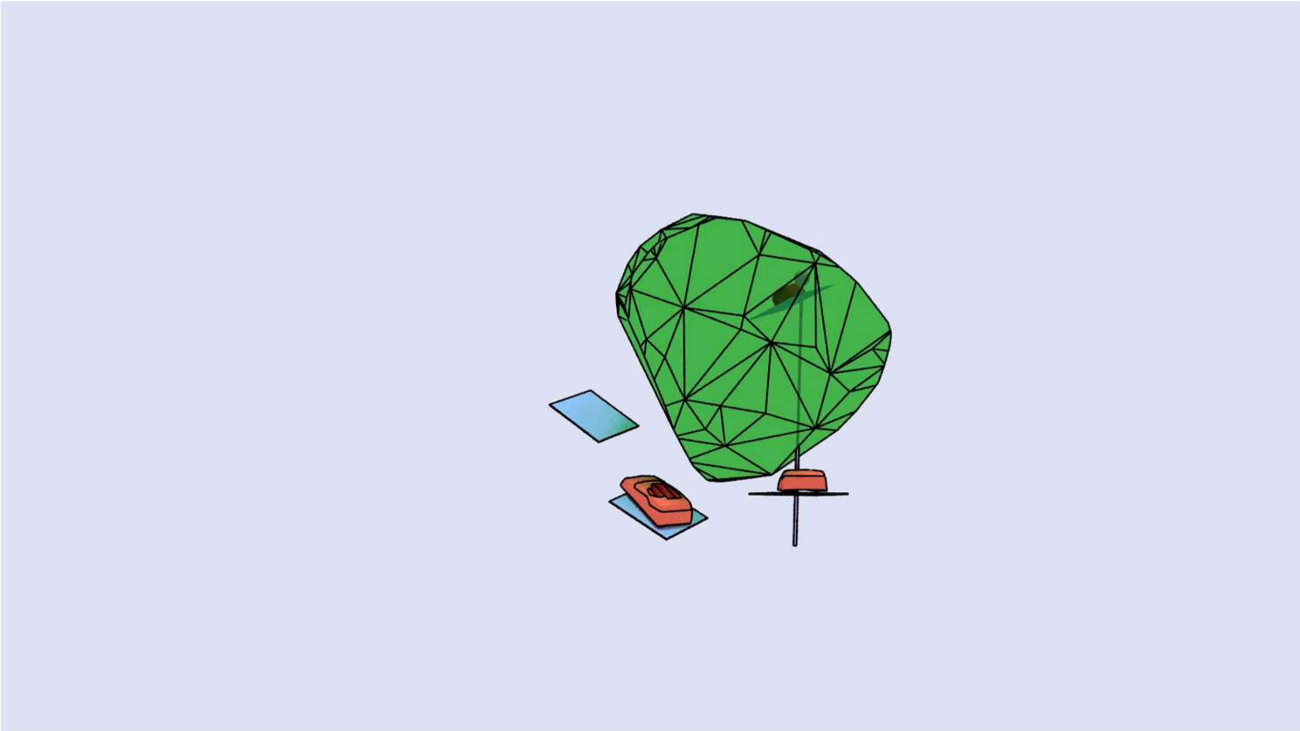
Low dimensional root path planner (RBPRM) [Tonneau et al. 15]

Cheating assumption: low-dimensional space



Dimensionality reduction (Centroidal model) [kajita 03, Orin 09]

Cheating assumption: linear kinematic constraints



Linear geometric constraints [Tonneau et al. 18]

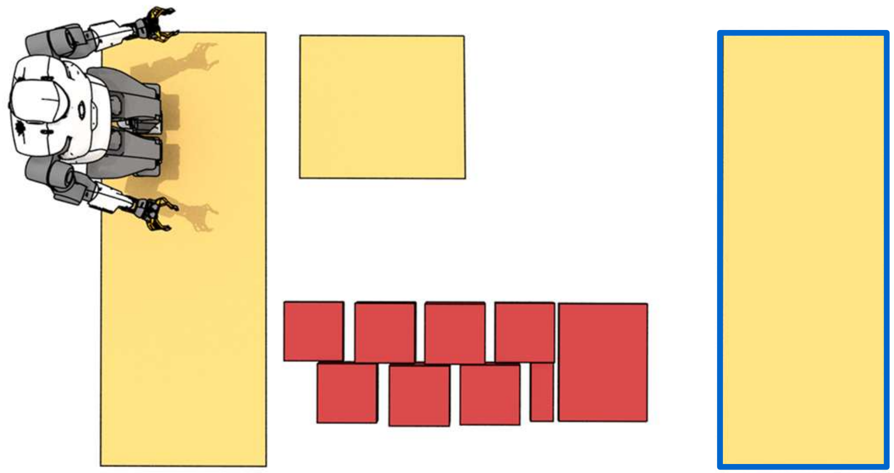
~~Cheat~~.. conservative assumption: linear dynamics

CROC [Fernbach, Tonneau el al. 2020]

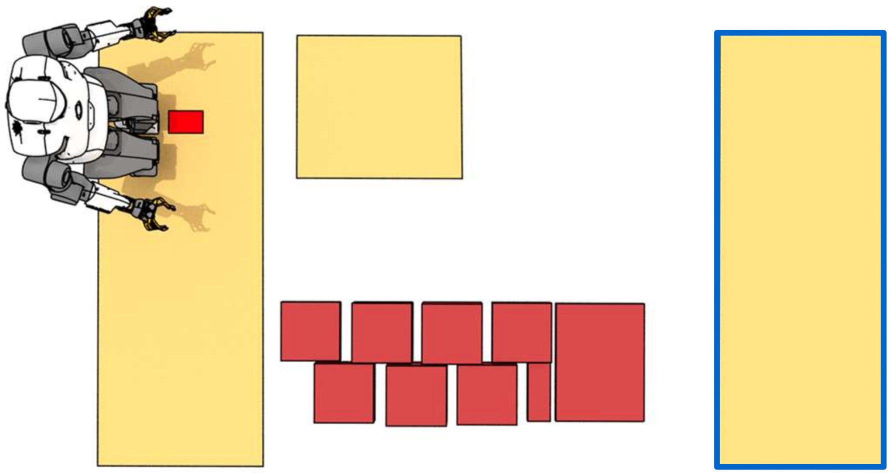
What’s left ?

A combinatorial problem of large polynomial complexity…

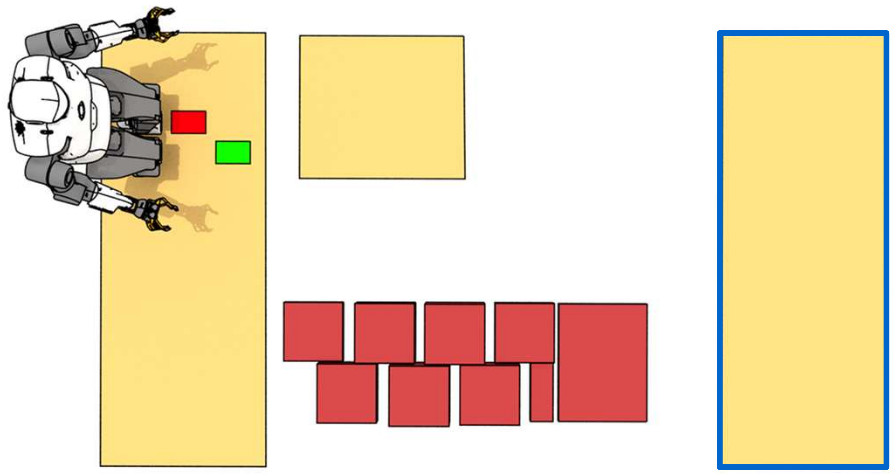
Example: how to reach the platform ?



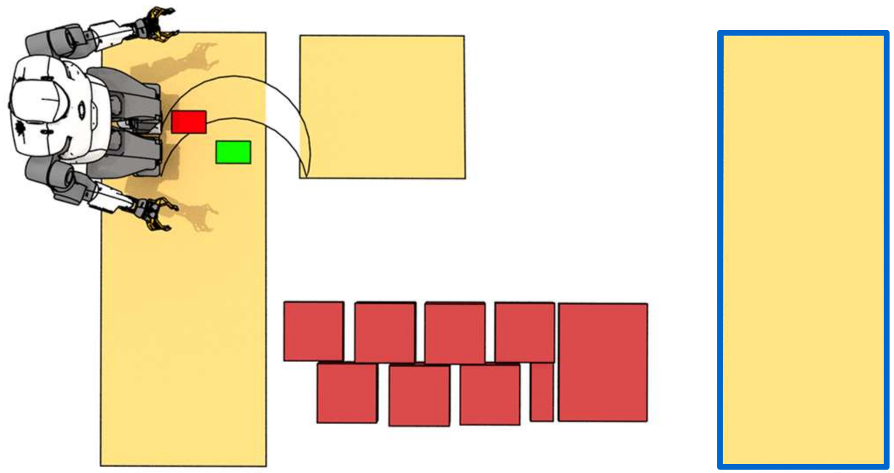
Combinatorial step planning



Combinatorial step planning



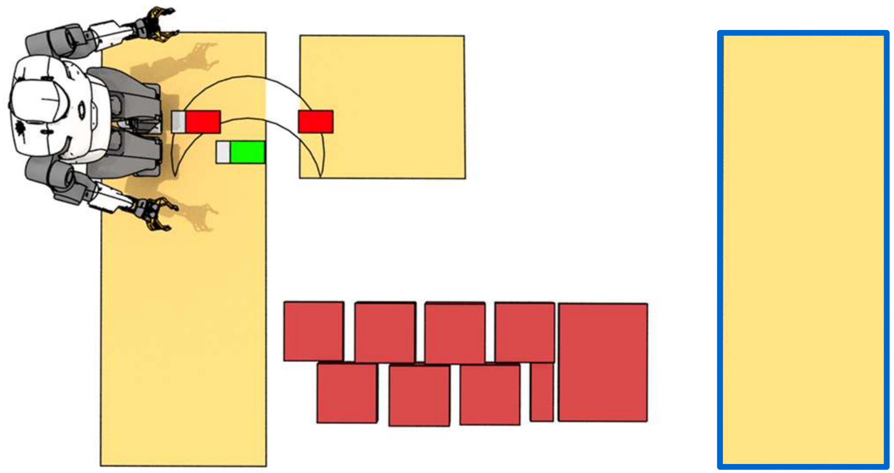
Combinatorial step planning



Feasibility $\mathcal{F}$:

Geometric constraints

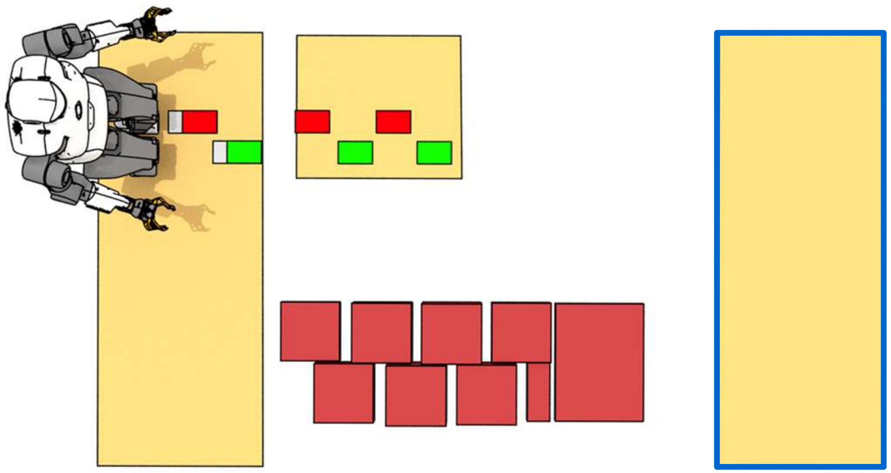
Combinatorial step planning



Feasibility $\mathcal{F}$:

Geometric constraints

Combinatorial step planning

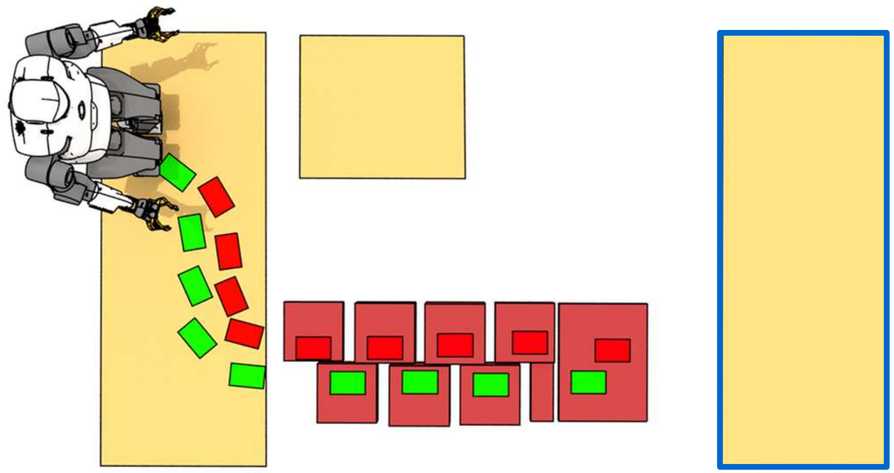


Global path search

Feasibility $\mathcal{F}$:

Geometric constraints

Combinatorial step planning

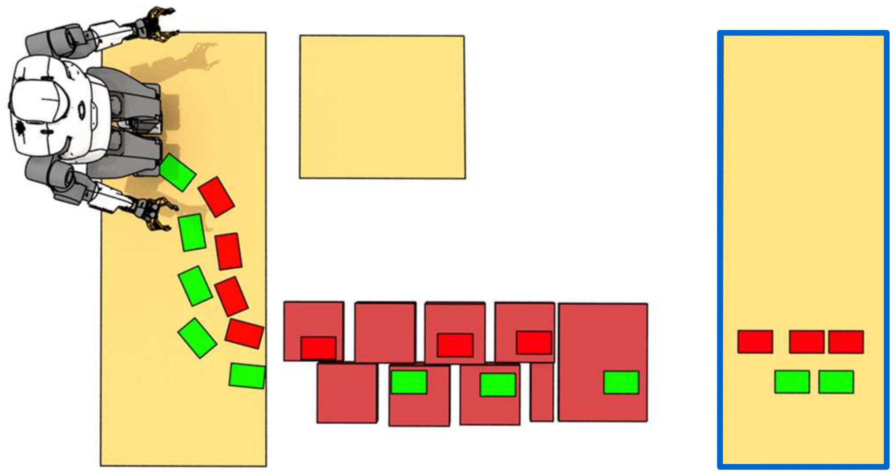


Global path search

Feasibility $\mathcal{F}$:

Geometric constraints

Combinatorial step planning



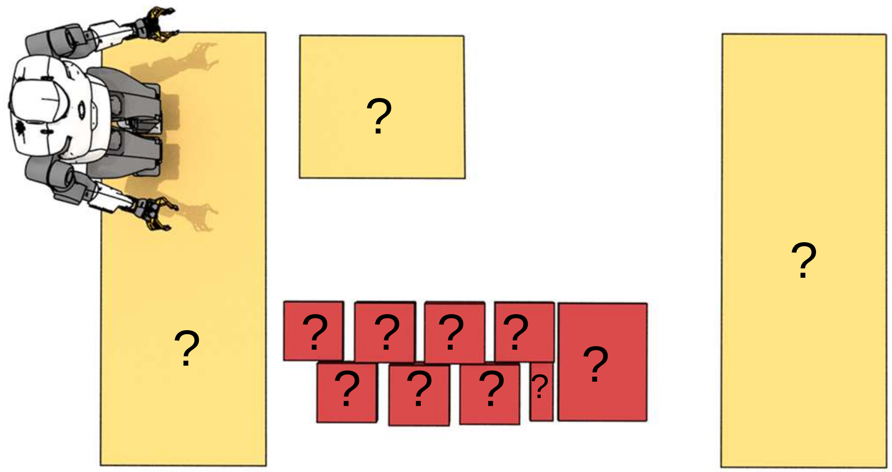
Global path search

Feasibility $\mathcal{F}$:

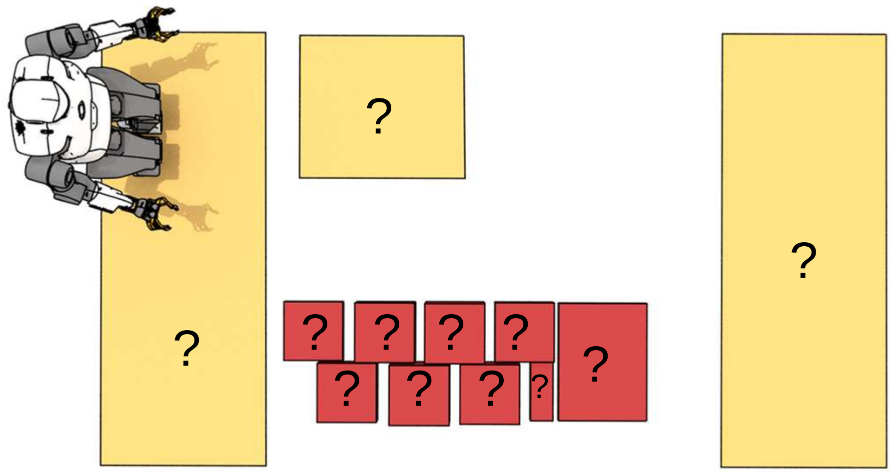
Geometric constraints

<u>Dynamic constraints</u>

Which contact surface ?



Which contact surface ?



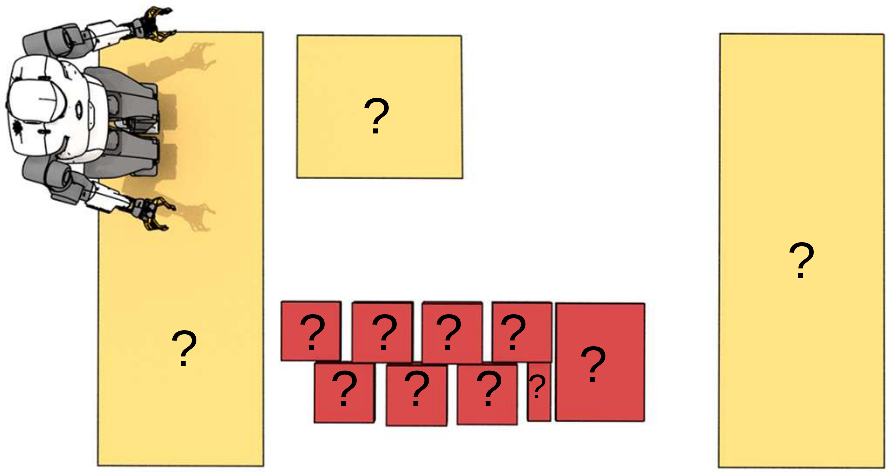
Convex contact surfaces:

$$\mathcal{S}_j : \{\mathbf{p}, \mathbf{S}_j \mathbf{p} \leq \mathbf{s}_j\}$$
$$1 \leq j \leq n$$

Footsteps positions:

$$\mathbf{p}_i, 1 \leq i \leq m$$

Which contact surface ?



Convex contact surfaces:

$$\mathcal{S}_j : \{\mathbf{p}, \mathbf{S}_j\mathbf{p} \leq \mathbf{s}_j\}$$
$$1 \leq j \leq n$$

Footsteps positions:

$$\mathbf{p}_i, 1 \leq i \leq m$$

Combinatorial:

$$m^n$$

Contact planning as a feasibility problem

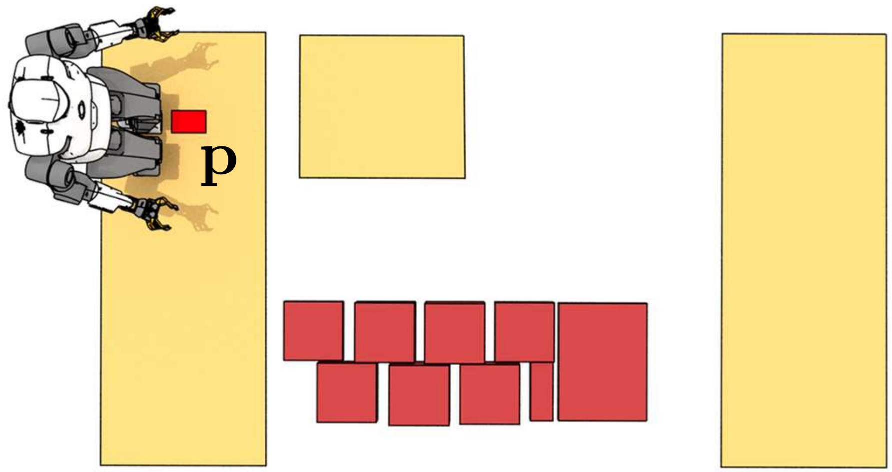
$$
\begin{aligned}
\text{find} \quad & \mathbf{X} = [\mathbf{p}_1 \dots \mathbf{p}_m] \\
\text{s.t.} \quad & \mathbf{X} \in \mathcal{F} && \text{// Feasibility} \\
& \mathbf{X} \in \mathcal{I} \cap \mathcal{G} && \text{// Initial and goal conditions} \\
& \forall i : \\
& \qquad \mathbf{p}_i \in \mathcal{S}_1 \vee \cdots \vee \mathbf{p}_i \in \mathcal{S}_n
\end{aligned}
$$

Contact planning as a feasibility problem

$$
\begin{aligned}
\text{find} \quad & \mathbf{X} = [\mathbf{p}_1 \ldots \mathbf{p}_m] \\
\text{s.t.} \quad & \mathbf{X} \in \mathcal{F} && \text{// Feasibility} \\
& \mathbf{X} \in \mathcal{I} \cap \mathcal{G} && \text{// Initial and goal conditions} \\
& \forall i : \\
& \qquad \mathbf{p}_i \in \mathcal{S}_1 \vee \cdots \vee \mathbf{p}_i \in \mathcal{S}_n
\end{aligned}
$$

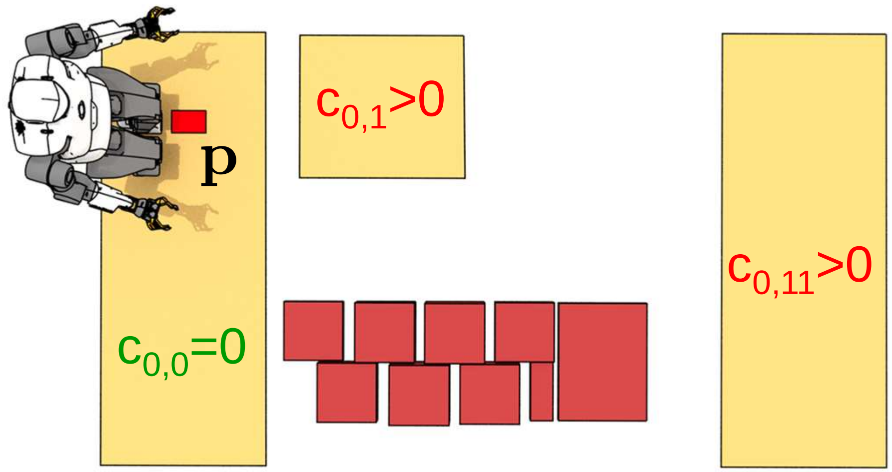
How to tackle the combinatorics ?

An equivalent cardinality minimisation problem



$$\mathbf{S}_j \mathbf{p}_i \leq \mathbf{s}_j \Leftrightarrow \mathbf{p}_i \in \mathcal{S}_j$$

An equivalent cardinality minimisation problem

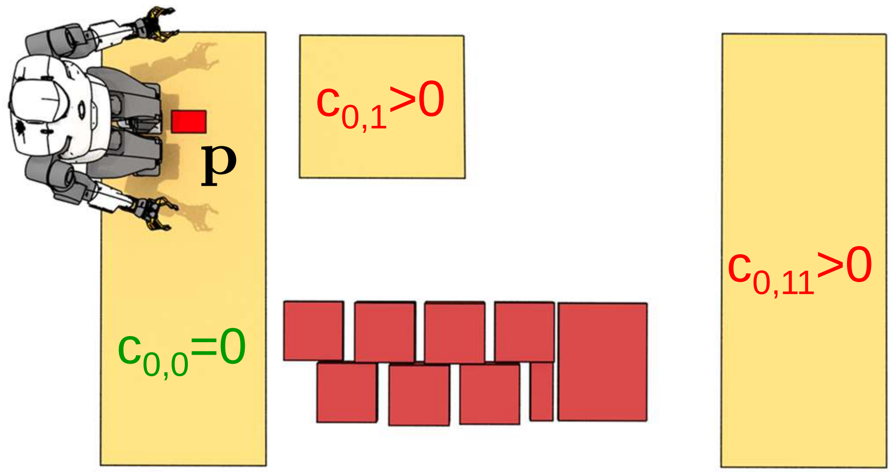


Slack variables $c_{i,j} \in \mathbb{R}^+$

$$\left.\begin{array}{l}\mathbf{S}_j \mathbf{p}_i - \mathbf{1} c_{i,j} \leq \mathbf{s}_j \\ c_{i,j} = 0\end{array}\right\} \Rightarrow \mathbf{p_i} \in \mathcal{S}_j$$

$$\mathbf{c}_i = [c_{i,1}, \ldots, c_{i,n}]$$

An equivalent cardinality minimisation problem



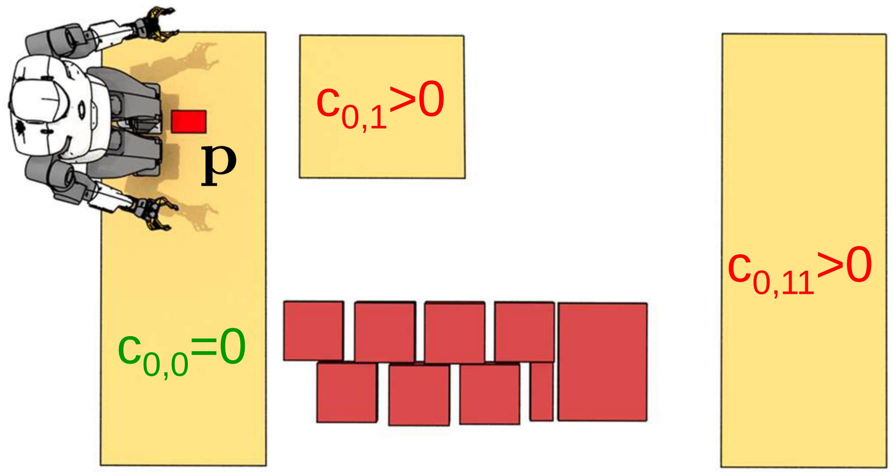
Slack variables $c_{i,j} \in \mathbb{R}^+$

$$\left.\begin{array}{l} \mathbf{S}_j \mathbf{p}_i - \mathbf{1} c_{i,j} \leq \mathbf{s}_j \\ c_{i,j} = 0 \end{array}\right\} \Rightarrow \mathbf{p_i} \in \mathcal{S}_j$$

$$\mathbf{c}_i = [c_{i,1}, \ldots, c_{i,n}]$$

$$\#NonZeros(\mathbf{c}_i) = n - 1 \Rightarrow \mathbf{p}_i \quad \text{on a surface}$$

An equivalent cardinality minimisation problem



$$
\begin{aligned}
\text{find} \quad & \mathbf{p}_i \\
\text{s.t.} \quad & \mathbf{p}_i \in \mathcal{S}_1 \vee \cdots \vee \mathbf{p}_i \in \mathcal{S}_n
\end{aligned}
$$

$$\Leftrightarrow$$

$$
\begin{aligned}
\text{find} \quad & \mathbf{p}_i, \mathbf{c}_i \\
\text{min} \quad & card(\mathbf{c}_i) \\
\text{s.t.} \quad & \mathbf{S}_j \mathbf{p}_i - \mathbf{1} c_{i,j} \leq \mathbf{s}_j \quad \forall j
\end{aligned}
$$

An equivalent cardinality minimisation problem

$$
\begin{aligned}
\text{find} \quad & \mathbf{X} = [\mathbf{p}_1 \dots \mathbf{p}_m] \\
\text{s.t.} \quad & \mathbf{X} \in \mathcal{F} \\
& \mathbf{X} \in \mathcal{I} \cap \mathcal{G} \\
& \forall i : \\
& \quad \mathbf{p}_i \in \mathcal{S}_1 \vee \dots \vee \mathbf{p}_i \in \mathcal{S}_n
\end{aligned}
$$

$$\Leftrightarrow$$

$$
\begin{aligned}
\text{find} \quad & \mathbf{X} = [\mathbf{p}_1 \dots \mathbf{p}_m] \\
& \mathbf{C} = [\mathbf{c}_1 \dots \mathbf{c}_n] \\
\text{min} \quad & \sum_{i=1}^{m} card(\mathbf{c}_i) \\
\text{s.t.} \quad & \mathbf{S}_j \mathbf{p}_i - \mathbf{1} c_{i,j} \leq \mathbf{s}_j \forall i, \forall j \\
& \mathbf{X} \in \mathcal{F} \\
& \mathbf{X} \in \mathcal{I} \cap \mathcal{G}
\end{aligned}
$$

Convex relaxation with sparsity inducing norm

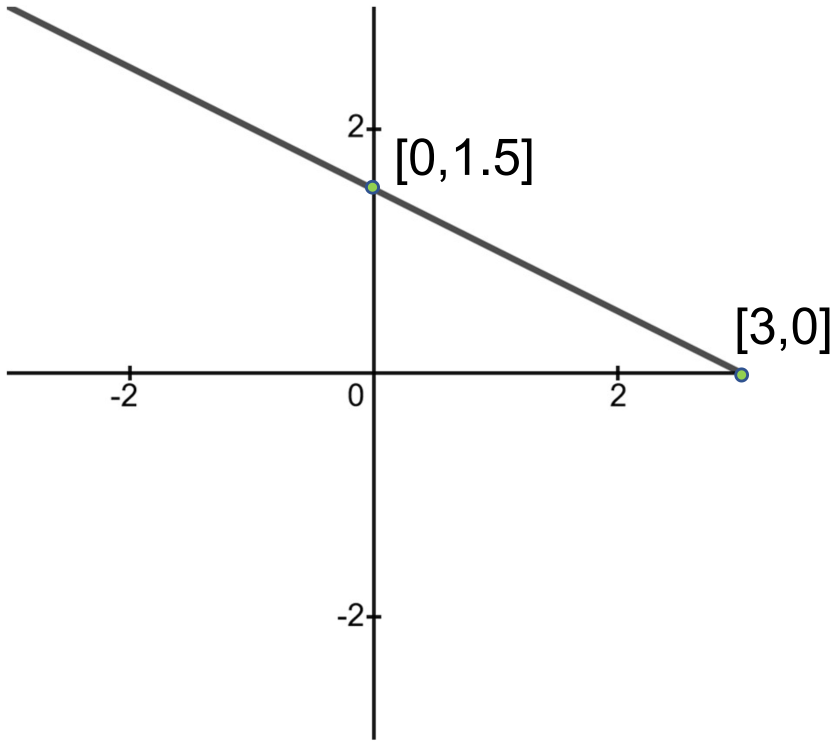
$$
\begin{aligned}
\text{find} \quad & \mathbf{X} = [\mathbf{p}_1 \dots \mathbf{p}_m] \\
& \mathbf{C} = [\mathbf{c}_1 \dots \mathbf{c}_n] \\
\min \quad & \sum_{i=1}^{m} card(\mathbf{c}_i) \\
\text{s.t.} \quad & \mathbf{S}_j \mathbf{p}_i - \mathbf{1} c_{i,j} \leq \mathbf{s}_j \forall i, \forall j \\
& \mathbf{X} \in \mathcal{F} \\
& \mathbf{X} \in \mathcal{I} \cap \mathcal{G}
\end{aligned}
$$

$$\simeq$$

$$
\begin{aligned}
\text{find} \quad & \mathbf{X} = [\mathbf{p}_1 \dots \mathbf{p}_m] \\
& \mathbf{C} = [\mathbf{c}_1 \dots \mathbf{c}_n] \\
\min \quad & \sum_{i=1}^{m} ||\mathbf{c}_i||_1 \\
\text{s.t.} \quad & \mathbf{S}_j \mathbf{p}_i - \mathbf{1} c_{i,j} \leq \mathbf{s}_j \forall i, \forall j \\
& \mathbf{X} \in \mathcal{F} \\
& \mathbf{X} \in \mathcal{I} \cap \mathcal{G}
\end{aligned}
$$

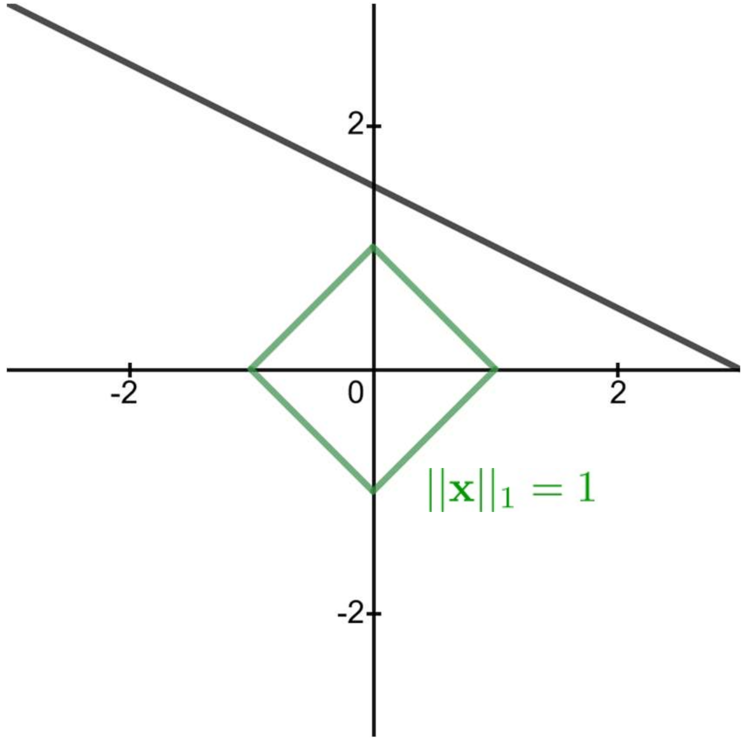
L1 Norm induces sparsity

$$\begin{aligned} &\text{find} \quad && \mathbf{x} = [x_1, x_2] \\ &\text{min} \quad && card(\mathbf{x}) \\ &\text{s.t.} \quad && x_2 = 1.5 \ - 0.5 \ x_1 \end{aligned}$$



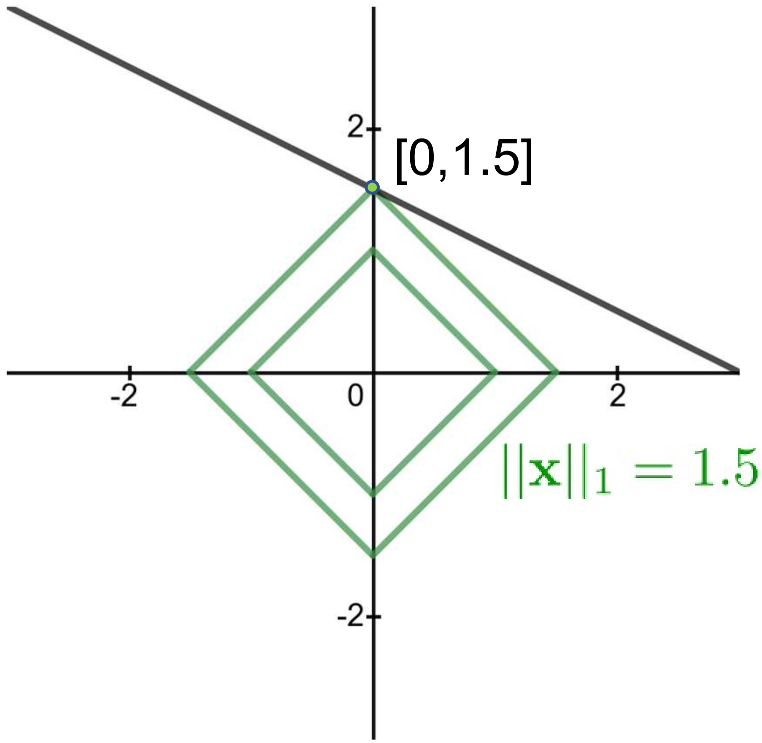
L1 Norm induces sparsity

$$\begin{array}{ll} \text{find} & \mathbf{x} = [x_1, x_2] \\ \text{min} & \cancel{card}(\mathbf{x})\ ||\mathbf{x}||_1 \\ \text{s.t.} & x_2 = 1.5\ - 0.5\ x_1 \end{array}$$



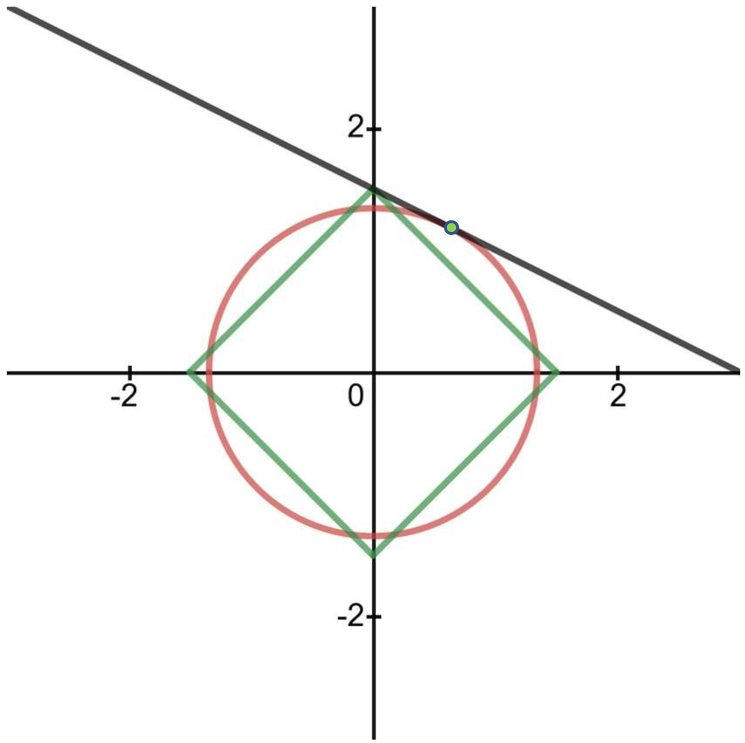
L1 Norm induces sparsity

find	$\mathbf{x} = [x_1, x_2]$
min	~~$card$~~$(\mathbf{x})$ $\vert\vert\mathbf{x}\vert\vert_1$
s.t.	$x_2 = 1.5 - 0.5\, x_1$



Euclidian norm does not lead to sparsity

$$\begin{array}{ll} \text{find} & \mathbf{x} = [x_1, x_2] \\ \text{min} & \sout{card}(\mathbf{x})\ ||\mathbf{x}||_2 \\ \text{s.t.} & x_2 = 1.5 - 0.5\, x_1 \end{array}$$

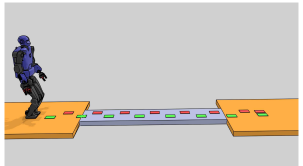
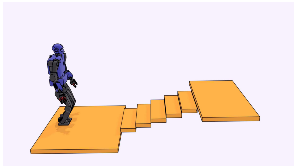
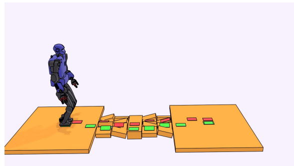
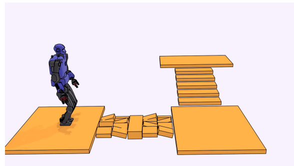


Convex relaxation with sparsity inducing norm

$$
\begin{aligned}
\text{find} \quad & \mathbf{X} = [\mathbf{p}_1 \dots \mathbf{p}_m] \\
& \mathbf{C} = [\mathbf{c}_1 \dots \mathbf{c}_n] \\
\min \quad & \sum_{i=1}^{m} card(\mathbf{c}_i) \\
\text{s.t.} \quad & \mathbf{S}_j \mathbf{p}_i - \mathbf{1} c_{i,j} \leq \mathbf{s}_j \forall i, \forall j \\
& \mathbf{X} \in \mathcal{F} \\
& \mathbf{X} \in \mathcal{I} \cap \mathcal{G}
\end{aligned}
\quad \simeq \quad
\begin{aligned}
\text{find} \quad & \mathbf{X} = [\mathbf{p}_1 \dots \mathbf{p}_m] \\
& \mathbf{C} = [\mathbf{c}_1 \dots \mathbf{c}_n] \\
\min \quad & \sum_{i=1}^{m} \boxed{||\mathbf{c}_i||_1} \\
\text{s.t.} \quad & \mathbf{S}_j \mathbf{p}_i - \mathbf{1} c_{i,j} \leq \mathbf{s}_j \forall i, \forall j \\
& \mathbf{X} \in \mathcal{F} \\
& \mathbf{X} \in \mathcal{I} \cap \mathcal{G}
\end{aligned}
$$

Does it really work ?

Does it really work ?

				
SL1M	V	V	X	X

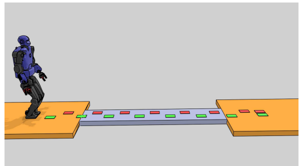
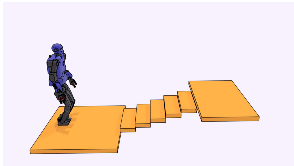
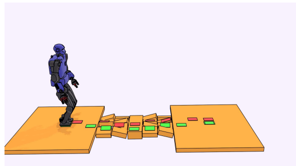
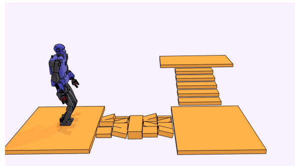
L1 relaxation is faster, but less reliable

L1 up to 10x faster than Mixed-Integer (MI) solvers…

What if relaxed problem does not converge to sparse solution ?

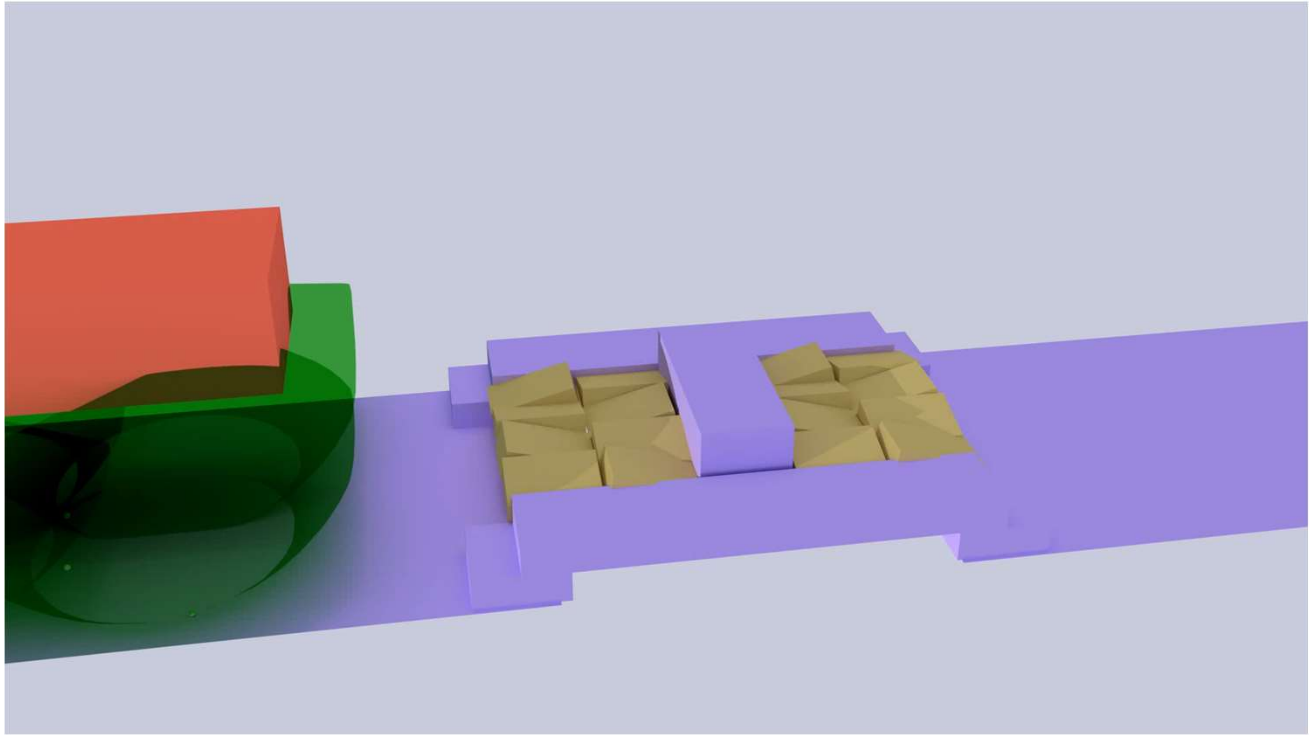
- Can we guarantee convergence ?
- A more general question: Is this really a combinatorial problem?

Let’s look at the number of nodes explored by MI:

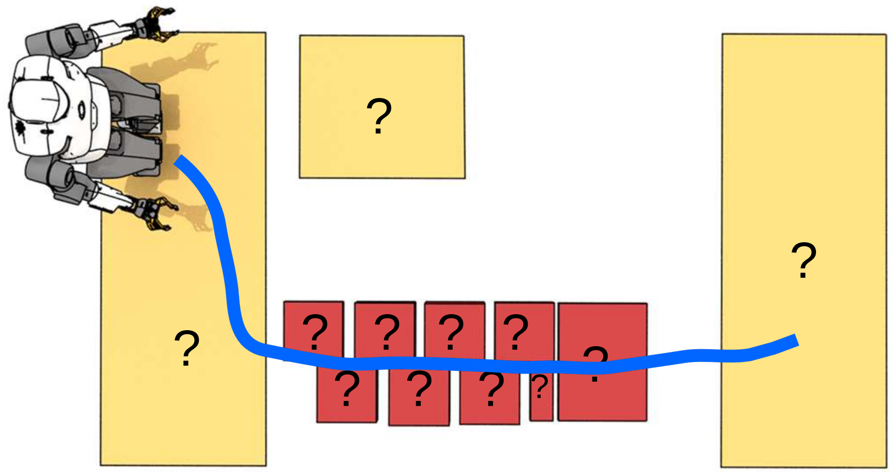
				
MIP (+cost)	**0 (1)**	**0 (1)**	**383 (2742)**	**2724 (20452)**
SL1M	**V**	**V**	**X**	**X**

Let’s do some pruning !

COMPLEXITY CAN BE REDUCED WITH A GUIDE



COMPLEXITY CAN BE REDUCED WITH A GUIDE



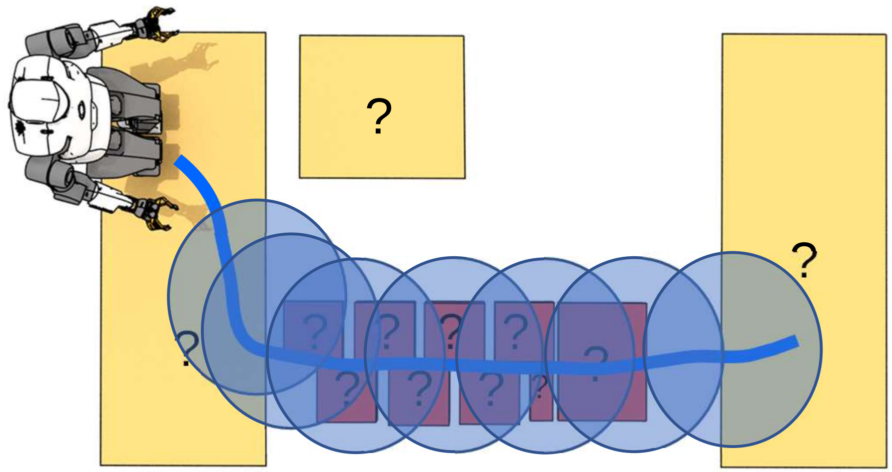
Contact surfaces

$$\mathcal{S}_j, 1 \leq j \leq n$$

Steps:

$$\mathbf{p}_i, 1 \leq i \leq m$$

COMPLEXITY CAN BE REDUCED WITH A GUIDE



Contact surfaces

$$\mathcal{S}_j, 1 \leq j \leq n$$

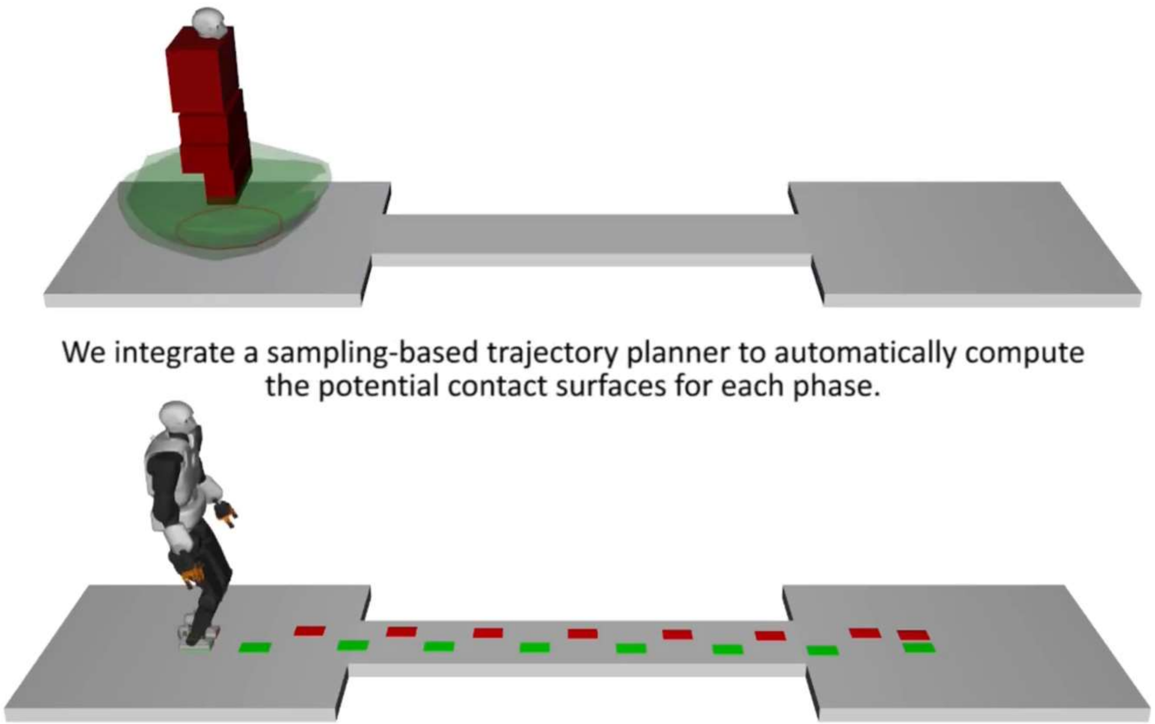
Steps:

$$\mathbf{p}_i, 1 \leq i \leq m$$

Combinatorial:

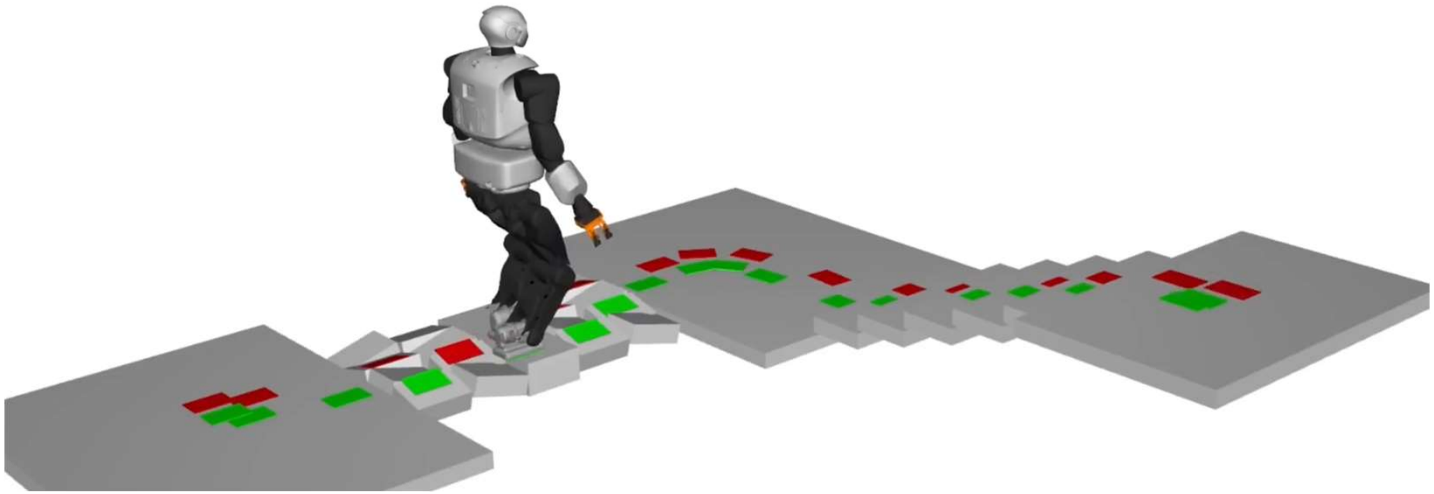
$$n^h, h < m$$

Only hyper parameter is discretisation step

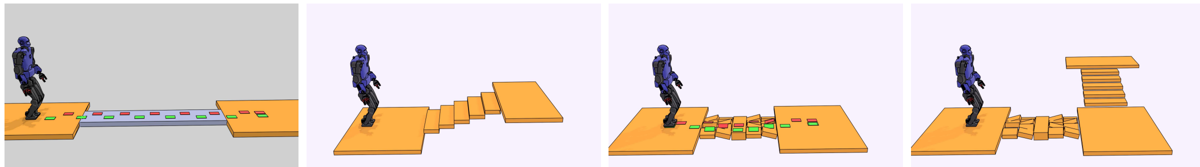


Does it really work ?

Our footstep planning framework for legged robots is capable of computing 30-step long contact sequences on uneven terrain within a second.



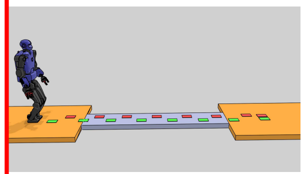
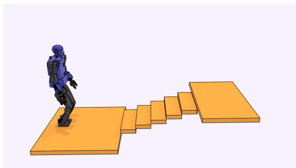
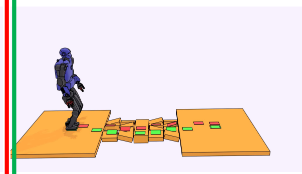
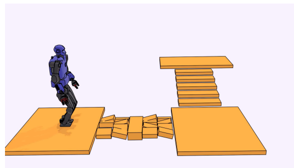
Guide-path improves SL1M convergence, but also MIP performances



L1 success	V	V	X	X
Guide + L1	V	V	V	V

Ok but why bother ? Computational time (ms)

Tradeoff guide cost (about 100 ms) / Solver time

				
Guide + L1	**35**	**20**	**73**	**219**
Guide + MIP	**77**	**34**	**230**	**508**
MIP	**166**	**59**	**389**	**74710**

Early conclusions for footstep planning

- Refining problem definition leads to vanishing combinatorics

    Using domain-specific knowledge seems promising
    Other issues to consider (what is the right path discretisation ?)

- L1-norm relaxation converges to feasible solution faster

- What about more challenging problems (gait selection) ?

- Optimality ? I think we do not care!

Overall conclusions for Contact planning

Should we embrace combinatorics or try to ignore it ?

What can we expect for more challenging tasks ?

ie. Can we « presolve » harder problems ?

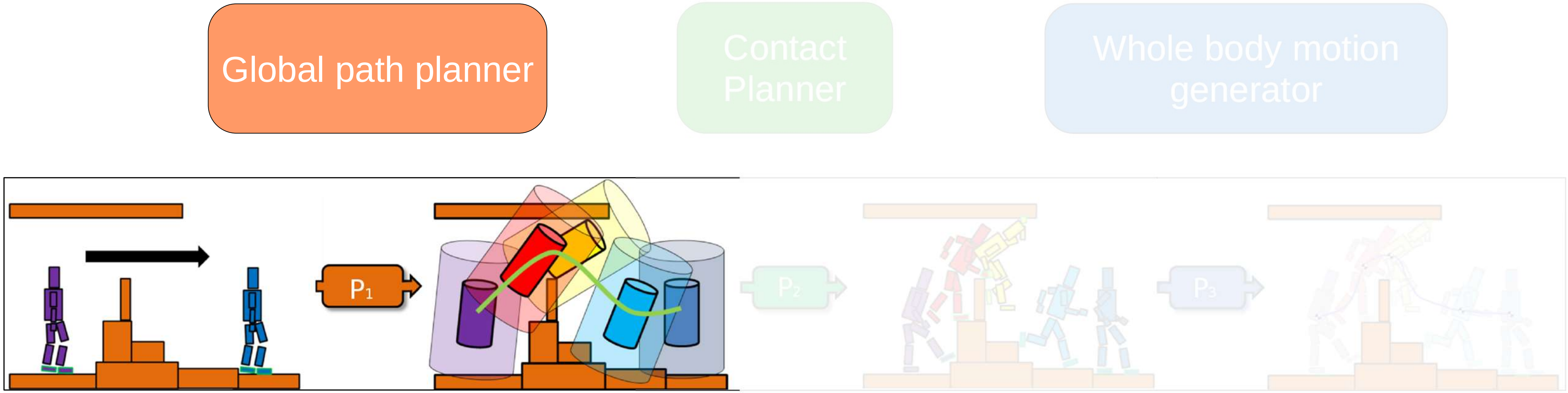
Non-linear problems, unknown gait patterns

I believe those questions are challenging, important and fun

That’s all for now

https://stevetonneau.fr for papers, videos, source code

II. Learning feasible guide trajectories



Collaborators

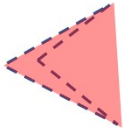
Jason Chemin

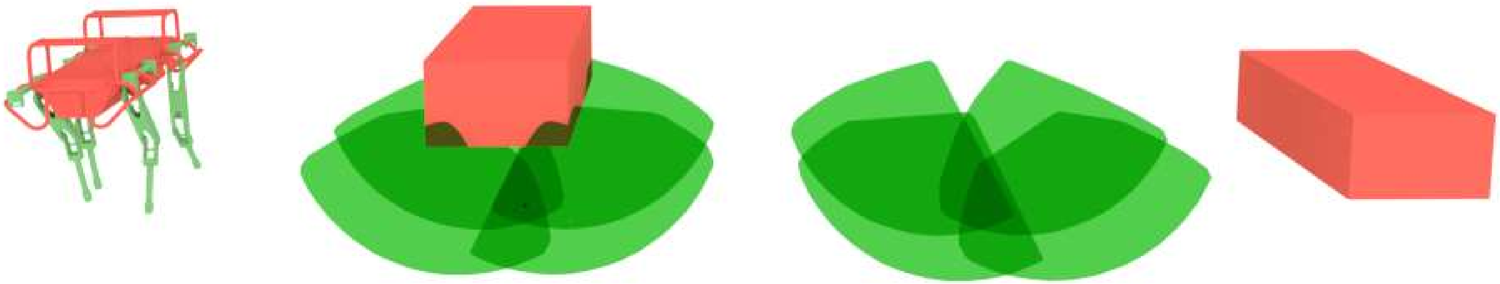
Nicolas Mansard

Gepetto
LAAS-CNRS

Learning to steer a locomotion contact planner, *ICRA* 2021

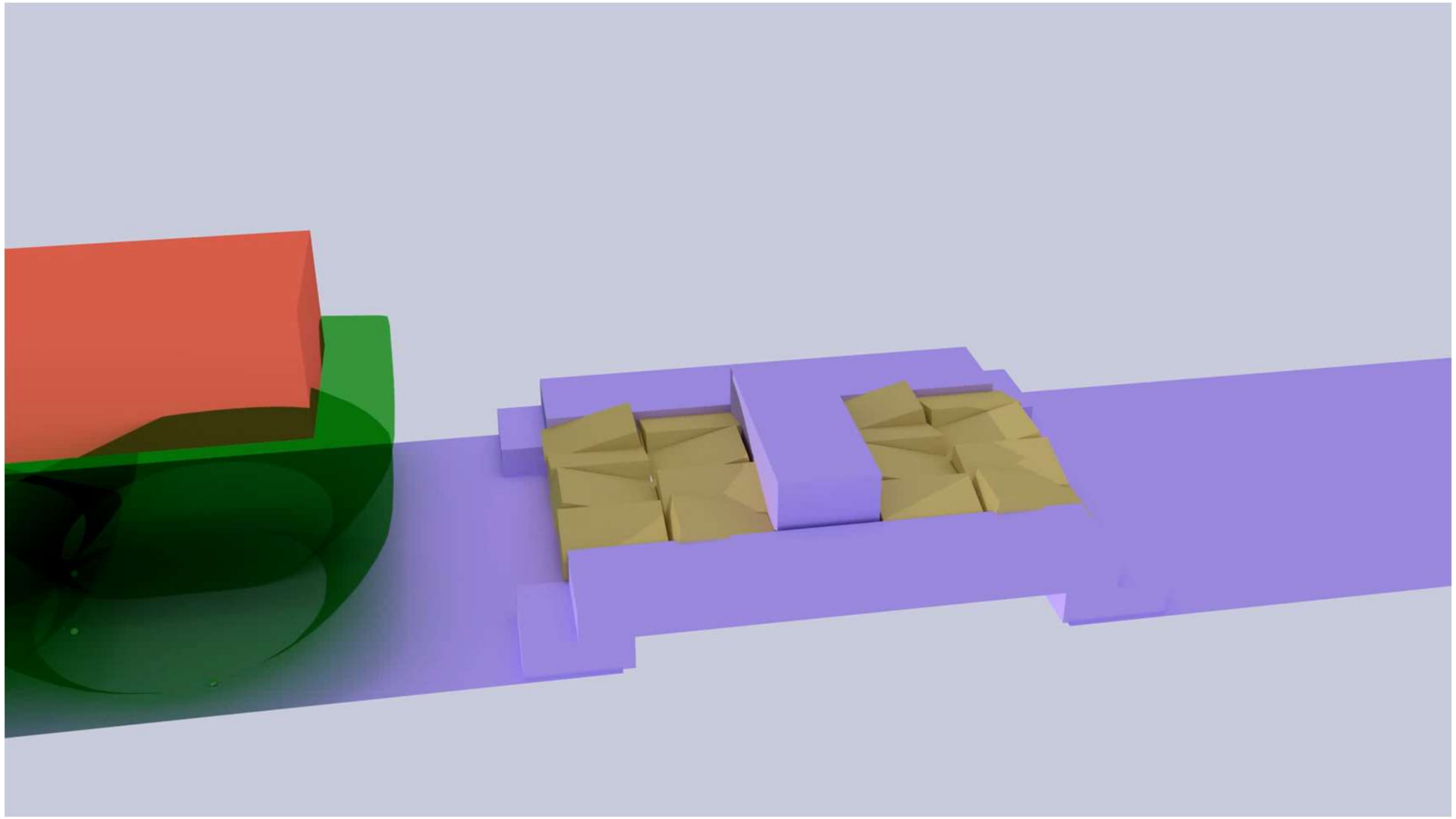
Necessary condition for contact creation

 : Contact surfaces in reachable workspace of effector

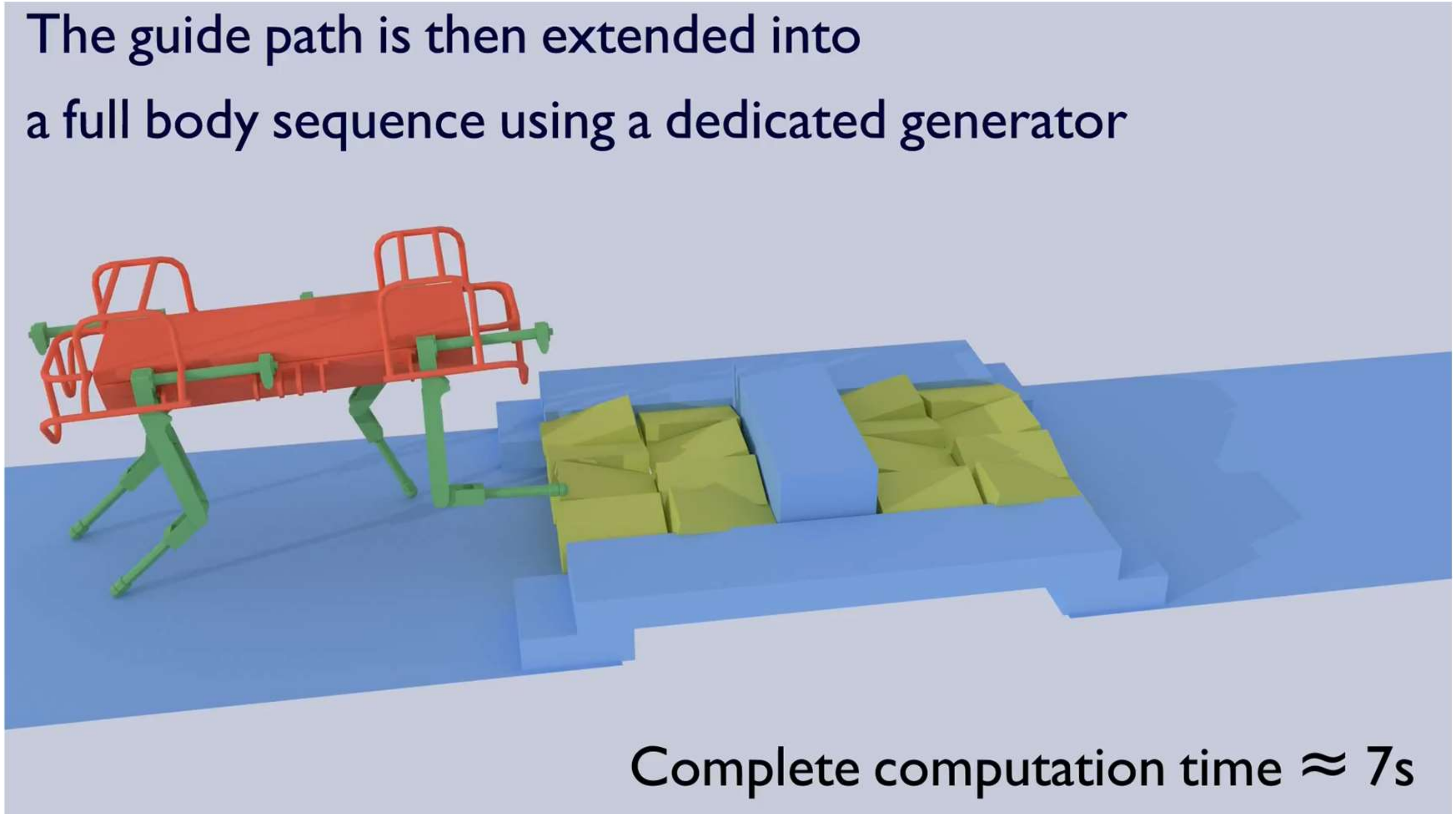


Tonneau et al. 2015 (RSS) / 2019 (TRO)

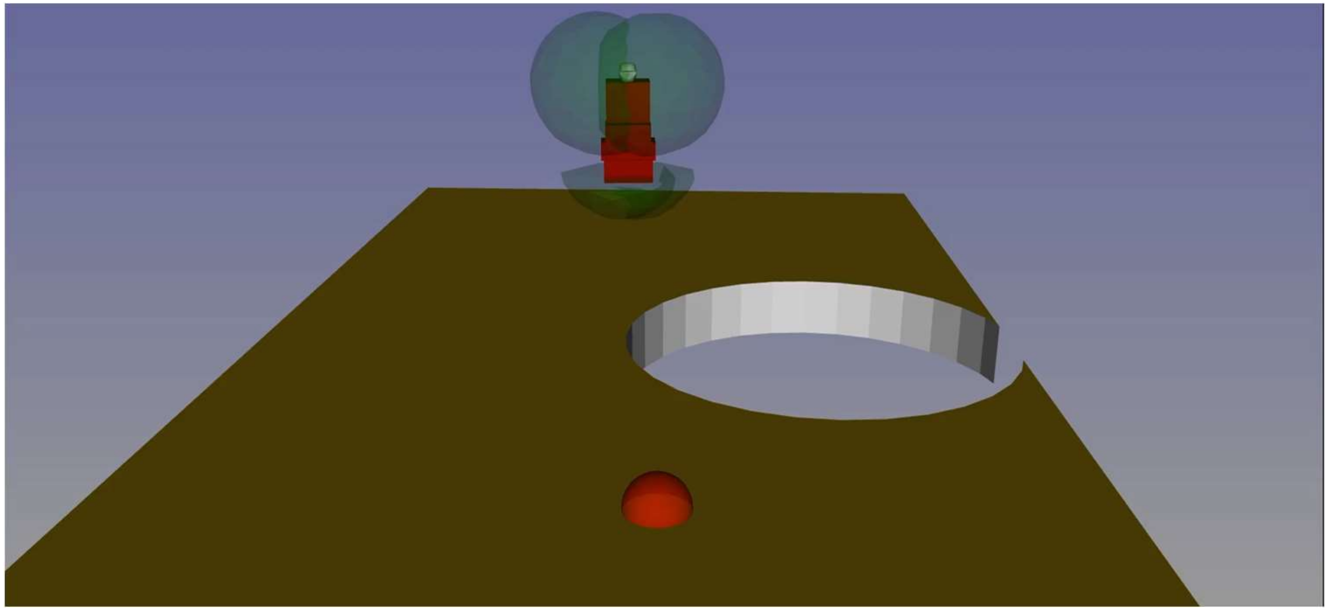
RRT planner to compute guide path in 6D

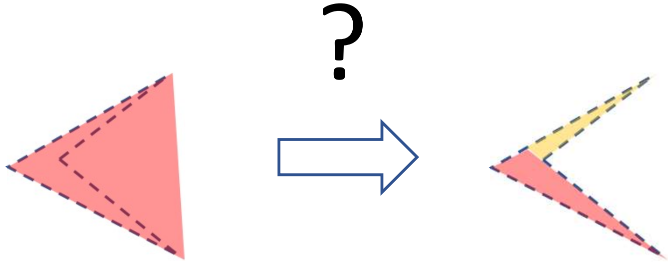


Random contact generator along guide (not SL1M)



Example of failure case



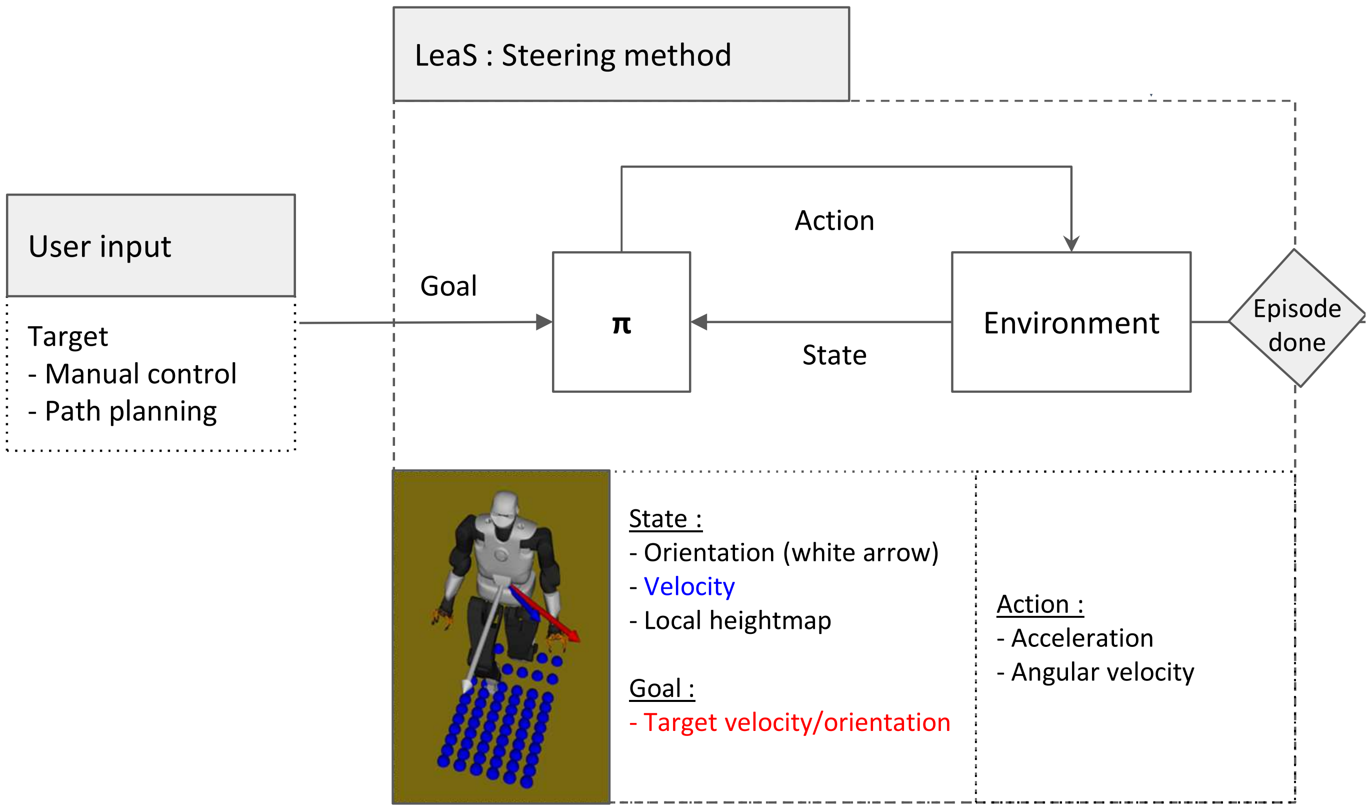


The difficulty lies in the compromise between necessary and sufficient condition

- Active research around the reachability condition since its original contribution, but more on P2 than on the actual condition

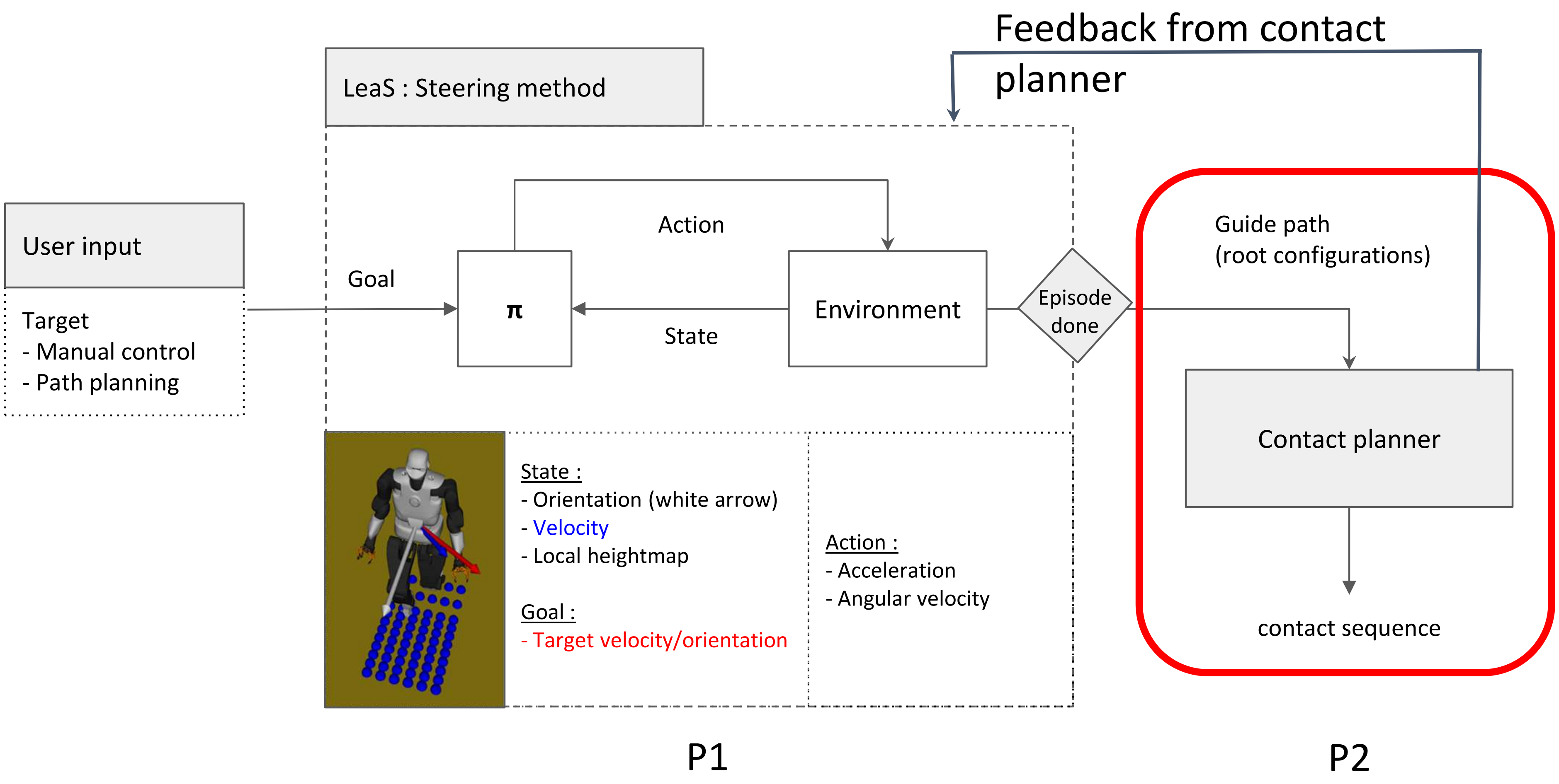
- Our proposition: Reinforcement Learning (RL) framework for improving the reachability condition

Framework



P1

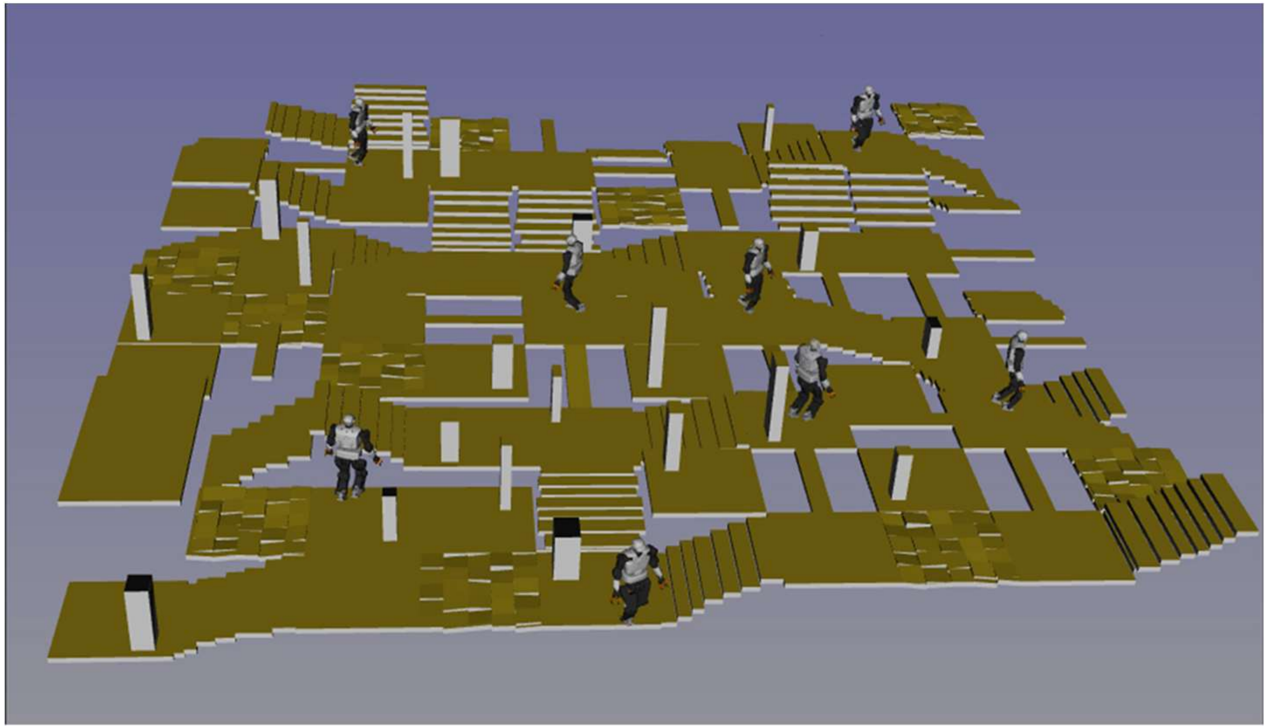
Framework



Feedback from contact planner
LeaS : Steering method
User input
Target
- Manual control
- Path planning
Goal
Action
π
State
Environment
Episode done
Guide path
(root configurations)
Contact planner
contact sequence
State :
- Orientation (white arrow)
- Velocity
- Local heightmap
Goal :
- Target velocity/orientation
Action :
- Acceleration
- Angular velocity
P1
P2

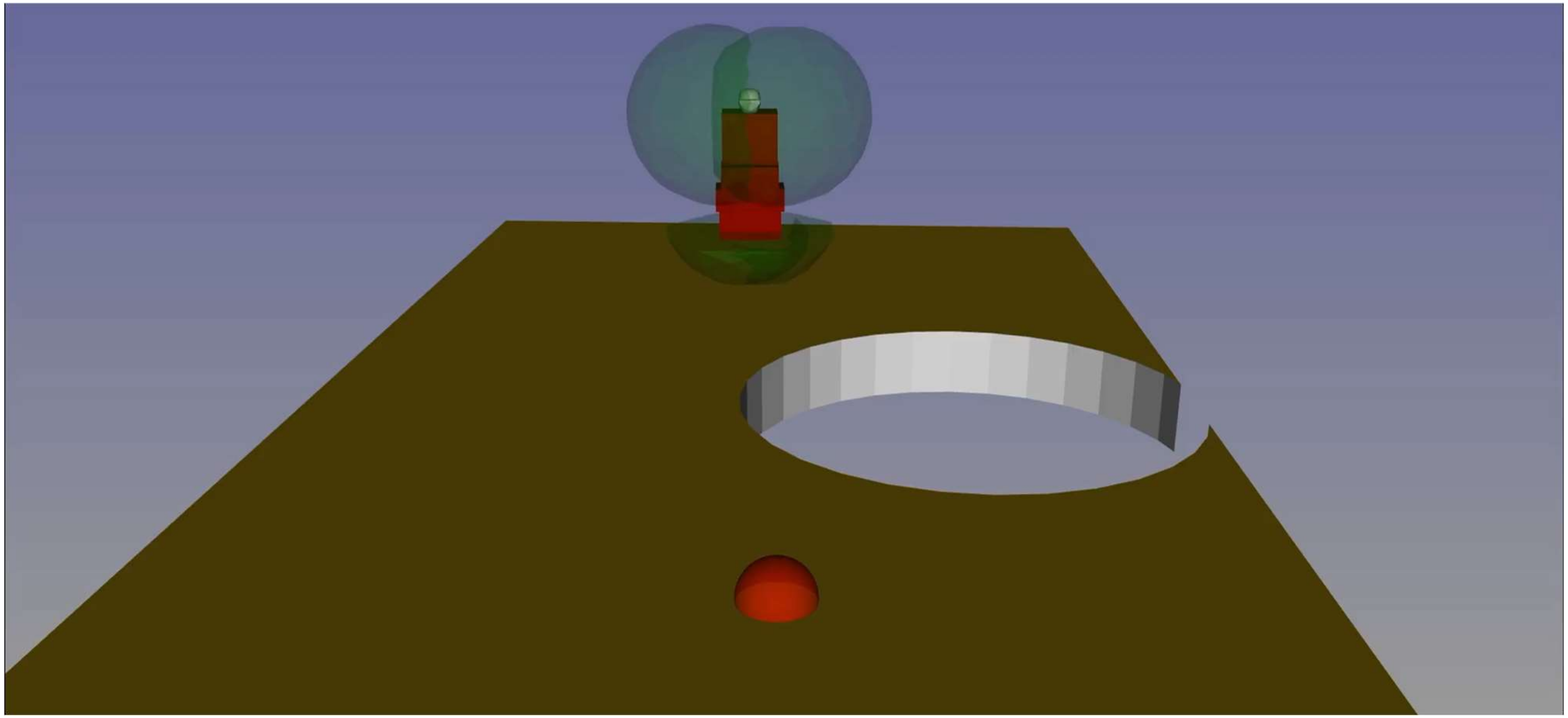
Training

- Arena generator : stairs, obstacles and bridges
- Random initial configuration and target velocity
- Asynchronous version of Proximal Policy optimization (PPO)[1] for 53 hours
- Sample-based contact planner[2]
- Talos model[3]



[1] "Proximal policy optimization algorithms" 2017
[2] "A Reachability-based planner for sequences of acyclic contacts in cluttered environments" 2015
[3] "TALOS: A new humanoid research platform targeted for industrial applications" 2017

Some results….



Early conclusions on RL for path planning

- A robust approach:
  - higher success rates, good generalisation
  - Ongoing tests to quantify exactly to what extent
- Visual quality wrt pure sampling-based approach
- A limitation: no specification of precise target point

- Future work:
  - SL1M as a contact planner to learn constraint tightening
 (Promising early results)
  - Include continuous motion generation in feedback loop. Performance issue in terms of training ? Efficient models for motion generation ?