2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)
November 3 - 5, 2015, Seoul, Korea

Whole-Body Motion Planning for Humanoids based on CoM Movement Primitives

Marco Cognetti, Pouya Mohammadi, Giuseppe Oriolo

Abstract— **This work addresses the problem of whole-body motion planning for a humanoid robot that must execute a certain task in an environment containing obstacles. A randomized planner is proposed that builds a solution by concatenating whole-body motions. Each whole-body motion is generated so as to realize a center of mass (CoM) movement selected from a set of primitives and simultaneously accomplish a portion of the task. The CoM primitives are representative of typical humanoid actions such as walking gaits (static and dynamic), and can in principle include more sophisticated movements (e.g., jumping, crouching, etc). Implementation on the NAO humanoid proves that the proposed method generates sensible plans for a variety of composite tasks requiring a combination of navigation and manipulation.**

I. Introduction

While most early work on humanoid robots focused on the problem of achieving stable and efficient locomotion, researchers are now turning their attention to more articulated tasks, that may use locomotion as building block but in general require complex whole-body motions. These motions should be feasible, i.e., comply with the various kinematic and dynamic constraints of the specific robot; at the same time, collisions should be avoided with the obstacles that are invariably present in the robot workspace.

On-line generation of task-constrained robot motions is typically achieved using kinematic control; a well-known example is the task-priority framework [1], which can be extended to handle inequality constraints for collision avoidance [2]. This framework was applied to humanoid footstep generation for manipulation tasks in [3]. However, kinematic control is a local technique: in complex situations, it may fail to find a solution even if one exists. If the scene geometry is known, a deliberative approach based on motion planning is expected to outperform any reactive approach.

Even in the simple configuration-to-configuration case, motion planning for humanoids is a challenging problem in view of the high dimension of the configuration space, the inherent underactuation of the mechanism, and the necessity of maintaining some form of equilibrium. A further complication is met when the robot is assigned a task. This may be a single operation (e.g., 'grasp this object', 'open the door handle') or a composite sequence of navigation and manipulation actions ('take the object on that table and bring it in the other room'). This is exactly the kind of task-constrained motion planning we consider in this paper.

The authors are with the Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Sapienza Università di Roma, via Ariosto 25, 00185 Roma, Italy. E-mail: {cognetti, oriolo}@diag.uniroma1.it. This work is supported by the EU Horizon 2020 project COMANOID.

In [4], we have proposed a planning approach that does not separate locomotion from task execution. Building on the task-constrained planning framework of [5], a randomized planner is proposed that simultaneously generates foot displacements and whole-body motions. As a result, the planner was naturally able to generate walking motions when these are implicitly required by the task.

Here, we generalize our previous planner by replacing foot displacements with movements of the center of mass (CoM). These movements are representative of typical humanoid actions, such as static walking, dynamic walking, and more. Assuming that a catalogue of CoM movement primitives has been precomputed, solutions are built by concatenating feasible whole-body motions that realize such primitives and simultaneously accomplish portions of the task. Another important aspect under which the new planner improves over the previous is that it can indifferently handle tasks specified as trajectories or as simple destinations in the task space.

Results obtained on a NAO humanoid will confirm the higher versatility gained by the use of CoM movements primitives. For example, we obtain plans that automatically toggle between dynamic and static walking gaits when required by the characteristics of the environment (e.g., obstacles) or the task itself. In addition, the possibility of assigning destination points allows a simple definition of composite tasks.

The paper is organized as follows. In the next section, a quick review of the related literature is presented. In Section III we introduce a humanoid motion model and formulate our planning problem. Motion generation based on CoM movement primitives is described in Section IV, and a randomized planner is proposed in Section V. Motion planning experiments for the NAO humanoid robot are presented in Section VI. Future research directions are briefly discussed in Section VII.

II. Related work

Many motion planning methods for humanoids use simplifying assumptions on either the environment or the robot geometry. One such approach consists in checking collisions, at least in a first phase, only at the footstep [6], [7] or at the leg [8] level; although these techniques have the merit of simplicity, the final motion may need reshaping due to collisions not predicted by the simplified model. Another approach is to find a collision-free path for a simplified robot model (e.g., its bounding box) and then convert this path into a feasible locomotion trajectory: this is done in [9] for digital characters, and in [10] for a humanoid robot. Due to the coarse approximation of the robot occupancy, this method may fail to find a solution in cluttered environments; for example, extending over a table to pick up something is not possible. In [11], a whole-body planner is presented that explores the whole configuration space of the humanoid robot by first computing collision-free motions and then converting them into dynamically stable motions. The technique in [12] is one example of simultaneous generation of footsteps and whole-body motions. Most of the above methods do not directly incorporate task constraints.

Task-constrained motion planning for humanoids has been addressed in [13] for the fixed-stance case (the robot cannot take steps) or separating locomotion from task execution in [14]. The approach in [15] works in two stages: first, statically stable collision-free paths are computed for a free-flying humanoid robot; then, these paths are converted to dynamically stable humanoid motions. Reshaping of the final motion may be needed due to collisions appearing only in the second phase. In [16], acyclic locomotion and task execution are achieved through whole-body contact planning at specified locations. With respect to these works, our planner does neither rely on any form of separation nor require the advance specification of footsteps: stepping motions emerge naturally from the solution of the planning problem as driven by the assigned task.

Primitives have been used, e.g., for planning humanoid motions on varied terrain [17]. However, those primitives are actually whole-body motions (they specify the motion of all joints) while in our approach they only describe the trajectory of the CoM and can thus map to different robot movements as required by the various phases of the plan; this gives a higher *plasticity* to our planning method. Moreover, the above planners cannot directly handle the case in which a task (e.g., manipulation) is assigned to the robot.

III. Problem Formulation

We first introduce a motion model for a humanoid robot, and then discuss the nature of the considered tasks.

A. Humanoid Motion Model

Planning for a humanoid requires an appropriate definition of the configuration space. Denote by n the number of articulation variables (joint angles) of the humanoid. In general, to specify the configuration of a free-flying humanoid one should assign the n values of the joint angles as well as the pose (position and orientation) of a reference frame linked to one of the robot bodies. In this paper, we will use to this end a frame attached to the center of mass (CoM in the following) and oriented as the torso. A configuration will be then defined as follows

$$\boldsymbol{q} = \begin{pmatrix} \boldsymbol{q}_{\mathrm{CoM}} \\ \boldsymbol{q}_{\mathrm{jnt}} \end{pmatrix},$$

where $\boldsymbol{q}_{\mathrm{CoM}} \in SE(3)$ is the pose of the CoM frame and $\boldsymbol{q}_{\mathrm{jnt}} \in \mathcal{C}_{\mathrm{jnt}}$ is the n-vector of joint angles. The humanoid configuration space $SE(3) \times \mathcal{C}_{\mathrm{jnt}}$ has thus dimension $n+6$.

The natural partitioning of the configuration vector $\boldsymbol{q}$ also reflects the different way in which motion will be generated within our planner. For the CoM, we shall concatenate whole movements (i.e., subtrajectories) rather than defining instantaneous motions. These subtrajectories will be extracted from a catalogue of *CoM movement primitives* that are associated to typical humanoid actions such as walking, jumping, squatting, etc. (see Fig. 1). Each primitive may actually specify the trajectory of other points of the robot in addition to the CoM: for example, a stepping primitive will include also the trajectory of the swing foot. Note that selecting a particular CoM primitive does not specify the whole-body motion, which can be freely chosen[1] by the planner among those compatible with the primitive.

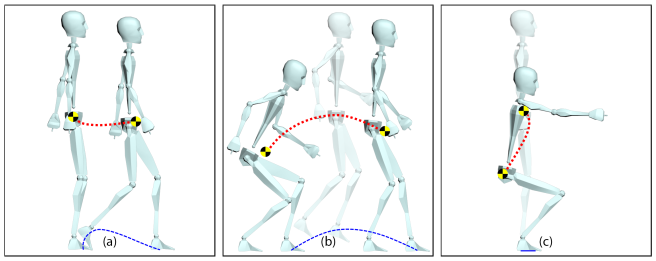

Fig. 1. Examples of CoM movement primitives: (a) stepping (b) jumping (c) squatting. Each primitive specifies a trajectory for the CoM (red, dotted) and possibly other points of the robot, such as a foot (blue, dashed). However, the planner can freely choose the whole-body motion among the infinite that are compatible with the primitive.

Once a CoM movement primitive has been chosen, the displacement of $\boldsymbol{q}_{\mathrm{CoM}}$ is defined throughout its whole duration. At this point, the instantaneous motion of the joint coordinates $\boldsymbol{q}_{\mathrm{jnt}}$ can be chosen so as to realize the chosen primitive, as well as to meet other planning requirements.

The above motion generation approach can be represented by the following conceptual model

$$\boldsymbol{q}_{\mathrm{CoM}}(t) = \boldsymbol{q}^k_{\mathrm{CoM}} + \boldsymbol{A}(\boldsymbol{q}^k_{\mathrm{CoM}})\boldsymbol{u}^k_{\mathrm{CoM}}(t) \tag{1}$$

$$\dot{\boldsymbol{q}}_{\mathrm{jnt}}(t) = \boldsymbol{v}_{\mathrm{jnt}}(t) \tag{2}$$

for $t \in [t_k, t_{k+1}]$, an interval in which the CoM performs a certain primitive movement of duration $T_k = t_{k+1} - t_k$. Here:

- $\boldsymbol{q}^k_{\mathrm{CoM}} = \boldsymbol{q}_{\mathrm{CoM}}(t_k)$ is the CoM frame pose at t_k;
- $\boldsymbol{A}(\boldsymbol{q}^k_{\mathrm{CoM}})$ is a transformation matrix from the CoM frame at t_k to the world frame, whose structure depends on the choice of orientation coordinates in $\boldsymbol{q}_{\mathrm{CoM}}$;
- $\boldsymbol{u}^k_{\mathrm{CoM}}(t)$ is the pose displacement of the CoM frame at t relative to the pose at t_k;
- $\boldsymbol{v}_{\mathrm{jnt}}$ is the velocity input vector for the humanoid joints.

Note the hybrid nature of model (1–2). The first equation is algebraic, because the CoM motion is generated by patching whole subtrajectories extracted from the catalogue of primitives; in fact, the primitive chosen for application at t_k specifies the history of the relative pose displacement $\boldsymbol{u}^k_{\mathrm{CoM}}(t)$ for all $t \in [t_k, t_{k+1}]$. The second equation is differential, as joint variables are changed instantaneously to track the CoM motion and pursue other tasks.

[1]In particular, repetition of the same primitive (e.g., a step) in different parts of the plan will correspond in general to different whole-body motions, depending on the local task history and obstacle placement.

B. Task-Constrained Planning

In this paper, a *task* is described as the trajectory for the position (and possibly orientation) of a specific point (body) of the humanoid. For instance, a manipulation task may be specified as a trajectory assigned to one hand, while a navigation task may be assigned in terms of motion of the midpoint between the feet. This simple viewpoint makes it very easy to translate a task from natural language ('pick up that object and bring it to me') to an assignment that can be directly used by our planner.

Collect the task coordinates in a vector $\boldsymbol{y}$ taking values in an appropriate space. Task coordinates are related to configuration coordinates by a forward kinematic map

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{q}_{\mathrm{CoM}}, \boldsymbol{q}_{\mathrm{jnt}}).$$

Suppose that a desired task trajectory $\boldsymbol{y}^*(t)$, $t \in [t_i, t_f]$, is assigned. The trajectory may be a geometric path (with a path parameter s in place of t) or also degenerate to a single point $\boldsymbol{y}^*(t_f)$ representing a desired task set-point.

In short, the planning problem considered in this paper is to find a feasible whole-body motion of the humanoid over $[t_i, t_f]$ that realizes the assigned task while avoiding collisions with workspace obstacles, whose geometry is known in advance. In our approach, a solution is identified by a concatenation of CoM movement primitives that has been 'fleshed out' by defining collision-free whole-body motions which realize such movements while complying with the task. In the end, however, the solution can be directly described in terms of joint motions.

More explicitly, a solution to our problem consists of a trajectory $\boldsymbol{q}_{\mathrm{jnt}}(t)$, $t \in [t_i, t_f]$ that satisfies three requirements:

R1 The assigned task trajectory is exponentially realized; that is,

$$\lim_{t\to\infty} (\boldsymbol{y}(t) - \boldsymbol{y}^*(t)) = \boldsymbol{0}$$

with an exponential rate of convergence.

R2 Collisions with workspace obstacles and self-collisions are avoided.

R3 Position and velocity limits on the robot joints, respectively in the form $\boldsymbol{q}_{\mathrm{jnt,m}} < \boldsymbol{q}_{\mathrm{jnt}} < \boldsymbol{q}_{\mathrm{jnt,M}}$ and $\boldsymbol{v}_{\mathrm{jnt,m}} < \boldsymbol{v}_{\mathrm{jnt}} < \boldsymbol{v}_{\mathrm{jnt,M}}$, are satisfied.

Note that if $\boldsymbol{y}(t_i) = \boldsymbol{y}^*(t_i)$ (*matched* initial configuration, or 'the robot starts on the task'), requirement R1 automatically becomes

$$\boldsymbol{y}(t) = \boldsymbol{y}^*(t),\ \forall t \in [t_i, t_f],$$

i.e., the assigned task must be exactly realized at all times.

IV. Motion Generation

The proposed planner works in an iterative fashion by repeated calls to a *motion generator*. In particular, we use two interleaved procedures for generating motions of $\boldsymbol{q}_{\mathrm{CoM}}$ and $\boldsymbol{q}_{\mathrm{jnt}}$. The first (*CoM movement selection*) selects a particular CoM movement from the set of primitives. The second procedure (*joint motion generation*) computes feasible (in the sense of requirement R3) collision-free joint motions that realize the chosen primitive as well as the corresponding portion of the assigned task trajectory.

A. CoM Movement Selection

The CoM movement selector is invoked from the current configuration $\boldsymbol{q}^k = (\boldsymbol{q}^k_{\mathrm{CoM}}\ \boldsymbol{q}^k_{\mathrm{jnt}})^T$ at time t_k.

For sake of illustration, we consider a precomputed catalogue of CoM movement primitives that contains only stepping movements (as well as a non-stepping motion, see below). However, the proposed framework can use any set of primitives; indeed, the richer this set, the larger the set of tasks for which we will be able to plan a whole-body motion. For example, *crouching* and *crawling* primitives would allow to achieve tasks that require passing below obstacles.

As explained in Sect. III-A, each primitive specifies the history of the relative pose displacement $\boldsymbol{u}^k_{\mathrm{CoM}}(t)$ for all $t \in [t_k, t_k + T_k]$. In the following, we denote this history by $\boldsymbol{u}^k_{\mathrm{CoM}}$ for compactness. A CoM movement $\boldsymbol{u}^k_{\mathrm{CoM}}$ is then selected by picking one primitive from the catalogue

$$U = \left\{U^{\mathrm{S}}_{\mathrm{CoM}} \cup U^{\mathrm{D}}_{\mathrm{CoM}} \cup \texttt{free\_CoM}\right\} \tag{3}$$

where $U^{\mathrm{S}}_{\mathrm{CoM}}$ and $U^{\mathrm{D}}_{\mathrm{CoM}}$ are subsets of *static* and *dynamic* steps, respectively, and `free_CoM` a *non-stepping* movement.

The stepping motions in $U^{\mathrm{S}}_{\mathrm{CoM}}$ are extracted from a *static* walking gait, in which equilibrium is guaranteed by ensuring that the ground projection of the CoM falls at all times within the humanoid support polygon. This set will typically include a forward step ($\boldsymbol{u}^{\mathrm{SF}}_{\mathrm{CoM}}$), a backward step ($\boldsymbol{u}^{\mathrm{SB}}_{\mathrm{CoM}}$), left ($\boldsymbol{u}^{\mathrm{SL}}_{\mathrm{CoM}}$) and right ($\boldsymbol{u}^{\mathrm{SR}}_{\mathrm{CoM}}$) steps, and possibly others.

$U^{\mathrm{D}}_{\mathrm{CoM}}$ will instead include motions extracted from a *dynamic* walking gait, in which the Zero Moment Point (ZMP) is always contained in the support polygon. This set is more complex, in that it will contain a starting step ($\boldsymbol{u}^{\mathrm{DF,start}}_{\mathrm{CoM}}$), a cruise step ($\boldsymbol{u}^{\mathrm{DF,cruise}}_{\mathrm{CoM}}$) and a stopping step ($\boldsymbol{u}^{\mathrm{DF,stop}}_{\mathrm{CoM}}$) for each direction of motion.

Finally, with `free_CoM` the CoM is completely free to move as long as both feet remain fixed and the robot maintains equilibrium; this is obviously an important primitive for manipulation tasks. Moreover, it is a *stretchable* primitive, in that its duration can be chosen arbitrarily. Its inclusion is particularly important, because it allows to build a sequence of movements whose total duration is $t_f - t_i$ (as specified by the task) using motion primitives whose individual durations are otherwise fixed.

Note the following important points.

- In general, CoM movement primitives specify the motion of other points on the robot body in addition to that of the CoM displacement. For example, stepping primitives assign also the swing foot trajectory within the associated time interval.
- The above stepping movement primitives can be precomputed using suitable Walking Pattern Generators.
- At a given configuration $\boldsymbol{q}^k$, the set of primitives from which to choose is actually a subset of U that depends on the configuration itself, and in particular on which CoM primitive has produced $\boldsymbol{q}^k$. For example, no static step can be selected after a dynamic cruise step; only another cruise step or a stopping step are admissible. Similarly, the only dynamic step that can follow a static step is a starting step; and so on.

To explore the space of possible solutions, the CoM movement may be chosen within the set of primitives U either randomly or based on appropriate heuristics, which may be designed on the basis of the specific task.

Wrapping up, selection of a specific CoM movement primitive provides as outputs:

- a duration T_k for the movement and a time interval $[t_k, t_{k+1}]$;
- for all primitives but `free_CoM`, a reference trajectory $\boldsymbol{z}^*_{\rm CoM}$ for the position of the CoM (the position component of $\boldsymbol{q}_{\rm CoM}$) in $[t_k, t_{k+1}]$, where $t_{k+1} = t_k + T_k$;
- for all primitives, a reference trajectory $\boldsymbol{z}^*_{\rm swg}$ in $[t_k, t_{k+1}]$ for the swing foot (position and orientation).

Note that the swing foot reference trajectory in $[t_k, t_{k+1}]$ for the `free_CoM` primitive is simply $\boldsymbol{z}^*_{\rm swg}(t) = \boldsymbol{z}^*_{\rm swg}(t_k)$.

B. Joint Motion Generation

Once a CoM movement primitive of duration T_k has been chosen, instantaneous joint motion is generated from $\boldsymbol{q}^k$ at t_k with the objective of realizing the portion of the assigned task $\boldsymbol{y}^*$ contained between t_k and $t_{k+1} = t_k + T_k$, as well as $\boldsymbol{z}^*_{\rm CoM}$ and $\boldsymbol{z}^*_{\rm swg}$ in the same interval.

Define the augmented task vector as $\boldsymbol{y}_a = (\boldsymbol{y}^T \;\; \boldsymbol{z}^T_{\rm swg})^T$ if `free_CoM` has been chosen as CoM movement primitive, and as $\boldsymbol{y}_a = (\boldsymbol{y}^T \;\; \boldsymbol{z}^T_{\rm CoM} \;\; \boldsymbol{z}^T_{\rm swg})^T$ in any other case. Denote by $\boldsymbol{J}_a$ the Jacobian matrix of $\boldsymbol{y}_a$ with respect to $\boldsymbol{q}_{\rm jnt}$, and by $\boldsymbol{e} = \boldsymbol{y}^*_a - \boldsymbol{y}_a$ the augmented task error, where $\boldsymbol{y}^*_a(t)$ is the reference value of the augmented task in $[t_k, t_{k+1}]$.

Joint velocity commands are generated as

$$\boldsymbol{v}_{\rm jnt} = \boldsymbol{J}^\dagger_a(\boldsymbol{q}_{\rm jnt})\,(\dot{\boldsymbol{y}}^*_a + \boldsymbol{K}\boldsymbol{e}) + (\boldsymbol{I} - \boldsymbol{J}^\dagger_a(\boldsymbol{q}_{\rm jnt})\boldsymbol{J}_a(\boldsymbol{q}_{\rm jnt}))\boldsymbol{w}, \quad (4)$$

where $\boldsymbol{J}^\dagger_a$ is the pseudoinverse of $\boldsymbol{J}_a$, $\boldsymbol{K}$ is a positive definite gain matrix and $\boldsymbol{w}$ is an n-vector that may be chosen arbitrarily without perturbing the execution of the augmented task. Substitution of (4) in (2) yields $\dot{\boldsymbol{e}} = -\boldsymbol{K}\,\boldsymbol{e}$, i.e., exponential convergence of the augmented task to its reference trajectory.

Given the kinematic redundancy of the humanoid with respect to the augmented task, further exploration of the space of possible solutions is achieved by letting

$$\boldsymbol{w} = \boldsymbol{w}_{\rm rnd}, \quad (5)$$

where $\boldsymbol{w}_{\rm rnd}$ is a bounded-norm random n-vector. In the case in which `free_CoM` has been selected as primitive movement, we use a slightly different choice of $\boldsymbol{w}$:

$$\boldsymbol{w} = -\eta \cdot \nabla_{\boldsymbol{q}_{\rm jnt}} H(\boldsymbol{q}_{\rm jnt}) + \boldsymbol{w}_{\rm rnd}, \quad \eta > 0, \quad (6)$$

where $H(\boldsymbol{q}_{\rm jnt})$ is the squared distance between the centroid of the support polygon and the ground projection of the CoM. The first term in eq. (6) tends to move the CoM towards the center of the support polygon, in order to privilege the generation of robot configurations that are statically stable.

The trajectories generated by (4–5), or (4–6), are continuously checked for collisions (requirement R2) as well as for position and velocity joint limits (requirement R3). For the `free_CoM` primitive, static equilibrium is also explicitly checked. If any of these conditions is violated, the current execution of the motion generator is interrupted. Otherwise, integration proceeds up to t_{k+1}. In this case, we have obtained a feasible, collision-free whole-body motion over that complies with the task over $[t_k, t_{k+1}]$ and can be used by the planner to compose a solution.

V. Planner Overview

Our planner builds a tree in configuration space with the root at the initial configuration $\boldsymbol{q}(t_i)$. Nodes are configurations of the humanoid associated to a time instant, while arcs represent feasible, collision-free whole-body motions that realize a portion of the task. As explained in the previous section, each of these motions has been computed using a CoM movement primitive as a ‘seed’.

The algorithm makes use of a task compatibility metric in configuration space. In particular, given a certain point $\bar{\boldsymbol{y}}$ of the task trajectory, function $\gamma(\boldsymbol{q}, \bar{\boldsymbol{y}})$ characterizes the compatibility of the robot configuration $\boldsymbol{q}$ with the task point $\bar{\boldsymbol{y}}$. For example, in a manipulation task, $\gamma(\boldsymbol{q}, \bar{\boldsymbol{y}})$ can be defined as the inverse of the Euclidean distance between the ground projection of $\bar{\boldsymbol{y}}$ and the midpoint between the feet when the robot is in $\boldsymbol{q}$; the rationale being that motion generation from configurations where this distance is large is more prone to failure, because some joints will be close to the limits of their available ranges. This kind of compatibility function is also appropriate for a navigation task, and will therefore be chosen for the planning experiments of Section VI.

The pseudocode of the planner is shown in Fig. 2. The generic iteration of the planner starts by choosing a random sample $\boldsymbol{y}^*_{\rm rand}$ from the assigned task trajectory $\boldsymbol{y}^*$. Then, a node $\boldsymbol{q}_{\rm near}$ is randomly extracted from the tree using a probability that is proportional to the task compatibility $\gamma(\cdot, \boldsymbol{y}^*_{\rm rand})$. Once $\boldsymbol{q}_{\rm near}$ has been identified with its associated time instant, the motion generator is called. First, a CoM movement is selected from the currently available subset of primitives; as explained before, this subset depends on $\boldsymbol{q}_{\rm near}$. The new CoM movement comes with a duration as well as reference trajectories for the CoM and the swing foot in the next time interval. The portion of the assigned task contained in the time interval can then be extracted. Finally, the joint motion generator computes a whole-body motion that complies with the task. If this trajectory is feasible and collision-free, its final configuration $\boldsymbol{q}_{\rm new}$ is added to the tree; otherwise, a new iteration is started.

VI. Planning Experiments

The proposed planner has been implemented in V-REP for NAO, a small humanoid by Aldebaran Robotics, and runs on an Intel Core 2 Quad at 2.66 GHz.

The set of CoM movement primitives is defined as in (3). Static steps in $U^{\rm S}_{\rm CoM}$ have been precomputed by a Static Walking Pattern Generator using step lengths in the range $[0.03, 0.12]$ m for forward/backward steps and $[0.01, 0.03]$ m for lateral steps; static steps of different heights (in the range $[0.02, 0.06]$ m) were also included to give the robot the possibility of passing over low obstacles. All static steps

```
Algorithm 1: Planner
1  root the tree $\mathcal{T}$ at $q(t_i)$;
2  repeat
3      $i \leftarrow i + 1$;
4      select a random sample $y^*_{\rm rand}$ on the task trajectory;
5      select a random node $q_{\rm near}$ from $\mathcal{T}$ with probability
       proportional to $\gamma(\cdot, y^*_{\rm rand})$;
6      get the time instant $t_k$ associated with $q_{\rm near}$;
7      $[q_{\rm new}, \overline{q_{\rm near}q_{\rm new}}, t_{k+1}]$←GenerateMotion($q_{\rm near}, t_k$);
8      if $q_{\rm new} \neq \emptyset$ then
9          add node $q_{\rm new}$ and arc $\overline{q_{\rm near}q_{\rm new}}$ to $\mathcal{T}$;
10     end
11 until $t_{k+1} = t_f$ or $i =$ MAX_IT ;
```

```
Procedure GenerateMotion ($q_{\rm near}, t_k$)
1  select a random CoM primitive $u^k_{\rm CoM}$ from the
   currently available subset of $U$ given by (3);
2  get the associated duration $T_k$, CoM trajectory $z^*_{\rm CoM}$
   and swing foot trajectory $z^*_{\rm swg}$;
3  extract the portion of task trajectory $y^*$ in $[t_k, t_k + T_k]$;
4  build the extended task $y_a = (y^T \;\; z^T_{\rm CoM} \;\; z^T_{\rm swg})^T$ ;
5  repeat
6      generate motion by integrating joint velocities (4);
7      if collision or joint position/velocity limit violation
       then
8          return $[\emptyset, \emptyset, \emptyset]$
9      end
10 until $t = t_k + T_k$ ;
11 return $[q_{\rm new}, \overline{q_{\rm near}q_{\rm new}}, t_k + T_k]$
```

Fig. 2. Pseudocode of the proposed planner

have a duration of around 2 s. Dynamic steps have been precomputed by a ZMP-based Walking Pattern Generator; in particular, $U^{\rm D}_{\rm CoM}$ includes a starting step of length 0.038 m and duration 1.6 s, a cruise step of length 0.04 m and duration 0.425 s, and a stopping step of length 0.038 m and duration 1.325 s, all in the forward direction. In all dynamic steps, the maximum height for the swing foot is 0.02 m.

CoM movement primitives are selected from U with uniform probability distribution. As for joint motion generation (4-5-6), we use $K = \mathrm{diag}\{2, 2, 1\}$ while $w_{\rm rnd}$ is chosen using uniform probability and a norm limit at 0.4 rad/sec.

We consider two planning scenarios. In the first experiment (Fig. 3), the humanoid must pick up an object (a ball), which is placed on a low stool away from the robot's initial workspace, and then reach a final position in a corridor across an automatic door. This is simply translated to a composite task consisting of two goal points to be reached in sequence by the robot: the first is a position for the right hand in order to pick the ball (manipulation task), while the second is a position for the midpoint between the feet (navigation task). The navigation task is automatically activated when the manipulation task is completed. In view of the nature of our composite task, the task compatibility function γ is defined as described in Section V.

In order to avoid unnatural movements, we chose not to constrain the hand motion during the early stages of the manipulation phase. This is obtained by deleting the task component y from the augmented task vector y_a in eq. (4) when the hand is outside a certain ball centered at the object to be grasped. Note that, even in this condition, expansion of the tree towards the manipulation goal point is still guaranteed in view of the metric used by our planner to select $q_{\rm near}$. Once the hand enters the ball, the task is activated; as a consequence, the robot performs a natural reaching motion only when is sufficiently close to the object.

Figure 3 shows a few snapshots from a solution computed by our planner. The accompanying video contains a dynamic playback clip of the planned motion in which full physical simulation (including joint control) is enabled: this means that the joint motions in the computed plan are feasible and can be effectively tracked by the NAO low-level joint controllers. Note how the planner has correctly generated an approach phase that the humanoid must execute before the actual manipulation can occur. Also, the solution includes different kind of CoM motion primitives: a dynamic walk is used in the approach phase, while the `free_CoM` primitive is used for grasping the object in double support. Once this part of the task is completed, the navigation task is activated and the robot resumes dynamic walking in order to cross the automatic door and reach its final destination. This sensible solution was automatically produced by our planner by taking advantage of the catalogue of movements represented by the set of primitives.

The scenario of the second experiment, shown in Fig. 4, considers a navigation task. In particular, the humanoid is assigned a goal position for its midpoint between the feet. To reach it, the robot has to go through an automatic door whose guide rail represents a ground obstacle. As shown by the snapshots extracted from a solution, and by the dynamic playback clip in the accompanying video, the humanoid initially chooses dynamic walking to reach the door; once there, it performs two static steps of sufficient height to overcome the rail, and finally resumes dynamic walking in order to complete the task. The planner is using exactly the same set of primitives of the first scenario: here, the switch from a dynamic to a static gait is triggered by the characteristics of the environment (the presence of an obstacle that cannot be avoided using a dynamic step), whereas in the first scenario it was a consequence of the composite nature of the task (the robot stopped to pick up the bal). Once again, the planner takes full advantage of the richness of the set of CoM movement primitives.

Table I collects some data (averaged over 10 runs) related to the planner performance in both experiments. Solutions for the first experiment have a longer duration and this is reflected in a larger exploration tree. On the other hand, the time needed to find a solution is longer for the second scenario. This is due to the relative difficulty of overcoming the door guide rail: during planning, many configurations in that area were rejected due to collisions.

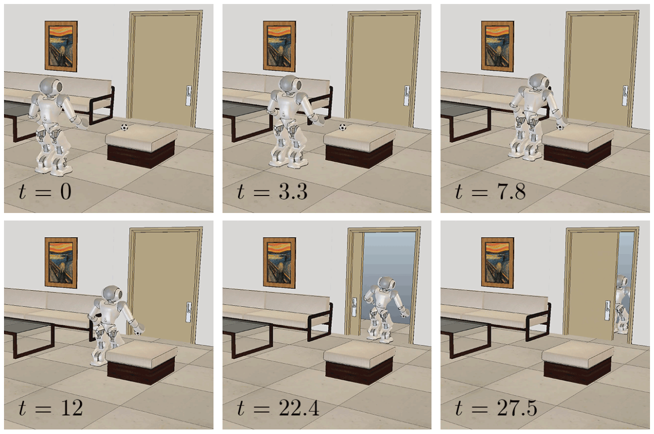

Fig. 3. Planning experiment 1: snapshots from a solution. See the accompanying video.

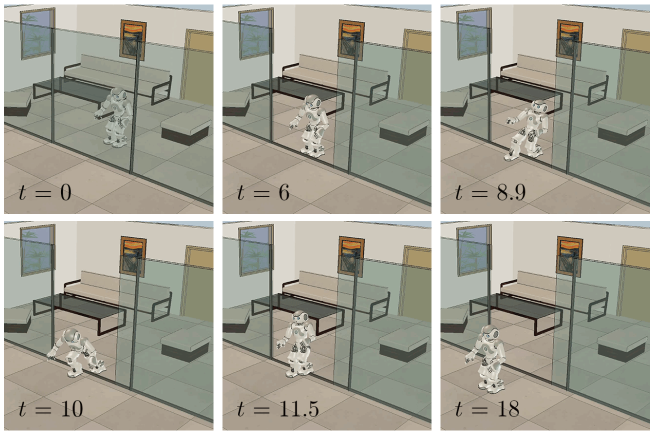

Fig. 4. Planning experiment 2: snapshots from a solution. See the accompanying video.

VII. Conclusions

We considered the problem of planning whole-body motions for a humanoid robot that must execute tasks in an environment containing obstacles. Our approach hinges on the concept of CoM movement primitives, defined as precomputed trajectories of the CoM (and possibly other points of the robot). The proposed planner builds a tree in configuration space by concatenating feasible, collision-free whole-body motions that realize a succession of CoM movements and, at the same time, the assigned task. The algorithm has been successfully implemented in V-REP and validated for the NAO humanoid.

We intend to extend the current framework in order to:

- include additional CoM movement primitives (jumping, crouching, etc) to accomplish more complex tasks;
- design task-based heuristics for guiding the choice of the primitive;
- consider moving obstacles, following the ideas in [18];
- take into account torque bounds, using a second-order motion generation scheme as in [19].

data	exp 1	exp 2
planning time (s)	6.4	8.3
tree size (# nodes)	144.2	72.1
motion duration (s)	27.5	18.0

TABLE I

Planner performance at a glance

References

[1] B. Siciliano and J.-J. Slotine, “A general framework for managing multiple tasks in highly redundant robotic systems,” in *5th Int. Conf. on Advanced Robotics*, 1991, pp. 1211–1216.

[2] O. Kanoun, F. Lamiraux, and P. B. Wieber, “Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task,” *IEEE Trans. on Robotics*, vol. 27, no. 4, pp. 785–792, 2011.

[3] O. Kanoun, J.-P. Laumond, and E. Yoshida, “Planning foot placements for a humanoid robot: A problem of inverse kinematics,” *Int. J. of Robotics Research*, vol. 30, no. 4, pp. 476–485, 2011.

[4] M. Cognetti, P. Mohammadi, G. Oriolo, and M. Vendittelli, “Task-oriented whole-body planning for humanoids based on hybrid motion generation,” in *2014 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2014, pp. 4071–4076.

[5] G. Oriolo and M. Vendittelli, “A control-based approach to task-constrained motion planning,” in *2009 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2009, pp. 297–302.

[6] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, “Motion planning for humanoid robots,” in *11th Int. Symp. of Robotics Research*, 2003.

[7] J. Chestnutt, M. Lau, G. Cheung, J. Kuffner, J. Hodgins, and T. Kanade, “Footstep planning for the honda ASIMO humanoid,” in *2005 IEEE Int. Conf. on Robotics and Automation*, 2005, pp. 629–634.

[8] N. Perrin, O. Stasse, L. Baudouin, F. Lamiraux, and E. Yoshida, “Fast humanoid robot collision-free footstep planning using swept volume approximations,” *IEEE Trans. on Robotics*, vol. 28, no. 2, pp. 427–439, 2012.

[9] J. Pettré, J.-P. Laumond, and T. Siméon, “A 2-stages locomotion planner for digital actors,” in *2003 ACM SIGGRAPH/Eurographics Sym. on Computer Animation*, 2003, pp. 258–264.

[10] E. Yoshida, I. Belousov, C. Esteves, and J.-P. Laumond, “Humanoid motion planning for dynamic tasks,” in *5th IEEE-RAS Int. Conf. on Humanoid Robots*, 2005, pp. 1–6.

[11] J. Kuffner, S. Kagami, K. Nishiwaki, M. Inaba, and H. Inoue, “Dynamically-stable motion planning for humanoid robots,” *Autonomous Robots*, vol. 12, no. 1, pp. 105–118, 2002.

[12] K. Harada, M. Morisawa, K. Miura, S. Nakaoka, K. Fujiwara, K. Kaneko, and S. Kajita, “Kinodynamic gait planning for full-body humanoid robots,” in *2008 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2008, pp. 1544–1550.

[13] F. Burget, A. Hornung, and M. Bennewitz, “Whole-body motion planning for manipulation of articulated objects,” in *2013 IEEE Int. Conf. on Robotics and Automation*, 2013, pp. 1656–1662.

[14] K. Hauser and V. Ng-Thow-Hing, “Randomized multi-modal motion planning for a humanoid robot manipulation task,” *Int. J. of Robotics Research*, vol. 30, no. 6, pp. 678–698, 2011.

[15] S. Dalibard, A. El Khoury, F. Lamiraux, A. Nakhaei, M. Taïx, and J.-P. Laumond, “Dynamic walking and whole-body motion planning for humanoid robots: An integrated approach,” *Int. J. of Robotics Research*, vol. 32, no. 9–10, pp. 1089–1103, 2013.

[16] K. Bouyarmane and A. Kheddar, “Humanoid robot locomotion and manipulation step planning,” *Advanced Robotics*, vol. 26, no. 10, pp. 1099–1126, 2012.

[17] K. Hauser, T. Bretl, J.-C. Latombe, K. Harada, and B. Wilcox, “Motion planning for legged robots on varied terrain,” *Int. J. of Robotics Research*, vol. 27, no. 11–12, pp. 1325–1349, 2008.

[18] M. Cefalo, G. Oriolo, and M. Vendittelli, “Task-constrained motion planning with moving obstacles,” in *2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2013, pp. 5758–5763.

[19] M. Cefalo and G. Oriolo, “Dynamically feasible task-constrained motion planning with moving obstacles,” in *2014 IEEE Int. Conf. on Robotics and Automation*, 2014, pp. 2045–2050.