Autonomous and Mobile Robotics

Whole-body motion planning for humanoid robots

(Slides prepared by Marco Cognetti)

Prof. Giuseppe Oriolo

Dipartimento di Ingegneria Informatica Automatica e Gestionale Antonio Ruberti

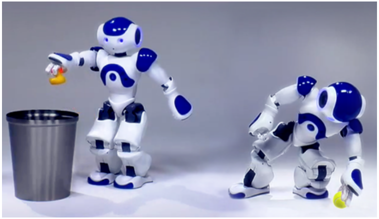
Sapienza Università di Roma

Motivations

task-constrained motion planning: find **collision-free** motions for a humanoid that is assigned a certain task whose execution may require **stepping**

it is challanging for a number of reasons

- high number of degrees of freedom
- it is not a free-flying system – motion must be generated appropriately
- the robot has to maintain equilibrium at all times (constraining the trajectory of the CoM)



Motivations

literature approaches

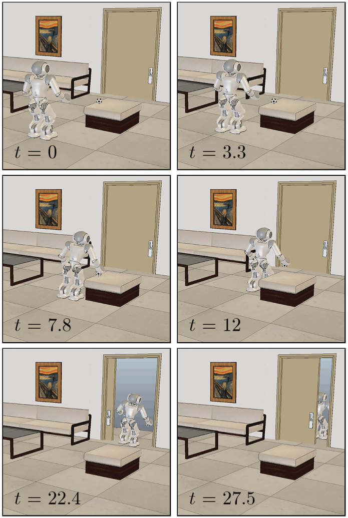
- separate locomotion from task execution (e.g., Burget et al., 2013 and Hauser and Ng-Thow-Hing, 2011)
- compute a collision-free, statically stable paths for a free-flying humanoid base, and then approximate it with a dynamically stable walking motion (e.g., Dalibar et al., 2013)
- achieve acyclic locomotion and task execution through whole-body contact planning (e.g, Bouyarmane and Kheddar, 2012)

our approach

- **does not separate** locomotion from task execution, making advantage of the whole-body structure of the humanoid
- walking emerges **naturally** from the solution of the planning problem

Overview

goal: plan **whole-body** motion for a humanoid over $[t_i, t_f]$ that must execute a **task** assigned as a composition navigation and manipulation actions



assumptions:

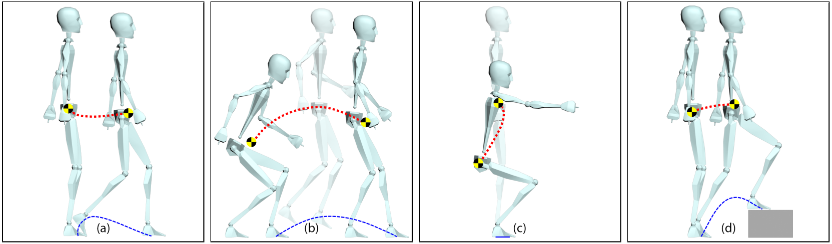
- the task is already assigned by an higher level task-planner
- robot configuration $\mathbf{q} = \begin{pmatrix} \mathbf{q}_{\mathrm{CoM}} \\ \mathbf{q}_{\mathrm{jnt}} \end{pmatrix}$, where

  $\mathbf{q}_{\mathrm{CoM}}$ is the planar pose of CoM frame
  $\mathbf{q}_{\mathrm{jnt}}$ is the n-vector of joint angles
- a catalogue of precomputed trajectories for the CoM is avalaible to the planner
- each primitive has a given duration T_k

solution:

- sequence of elementary joint motions, $\mathbf{q}_{\mathrm{jnt}}(t)$, either stepping or non-stepping

Humanoid motion model



CoM movement primitive: trajectories for the CoM and possibly other parts (e.g., swing foot) associated to typical human actions (e.g., walking, jumping, squatting)

$$\begin{aligned} \mathbf{q}_{\text{CoM}}(t) &= \mathbf{q}^k_{\text{CoM}} + \mathbf{A}(\mathbf{q}^k_{\text{CoM}})\mathbf{u}^k_{\text{CoM}}(t) \\ \dot{\mathbf{q}}_{\text{jnt}}(t) &= \mathbf{v}_{\text{jnt}}(t) \qquad t \in [t_k, t_k + T_k]. \end{aligned}$$

where

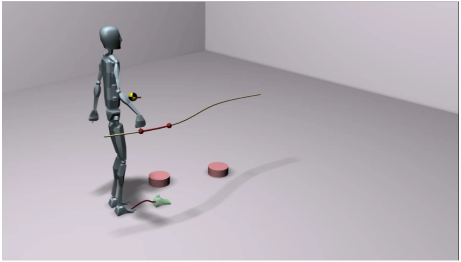
- $\mathbf{A}(\mathbf{q}^k_{\text{CoM}})$ is the transformation matrix from the CoM at $\mathbf{q}^k$ to the world frame
- $\mathbf{u}^k_{\text{CoM}}$ is the pose displacement of the CoM frame relative to the pose at $\mathbf{q}^k$
- $\mathbf{v}_{\text{jnt}}$ is the velocity command vector for the humanoid joints
- a catalogue of movements is **precomputed** (each with a given duration T_k)

Task definition

main task: assigned trajectory $\mathbf{y}^*(t)$ (or a geometric path $\mathbf{y}^*(s)$ or a single point $\mathbf{y}^*(t_f)$) for a relevant point on the robot (e.g., a manipulation task for one hand).

formal definition of the solution

- the assigned task trajectory is exponentially realized
$$\lim_{t\to\infty}(\mathbf{y}(t) - \mathbf{y}^*(t)) = \mathbf{0}$$
- collisions with workspace obstacles and self-collisions are avoided
- position and velocity limits on the robot joints are satisfied
$\mathbf{q}_{\rm jnt,m} < \mathbf{q}_{\rm jnt} < \mathbf{q}_{\rm jnt,M}$ and $\mathbf{v}_{\rm jnt,m} < \mathbf{v}_{\rm jnt} < \mathbf{v}_{\rm jnt,M}$



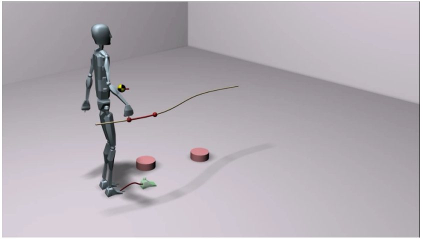
Motion generation

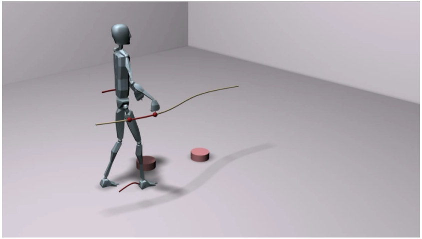
the planner works in an iterative fashion by repeated calls to a **motion generation**

the motion generator is invoked to produce a **feasible**, **collision-free** elementary motion that realizes a **portion** of the assigned task trajectory

we use two interleaved procedures to generate the motions for $\mathbf{q}_{\mathrm{CoM}}$ and $\mathbf{q}_{\mathrm{jnt}}$

1. *CoM movement selection*: choose a particular CoM movement from the set of primitives. It generates trajectories the CoM (and other points) of the robot
2. *joint motion generation*: compute the joint velocity commands. It takes into account the trajectories generated in step 1 and the assigned task





1 - CoM movement selection

it is invoked at t_k and selects $\mathbf{u}^k_{\mathrm{CoM}}$, a *CoM movement* among the set of *CoM primitives*

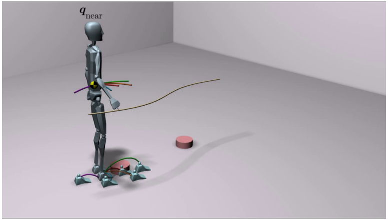
$$U = \left\{ U^{\mathrm{S}}_{\mathrm{CoM}} \cup U^{\mathrm{D}}_{\mathrm{CoM}} \cup \texttt{free\_CoM} \right\}$$

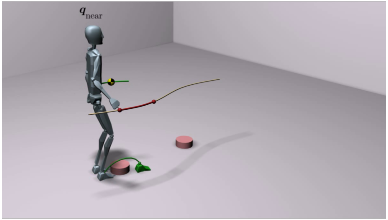
where

- $U^{\mathrm{S}}_{\mathrm{CoM}}, U^{\mathrm{D}}_{\mathrm{CoM}}$: **static/dynamic** stepping movements
- `free_CoM`: CoM **free** to move with both feet fixed and the robot in equilibrium

each primitive has a **duration**. As output, the CoM movement selector gives

- a duration T_k for the movement and a time interval $[t_k, t_{k+1}]$
- a trajectory $\mathbf{z}^*_{\mathrm{CoM}}$ for the humanoid CoM in $[t_k, t_{k+1}]$, where $t_{k+1} = t_k + T_k$
- a trajectory $\mathbf{z}^*_{\mathrm{swg}}$ in $[t_k, t_{k+1}]$ for the swing foot (position and orientation)



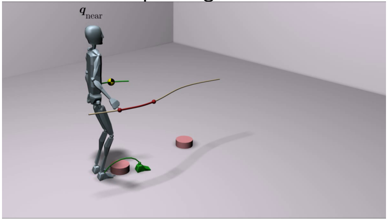


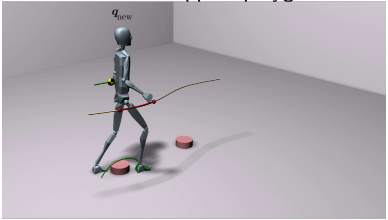
2 - Joint motion generation

given the output of the CoM movement selector $(\mathbf{z}^*_{\text{swg}}, \mathbf{z}^*_{\text{CoM}})$, the joint motion generator produces joint configurations in $[t_k, t_{k+1}]$ through

$$\mathbf{v}_{\text{jnt}} = \mathbf{J}_a^\dagger(\mathbf{q}_{\text{jnt}})\,(\dot{\mathbf{y}}_a^* + \mathbf{K}\mathbf{e}) + (\mathbf{I} - \mathbf{J}_a^\dagger(\mathbf{q}_{\text{jnt}})\mathbf{J}_a(\mathbf{q}_{\text{jnt}}))\mathbf{w},$$

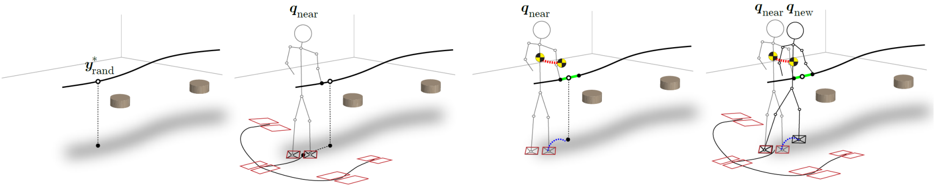
- $\mathbf{y}_a = (\mathbf{y}^T \;\; \mathbf{z}^T_{\text{swg}} \;\; \mathbf{z}^T_{\text{CoM}})^T$: augmented task vector
- $\mathbf{J}_a$: Jacobian matrix of $\mathbf{y}_a$ with respect to $\mathbf{q}_{\text{jnt}}$
- $\mathbf{e} = \mathbf{y}_a^* - \mathbf{y}_a$: augmented task error ($\mathbf{y}_a^*$ desired value of $\mathbf{y}_a$)
- $\mathbf{J}_a^\dagger$: pseudoinverse of $\mathbf{J}_a$. $\mathbf{K} > 0$
- $\mathbf{w}$: random n-vector Two possibilities:
  - walking: $\mathbf{w} = \mathbf{w}_{\text{rnd}}$, where $\mathbf{w}_{\text{rnd}}$ is a random bounded-norm vector
  - non-walking: $\mathbf{w} = -\eta \cdot \nabla_{\mathbf{q}_{\text{jnt}}} H(\mathbf{q}_{\text{jnt}}) + \mathbf{w}_{\text{rnd}}$, $\quad \eta > 0$, whose aim is to add an action pushing the CoM towards the centroid of the support polygon





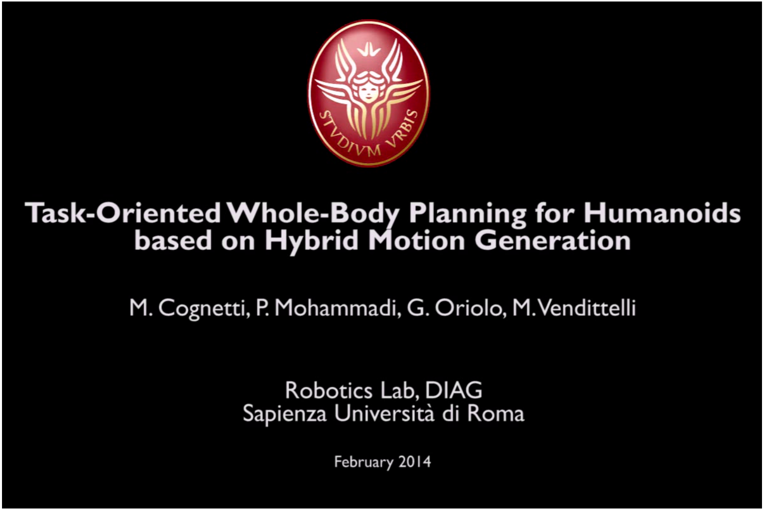
The planner

for each iteration



- (a): a random sample $\mathbf{y}^*_{\mathrm{rand}}$ is chosen from the assigned task trajectory, and its projection on the ground is computed
- (b): an high-compatibility configuration $\mathbf{q}_{\mathrm{near}}$ is extracted from the tree
- (c): a CoM movement of a certain duration is selected from catalogue. It defines reference trajectories for CoM and swing foot (dashed red and blue) as well as the portion of the assigned task to be executed (green)
- (d): joint motion is generated to simultaneously realize the chosen CoM movement and the portion of the task
- if the motion is feasible and collision-free, the final configuration $\mathbf{q}_{\mathrm{new}}$ is added to the tree; otherwise, the procedure is repeated

Planning experiments: video



Planning experiments: video

Motivation

task constraints (either explicit or implicit) may prevent motion planners from finding a feasible solution

Overview

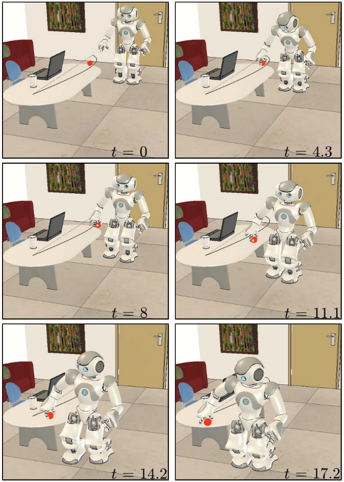
main limitation:
the task is supposed to be **assigned a priori**

features:

- the task is automatically modified, if needed
- the task is modified iteratively
- the humanoid model is still
  $\mathbf{q} = \begin{pmatrix} \mathbf{q}_{\rm CoM} \\ \mathbf{q}_{\rm jnt} \end{pmatrix}$, where
  $\mathbf{q}_{\rm CoM}$ is the planar pose of CoM frame
  $\mathbf{q}_{\rm jnt}$ is the n-vector of joint angles
- a catalogue of precomputed trajectories for the CoM is available to the planner
- the task is a **deformable** curve whose shape may be deformed acting on control points $\boldsymbol{\sigma}$

solution:

- sequence of elementary joint motions $\mathbf{q}_{\rm jnt}(t)$



Task definition

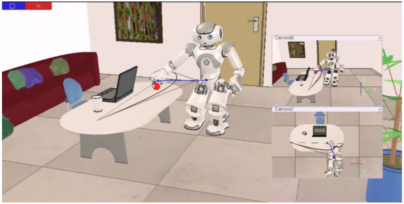
initial task: path $\mathbf{y}^{[0]}(s)$ and a time history $s^{[0]}(t)$ in $[t_i, t_f]$ for a relevant point on the robot (e.g., one hand)

formal definition of the solution

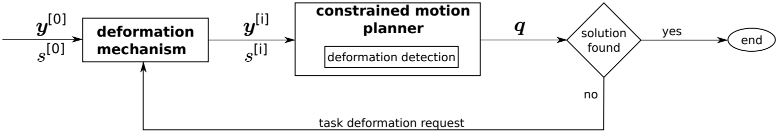
- the final task trajectory is realized:

$$\mathbf{f}(\mathbf{q}^*(t)) = \mathbf{y}^*(s^*(t)) = \mathbf{y}^*(t), \quad t \in [t_i, t_f]$$

- the robot maintains static or dynamic equilibrium
- all collisions with obstacles are avoided
- joint limits and velocity bounds, respectively expressed in the form $\mathbf{q}_{\mathrm{jnt,m}} < \mathbf{q}_{\mathrm{jnt}} < \mathbf{q}_{\mathrm{jnt,M}}$ and $\mathbf{v}_{\mathrm{jnt,m}} < \mathbf{v}_{\mathrm{jnt}} < \mathbf{v}_{\mathrm{jnt,M}}$, are satisfied



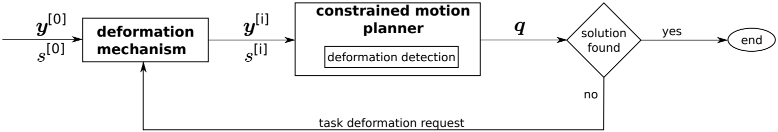
Overview



main idea

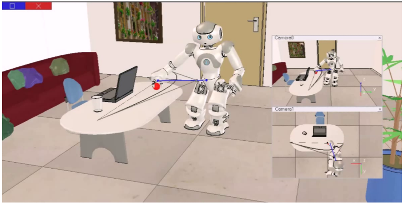
- the **constrained motion planner** is invoked on $\mathbf{y}^{[0]}$
- the algorithm proceeds until either it reaches the **final task point** $\mathbf{y}(t_f)$ or it repeatedly fails to go beyond a certain $\tilde{\mathbf{y}} \neq \mathbf{y}(t_f)$
- in the second case, the task is modified using a **deformation mechanism** which acts at the **limit task point** $\tilde{\mathbf{y}}$ (defined as the furthermost task point met by the tree)
- the constrained motion planner is invoked again on the deformed task
- this deformation-planning cycle is repeated until a solution is found

What is missing...

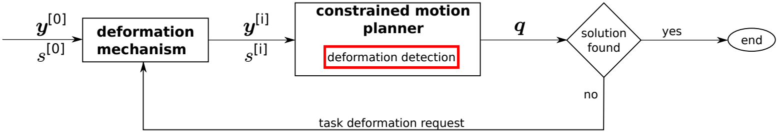


two missing components (w.r.t. previous work)

- detect when the task path $\mathbf{y}^{[i]}(s)$ has to be deformed
  chosen policy
  the number of failed expansions from the **limit task point** $\tilde{\mathbf{y}}$ exceeds a predefined threshold
- how to deform the assigned task path $\mathbf{y}^{[i]}(s)$

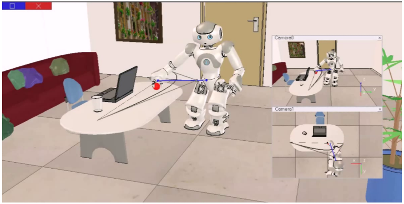


What is missing...

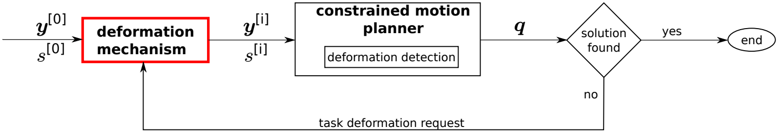


two missing components (w.r.t. previous work)

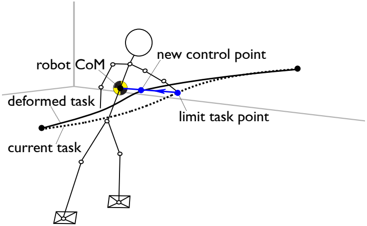
- detect when the task path $\mathbf{y}^{[i]}(s)$ has to be deformed
  chosen policy
  the number of failed expansions from the **limit task point** $\tilde{\mathbf{y}}$ exceeds a predefined threshold
- how to deform the assigned task path $\mathbf{y}^{[i]}(s)$



What is missing...

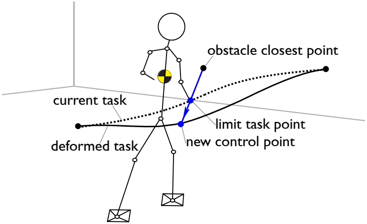


robot-based heuristic



idea: path closer to the CoM

obstacle-based heuristic



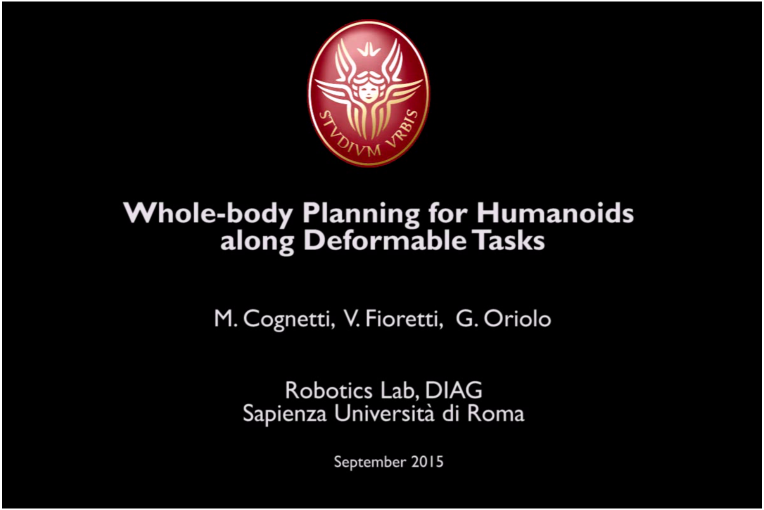
idea: path away from obstacle

chosen policy

if expansions from $\tilde{\mathbf{y}}$ fail for **joint limit violations**, robot-based heuristic is chosen

if expansions from $\tilde{\mathbf{y}}$ fail for **collisions**, obstacle-based heuristic is chosen

Planning experiments: video



Publications

- **M. Cognetti**, P. Mohammadi, G. Oriolo, M. Vendittelli. "Task-Oriented Whole-Body Planning for Humanoids based on Hybrid Motion Generation". In 2014 IEEE/RSJ Int. Conf. on Intelligent Robots & Systems Systems (IROS14), Chicago, Illinois, USA, Sep. 2014

- **M. Cognetti**, P. Mohammadi, G. Oriolo. "Whole-Body Motion Planning for Humanoids based on CoM Movement Primitives". In 15th IEEE-RAS Int. Conf. on Humanoid Robots, Seoul, Korea, Nov. 2015

- **M. Cognetti**, V. Fioretti, G. Oriolo. "Whole-body Planning for Humanoids along Deformable Tasks". In 2016 IEEE Int. Conf. on Robotics and Automation (ICRA16), Stockholm, Sweden, May 2016

Thank you!