Small-Space Controllability of a Walking Humanoid Robot

Sébastien Dalibard, Antonio El Khoury, Florent Lamiraux, Michel Taïx, Jean-Paul Laumond

Small-Time Controllability of a Walking Humanoid Robot

Sébastien Dalibard* †, Antonio El Khoury* †, Florent Lamiraux* †, Michel Taïx* †, Jean-Paul Laumond* †
* CNRS ; LAAS ; 7 avenue du colonel Roche, F-31077 Toulouse Cedex 4, France
† Université de Toulouse ; UPS, INSA, INP, ISAE ; UT1, UTM, LAAS ; F-31077 Toulouse Cedex 4, France
{sdalibar,aelkhour}@laas.fr

***Abstract*—This paper presents a two-stage motion planner for walking humanoid robots. A first draft path is computed using random motion planning techniques that ensure collision avoidance. In a second step, the draft path is approximated by a whole-body dynamically stable walk trajectory. The contributions of this work are: (i) a formal guarantee, based on small-time controllability criteria, that the first draft path can be approximated by a collision-free dynamically stable trajectory; (ii) an algorithm that uses this theoretical property to find a solution trajectory. We have applied our method on several problems where whole-body planning and walk are needed, and the results have been validated on a real platform: the robot HRP-2.**

I. Introduction

Humanoid robots are highly redundant and yet under-actuated kinematic systems. Their many degrees of freedom (DoFs), inspired by the human body kinematic tree, provide them with great capabilities for manipulation, but require locomotion in order to move globally. The DoFs that define the position and orientation of the whole robot in space are not directly controlled, they derive from the articular DoFs of the legs of the robot. Those latter should be controlled with care to guarantee a robot dynamic stability.

Integrating walking control systems in a whole-body motion planning architecture is not straightforward. Usual ways of taking care of collision avoidance during locomotion include:

- Computing a collision-free planar trajectory for a bounding box of a robot, then using a locomotion gait to follow the box trajectory.
- Guaranteeing collision avoidance only at a footstep level, and planning footstep positions.

Both methods are approximations, either of the geometry of the robot, or of the geometry of its environment. These simplifications are ways of decreasing the computational complexity of planning for a humanoid kinematic tree augmented with footstep positions.

The work presented here tackles the problem of whole-body motion planning, including locomotion. In a first stage, a randomized motion planner computes a collision-free path for a humanoid robot whose feet are sliding on the ground. In a second stage, this draft path is approximated by a dynamically stable walk trajectory. A theoretical result, based on small-time controllability criteria, guarantees that the draft path can be approximated arbitrarily close by a walking trajectory. This means that after the planning phase, we are sure that a collision-free walking trajectory can be computed from the first draft path. We present various examples of use of this algorithm in the experimental section of the paper. The experiments have been conducted on a model of the HRP-2 robot, and validated on a real platform.

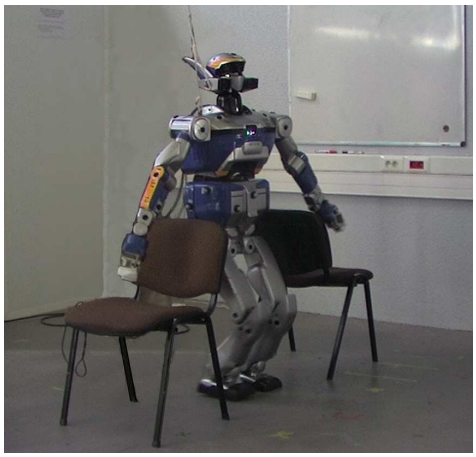

Fig. 1. The robot HRP-2 passing between two chairs. In that kind of environment whole-body collision avoidance is needed during locomotion (edited picture).

II. Related Work and Contribution

This work is based on two different fields of humanoid robotics research: first, randomized whole-body motion planners and second, walk pattern generators based on the Zero-Momentum Point (ZMP) formalism. These two fields rely at a lower level on the prioritized inverse kinematics (IK) formalism [1], [2]. Collision-avoidance can be integrated into an IK-solver, [3] presents such a method for whole-body motion planning, and [4] for collision avoidance at a footstep placement level. However, these methods are prone to fall into local minima. In this paper, we focus on global motion planning.

A. Whole-Body Motion Planning

When planning a whole-body motion for a humanoid robot, the first challenge is to cope with the curse of dimensionality. The complexity of motion planning is exponential in the dimension of the configuration space ($\mathcal{CS}$) to explore [5]. When dealing with high-dimensional configuration spaces, it

is typically impossible to explicitly represent them, leading to the use of randomized sampling techniques to solve global planning problems. In the past fifteen years, *Probabilistic Roadmaps* [6] and *Rapidly exploring Random Trees* (RRT) [7] have been developed and used to solve many high dimensional planning problems. When using sampling techniques on a humanoid robot, the second difficulty is to take into account stability constraints, i.e. to generate random configurations on zero volume submanifolds of $\mathcal{CS}$. This problem has been investigated with success during the last few years, [8], [9] present some solutions. The idea is to use prioritized inverse kinematics techniques within the framework of sampling-based motion planning. To our knowledge, recent contributions to this field do not cover walk planning.

B. Walk pattern generation

Another field of humanoid robotics research is the generation of dynamically stable walk patterns. Since the introduction of the ZMP formalism [10], several methods have been proposed to generate walking motions efficiently. One way to deal with the complexity of a humanoid robotics kinematic tree is to use the so-called "cart-table" simplified model [11]. Based on such a model, planning a trajectory for the ZMP is reduced to planning a trajectory for the Center of Mass (CoM) of the robot. Given a trajectory of the CoM and footstep positions, inverse kinematics solvers can animate the whole set of DoFs of the robot to generate a dynamically stable walk trajectory.

C. Collision-free Walk Trajectories

Collision-free locomotion trajectories is usually obtained by simplifying the model of the robot or its environment. By choosing a bounding volume of a humanoid robot, including its swaying motions, one can use a simple planar motion planner on this bounding volume and generate a valid locomotion trajectory. This strategy is used in [12] in a computer animation context. Variants of this method include dynamic path reshaping [13]: if collisions appear when animating the locomotion trajectory, it is locally reshaped and re-animated. This two-stage strategy does not guarantee that the locomotion trajectory can be followed or that the local reshaping will converge.

Simplifying the environment consists in considering obstacles at a footstep level. [14], [15] use an A* algorithm to find collision-free footsteps. In [16], the authors compute collision-free motions for the legs by using an RRT* algorithm.

D. Contribution

The main contribution of this work is a two-stage motion planner for a humanoid robot that computes a collision-free walking trajectory on the exact models of the robot and its environment. The first stage uses a sampling-based motion planner to compute a collision-free path for a robot sliding on the ground. Another contribution of this paper is the formal proof that this path can be approximated by a dynamically stable, collision-free walking trajectory. The proof relies on small-time controllability properties of humanoid robots. Based on these properties, we propose an algorithm that automatically computes footsteps along the sliding path to animate it. We have implemented this method and used it on a model of HRP-2 robot. The results have been validated on a real platform.

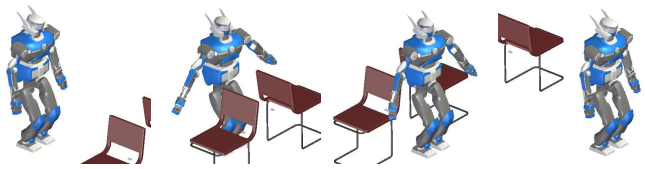

Fig. 2. Collision-free path for a humanoid robot sliding on the ground. The constraints applied to the configurations along this path concern: (i) the relative positions of the feet, (ii) the position of the CoM, (iii) the verticality of the waist.

III. Statically stable collision-free path for a "sliding" robot

The first stage of our planning architecture consists in computing a statically stable path for a humanoid robot sliding on the ground. This path will be approximated in a second stage by a dynamically stable trajectory. The static stability of the configurations of the robot along this path is defined by the following constraints:

1) The two feet are on the ground, and their relative position is fixed,
2) The robot CoM is projected vertically in the center of its support polygon.

To ensure that the approximation of the sliding path by a dynamically stable trajectory can be arbitrarily close, we require the sliding robot to respect the constraints used to validate the cart-table model approximation:

3) The robot CoM is at a constant height,
4) The robot waist is kept vertical.

Random motion planning under task constraints has been successfully investigated in the past years. Here, we apply the method described in [9]. Note that all the constraints applied on the robot at that stage are expressed in the robot frame, so the non-articular DoFs describing the global position and orientation of the robot are free to change.

Fig. 2 shows an example of a collision-free path found by an RRT algorithm. All the configurations along the path respect the set of constraints listed above. The CoM height and the feet positions in the robot frame are chosen such that the generated configurations avoid singularities.

IV. Existence of a dynamically stable trajectory

This section presents a proof that any sliding path p can be approximated by a collision-free walk trajectory. This property is based on ideas from control theory, and in particular small-time controllability. Let us recall briefly what a small-time controllable system is, and how this property is used in motion planning.

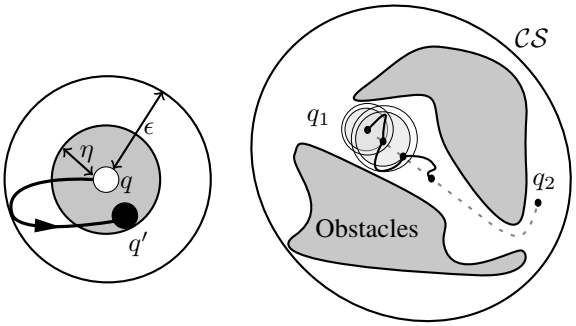

Fig. 3. Small-time controllability in motion planning. On the left, the local property: any configuration q' at distance less than η is reachable from q by an admissible trajectory included in a ball of size ϵ. On the right, a collision-free path from q_1 to q_2 is approximated by collision-free and admissible trajectories by using the local property.

A. Small-Time Controllability

A robotic system is controllable if for any two configurations q_1 and q_2, there exists a time $T > 0$ such that there exists a trajectory going from q_1 to q_2 in time T. It is small-time controllable if for any configuration q, for any time $T > 0$, the set of configurations accessible from q in time less than T is a neighborhood of q in $\mathcal{CS}$. In geometric terms, it means that for any configuration q, for any $\epsilon > 0$, there exist $\eta > 0$ such that all the configurations contained in the ball of center q and radius η are reachable by trajectories included in the ball of center q and radius ϵ. The main consequence of this property in motion planning is that any collision-free path (not necessarily admissible by the system) can be approximated by a sequence of both collision-free and admissible trajectories. This property is crucial in nonholonomic motion planning [17]. Fig. 3 shows an example of collision-free path approximation by admissible collision-free sub-trajectories. The fact that this algorithm converges is guaranteed by the small-time controllability property.

B. Small-Time Controllability of a Walking Humanoid Robot

We want to prove that any collision-free sliding path found by the method presented in section III - for example the one presented in fig. 2 - can be followed by a sequence of collision-free walk motions. Let $\mathcal{M}$ be the $\mathcal{CS}$ submanifold formed by configurations verifying the constraints (1) to (4) presented in section III. It is then sufficient to prove the following result:

Theorem 1. *$\forall q \in \mathcal{M}$, $\forall \epsilon > 0$, $\exists \eta > 0$ such that $\forall q' \in \mathcal{M}$ such that $d(q, q') < \eta$, there exists a dynamically stable walk motion going from q to q' included in the $\mathcal{CS}$-ball of center q and radius ϵ.*

The result is valid under the hypothesis of the cart-table simplified model. We will thus consider that the arms are of negligible mass and do not influence the position of the CoM of the robot. The DoFs of the robot upper-body are therefore free to follow exactly any input trajectory. On the other hand, the DoFs defining the position and orientation of the whole robot in space and the articular DoFs of the legs must generate a valid walk motion and cannot follow any path. The proof will consider first the non-articular DoFs defining the position and orientation of the robot and then the leg DoFs. To position the robot in space, we will consider the position of its CoM.

Following the cart-table model, we require that during the walk motion the CoM stays at a constant height and that the global rotations of the robot around (x) and (y) axes are constant of null angle, so overall, there are three non-articular DoFs of the robot that change along a walk trajectory: x, y and θ, where x and y define its CoM horizontal position, and θ the angle of the rotation of the robot around the (z) axis.

1) Walking in place: First, let us show that it is possible to walk in place while keeping the CoM of the robot in an arbitrarily small neighborhood. The equations giving the ZMP horizontal coordinates (p_x, p_y) as functions of CoM coordinates (x, y) in the cart-table model were presented in [11]:

$$\begin{pmatrix} p_x \\ p_y \end{pmatrix} = \begin{pmatrix} x - \frac{z_c}{g}\ddot{x} \\ y - \frac{z_c}{g}\ddot{y} \end{pmatrix} \quad (1)$$

where z_c is the constant height of the CoM and g is the gravity constant. In the following we will note $\omega_0 = \sqrt{\frac{g}{z_c}}$.

To be able to lift a foot without falling, the robot has to move its ZMP under its other foot. Let us consider a robot in configuration $(0, 0, 0)$. To move the ZMP under a given foot, only the y coordinate of the CoM is of interest. Thus, we will keep the x coordinates of the CoM and ZMP constant equal to 0.

We wish to walk in place while keeping the CoM in an arbitrarily small neighborhood. Let $\epsilon > 0$, arbitrarily chosen, be the size of that neighborhood. We require that for any time $t \geq 0$, $|y(t)| \leq \epsilon$. Let L be the horizontal distance between the CoM and the center of either of the robot feet. L is fixed by the geometry of the robot. We aim at making $p_y(t)$ oscillate between $-L$ and L. During this proof we will assume that the feet of the robot are rectangular, of length l_1 and width l_2. If the feet are not rectangular, we can adapt the proof by considering a rectangle included in the contact surface between a foot and the ground. If no such rectangle exists, for example if the contact is punctual, this proof does not hold.

The idea of this proof is to use the form of Eq. (1) to apply a scaling factor between the amplitude of the oscillations of the CoM and of the ZMP. For example for $\omega > 0$, let us assume the trajectory of the CoM is given by $y(t) = \epsilon \sin(\omega t)$. We can derive Eq. (1) and obtain $p_y(t) = (1 + \left(\frac{\omega}{\omega_0}\right)^2)\epsilon \sin(\omega t)$. The amplitude of the oscillations of y is multiplied by a factor $(1 + \left(\frac{\omega}{\omega_0}\right)^2)$. If we choose $\omega = \omega_0\sqrt{\frac{L}{\epsilon} - 1}$, p_y oscillates between $-L$ and L. At time $t_l^{(n)} = n\frac{2\pi}{\omega} + \frac{\pi/2}{\omega}$, the ZMP is located at the center of the left foot, the robot can lift its right foot and at time $t_r^{(n)} = n\frac{2\pi}{\omega} + \frac{3\pi/2}{\omega}$ the ZMP is located at the center of the right foot, the robot can lift its left foot.

Starting from a static configuration at time $(t = 0)$, we cannot apply directly a command $y(t) = \epsilon \sin(\omega t)$ because it generates a discontinuity in the speed of the CoM at time

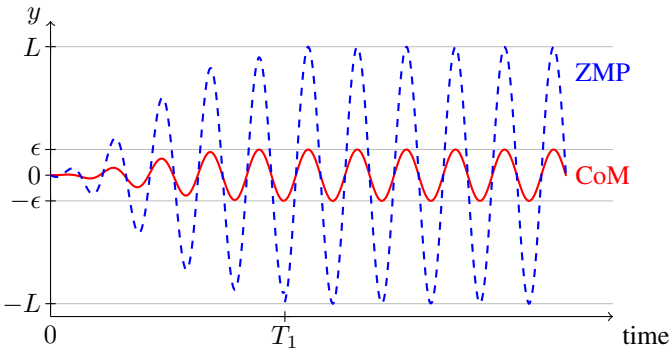

Fig. 4. CoM motion (in plain red) along y axis. The CoM stays in the interval $[-\epsilon, \epsilon]$ while during permanent state ($t \geq T$), the ZMP (dashed blue) oscillates between the centers of the feet, which allows in-place walk.

$(t = 0)$. To overcome this discontinuity, we go through a transient state between $(t = 0)$ and $(t = T)$ for some $T > 0$. Let $f : [0, T] \rightarrow [0, 1]$ be an increasing function of class C^∞ such that $f(0) = 0$, $\dot{f}(0) = 0$, $f(T) = 1$, $\dot{f}(T) = 0$ and $\ddot{f}(T) = 0$. We can explicitly construct such an f with a degree 4 spline. We also request that for all $t \in [0, T]$, $|2\epsilon\dot{f}(t)\frac{\omega}{\omega_0^2}| \leq \frac{l_2}{4}$ and $|\epsilon\ddot{f}(t)/\omega_0^2| \leq \frac{l_2}{4}$. These inequalities will be used to bound the trajectory of the ZMP. We can guarantee them by choosing T large enough. Let us now consider the following CoM motion:

$$y(t) = \begin{cases} f(t)\epsilon\sin(\omega t) & \text{if } t \in [0, T] \\ \epsilon\sin(\omega t) & \text{if } t \geq T \end{cases}$$

One can check that y is of class C^2 over $\mathbb{R}_+$, and that $\dot{f}(0) = 0$. When $t \geq T$, the robot is in the permanent state described above and can walk in place. The last point to check is that for $t \in [0, T]$ $p_y(t)$ stays inside the support polygon of the robot. The calculation of the successive derivatives of y gives:

$$\begin{aligned} p_y(t) = \quad & f(t)\epsilon(1 + \left(\tfrac{\omega}{\omega_0}\right)^2)\sin(\omega t) \\ & +2\epsilon\dot{f}(t)\tfrac{\omega}{\omega_0^2}\cos(\omega t) \\ & +\tfrac{\epsilon}{\omega_0^2}\ddot{f}(t)\sin(\omega t) \end{aligned}$$

For all $t \in [0, T]$, $f(t)\epsilon(1 + \frac{\omega}{\omega_0}^2)\sin(\omega t)$ lies between $-L$ and L. The bounds on the derivatives of f guarantee that $p_y(t)$ lies between $-L - l_2/2$ and $L + l_2/2$, which means that the ZMP stays inside the support polygon. Fig. 4 shows an example of CoM motion on the y axis and the corresponding ZMP motion. Once in permanent walk in-place state, the robot can come back to a static state by applying a symmetric transient state used to decrease gradually the amplitude of the oscillations of the CoM without generating a discontinuity in the first derivative of the command.

These CoM motions can be adapted to follow a (x, y, θ) linear segment by a dynamically stable walk motion, while keeping the CoM at distance at most ϵ from the segment. To do so, we add the walk-in-place CoM motion presented above to the desired trajectory. The details of the proof are shown in Appendix.

2) Legs degrees of freedom: During a walk motion, the six DoFs of each leg are controlled through a position and orientation task on the foot. For a given configuration of the robot waist, this task is defined by six equations determining the position and orientation of the foot. Let q_{waist} be the configuration of the waist of the robot, q_{foot} the desired configuration of the foot, and q_{leg} the articular configuration of the leg. The task on the foot is defined by a function $Tk : \mathbb{R}^6 \times \mathbb{R}^6 \times \mathbb{R}^6 \rightarrow \mathbb{R}^6$, such that the task is satisfied iff $Tk(q_{waist}, q_{foot}, q_{leg}) = 0$. Each component of Tk is a sum of trigonometric functions and as such, Tk is of class C^∞. Let $q^{(0)} \in \mathcal{M}$. By hypothesis, when the robot is in $q^{(0)}$, the tasks defining the positions and orientation of the feet are not in singularity. Hence, $\partial Tk/\partial q_{leg}(q^{(0)}_{waist}, q^{(0)}_{foot}, q^{(0)}_{leg})$ is invertible. The implicit function theorem can be applied and states that: there exist open sets $U \subset \mathbb{R}^{12}$ and $V \subset \mathbb{R}^6$ and $\phi : U \rightarrow V$ such that:

- $(q^{(0)}_{waist}, q^{(0)}_{foot}) \in U$ and $q^{(0)}_{leg} \in V$,
- $\forall (q_{waist}, q_{foot}) \in U, Tk(q_{waist}, q_{foot}, \phi(q_{waist}, q_{foot})) = 0$,
- ϕ is of class C^∞.

The continuity of ϕ implies that for a given ϵ, there exists an open set U' containing $(q^{(0)}_{waist}, q^{(0)}_{foot})$ and included in U such that for all $(q_{waist}, q_{foot}) \in U'$, $|\phi(q_{waist}, q_{foot}) - q^{(0)}_{leg}| < \epsilon$. Let $U'_{waist} \subset \mathbb{R}^6$ and $U'_{foot} \subset \mathbb{R}^6$ be products of open intervals containing respectively $q^{(0)}_{waist}$ and $q^{(0)}_{foot}$, such that $U'_{waist} \times U'_{foot} \subset U'$. For configurations in $\mathcal{M}$, there is a bijection $foot$ between the configuration of the waist and of the foot. Let U'' be an open ball containing $q^{(0)}_{waist}$ included in $U'_{waist} \cap foot^{-1}(U'_{foot})$. Let h_{max} be the maximum height for a foot positions in U'_{foot}. For any configuration $q \in \mathcal{M}$ such that when the robot is in q, its waist is in U'', following the linear segment $[q^{(0)}, q]$ with footsteps of height less than h_{max} will generate leg configurations at distance less than ϵ from $q^{(0)}_{leg}$.

3) Global Proof: We can now conclude the proof. Let $q^{(0)}$ be a configuration in $\mathcal{M}$ and $\epsilon > 0$ arbitrarily chosen. Let U''_l and U''_r be open balls of $\mathbb{R}^6$ as defined above respectively for the left and right foot. Let V be the set of configurations $q \in \mathcal{M}$ such that:

- the non-articular DoF values are in $U''_l \cap U''_r$
- the non-articular DoF values are at distance at most ϵ from $q^{(0)}$,
- the upper-body DoF values of q are at distance at most ϵ from $q^{(0)}$,

V contains a neighborhood of $q^{(0)}$ in $\mathcal{M}$. For any $q \in V$, the CoM and leg motions corresponding to a walk motion from $q^{(0)}$ to q as described in appendix, with a linear interpolation of the upper-body DoFs, generate a whole-body motion at distance at most $k.\epsilon$ from $q^{(0)}$ where k is a constant depending on the number of DoFs of the robot. This concludes the proof.

Remark: The control strategy presented in this proof may generate very long trajectories, because of the transient states at the beginning and end of the locomotion. In the actual implementation, we chose to generate CoM motions with a ZMP preview controller, as presented in [11]. We have observed experimentally that the amplitude of CoM trajectories decreases when the frequency of steps increases.

V. Animation of the statically stable path

The algorithm that animates a statically stable path into a dynamically stable walk trajectory has been inspired by the previous small-time controllability proof. Given a statically stable path p, we start by placing footsteps corresponding to the nominal walk pattern of the robot - in our experimental section, HRP-2. The footstep time parametrization also follows the robot nominal walk. Given the footsteps, we compute a ZMP trajectory, and a preview controller outputs a corresponding CoM trajectory. At this point, we use the global mechanism shown in [18] to solve a prioritized inverse kinematics problem. The stack of tasks applied to the robot is - in decreasing priority order:

1) Position and orientation of the moving foot,
2) Horizontal position of the CoM,
3) Height of the CoM,
4) Verticality of the waist,
5) Upper-body configuration task towards corresponding configuration of p.

Tasks (1) and (2) generate a dynamically stable motion by using the simplified cart-table model and the ZMP criterion. Tasks (3) and (4) ensure that the resulting motion is well described by the cart-table model. Task (5) is used to approximate p as well as possible given the walk parameters.

Because it comes at the lowest priority, task (5) is not necessarily fulfilled in the resulting trajectory. Hence, collisions may appear when animating p, if the resulting trajectory diverges too much from the initial sliding path. If so, it is necessary to approximate more closely p by a walk trajectory. To do so, we use the small-time controllability property of the system shown in the previous section. The way we use this property is inspired by similar results in non-holonomic mobile robot control presented in [19].

If the animated trajectory collides with the environment, we cut the initial path p into two sub-paths, that we try to animate recursively. When the paths to animate are too short for the robot nominal walk parameters, we accelerate the steps, and decrease the maximum height of the moving foot. As shown in previous section, the walk trajectory corresponding to smaller and faster steps converges toward the sliding path. Algorithm 1 shows pseudo-code that takes a sliding path p as input and returns a collision-free walk trajectory.

The `ComputeFootprints()` function uses the geometric length of p to decide if the footsteps have to be accelerated. The `Animate()` function is a call to an inverse kinematic solver able to generate ZMP and CoM trajectories based on the input footsteps. It includes a ZMP preview controller.

Algorithm 1 FindDynamicTrajectory(Path p)

$Footprints \leftarrow$ ComputeFootprints(p)
$StackOfTasks$.initialize()
$StackOfTasks$.addFootprintTask($Footprints$)
$StackOfTasks$.addWaistTask()
$StackOfTasks$.addConfigurationTask(p)
$DynamicTrajectory \leftarrow$ Animate($StackOfTasks$)
if (CheckForCollisions($DynamicTrajectory$) = Colliding) **then**
  $(p_1, p_2) \leftarrow$ CutInHalf(p)
  $DT_1 \leftarrow$ FindDynamicTrajectory(p_1)
  $DT_2 \leftarrow$ FindDynamicTrajectory(p_2)
  return Concatenate(DT_1, DT_2)
else
  return $DynamicTrajectory$
end if

VI. Experiments

The motion planning algorithms presented in this paper have been implemented using KineoWorks™ [20]. The planning times have been measured on an Intel Core 2 Duo 2.13 GHz PC with 2 GB of RAM. Evaluation of the randomized algorithm has been conducted by executing 50 trials on each problem, we present the average results. Videos of all presented examples can be found on the webpage: http://homepages.laas.fr/sdalibar/humanoids11/

A. Passing between two chairs

The environment shown in Fig. 1 and 2 was presented in [21]. The authors solved it by using a bounding box method, leading the robot to walk sideways between the two chairs. Our method generated a locomotion trajectory in which the robot walks forward, which could be required if the robot has to use vision during locomotion for example. The first planning stage required 29.6 s on average. The animation of the sliding path presented in Fig. 2 used 66.5 s of computation time.

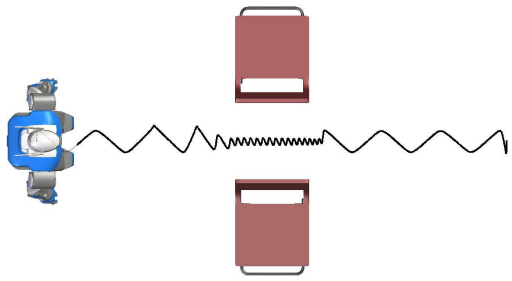

Fig. 5. Horizontal trajectory of the robot waist during locomotion. When the robot is close to obstacles, the amplitude of the oscillations decreases.

Fig. 5 shows the horizontal trajectory of the robot waist (equivalent to its CoM motion) during locomotion. One can see how the amplitude of the oscillations decreases when passing between the chairs. This motion has been validated on a real HRP-2 platform. In order to smooth the motion before

playing it on the real robot, once the collision-free dynamic trajectories had been computed, we saved the footprint positions and footstep parameters and generated a single dynamic walk motion executing all the footsteps. We checked this new motion for collision.

B. Cluttered environment

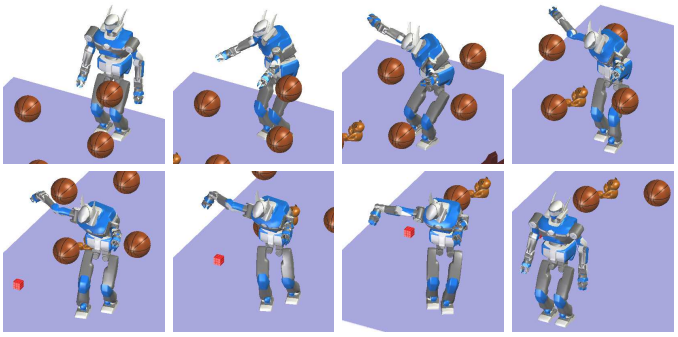

Fig. 6. Solution path for a cluttered environment, the robot walks among floating obstacles.

The environment shown in Fig. 6 was inspired by computer animation benchmarks. The robot has to find a way among many floating obstacles. In this kind of environment neither bounding box nor footstep planning strategies could find a collision-free walk trajectory. The first planning stage required 184.3 s on average, and the animation of the trajectory presented in Fig. 6 used 339.5 s of computation time. Fig. 7 shows the robot waist trajectory during locomotion.

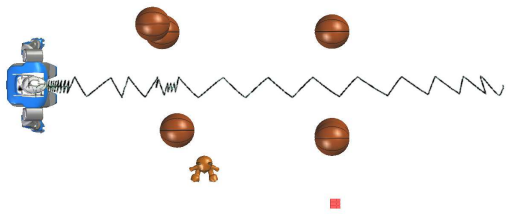

Fig. 7. Horizontal trajectory of the robot waist during locomotion.

C. Grasp Planning

The problem shown in Fig. 8 is defined as a grasping task. The final configuration is defined implicitly by a desired hand position. We generated automatically goal configurations solving the task by following the method proposed in [9]. Then, we applied our planner to generate a whole-body walk motion that solved the grasping task. Computation time for the first planning phase was on average 83.0 s, and the animation of the trajectory presented on Fig. 8 used 90.1 s. Fig. 9 shows the robot waist trajectory during locomotion.

VII. Conclusion and Perspectives

In this paper, we have presented a new planning strategy for humanoid whole-body motion planning including locomotion. The algorithm is based on a formal small-time controllability

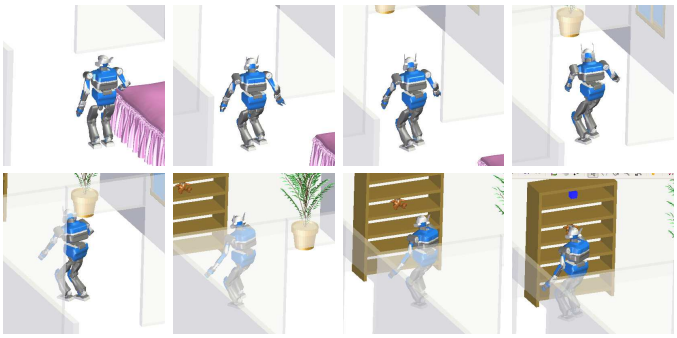

Fig. 8. Solution path for a grasp planning problem in an appartment. The goal is implicitly defined as an inverse kinematics task.

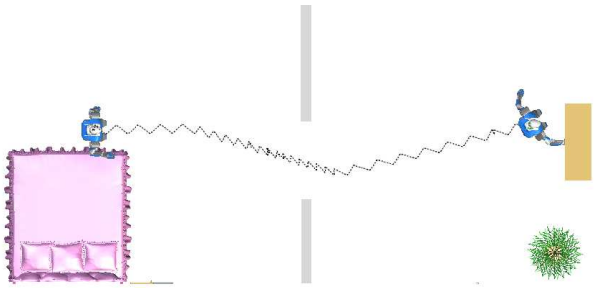

Fig. 9. Horizontal trajectory of the robot waist during locomotion.

property of humanoid robots, that states that they can achieve a dynamically stable walk motion while hardly moving their CoM. We have used our motion planner on different examples, and validated the generated motions on a real robot. Our method has some limitations that should be addressed in future work:

- Because of the kinematic constraints we apply at the planning stage, we are not able yet to plan motions where the robot steps over obstacles, while this is an important feature of humanoid robots,
- The fact that we use the simplified cart-table model forces us to keep the CoM of the robot at a constant height. A more complete model could allow us to plan other types of locomotions, for example to allow the robot to pass under obstacles. We will try to integrate the possibilities presented in [22].

Future experiments will also include manipulation tasks during locomotion. This is possible with the presented method, but not yet tested.

Appendix: Trajectory following with small CoM Motions

The CoM and ZMP motions presented in Sec. IV-B1 can be adapted to walk in-place at a position (x_0, y_0, θ_0). The coordinates of the left foot center are given by: $foot_l(x_0, y_0, \theta_0) = ((x_0 - L\sin(\theta_0), y_0 + L\cos(\theta_0))$ and of the right foot center: $foot_r(x_0, y_0, \theta_0) = (x_0 + L\sin(\theta_0), y_0 - L\cos(\theta_0))$. Let us note $c_0(x_0, y_0, \theta_0) : \mathbb{R}_+ \rightarrow \mathbb{R}^2$ the corresponding CoM motion:

$$c_0(x_0, y_0, \theta_0)(t) = \begin{pmatrix} x_0 - \epsilon \sin(\theta_0) \sin(\omega t) \\ y_0 + \epsilon \cos(\theta_0) \sin(\omega t) \end{pmatrix}$$

This motion stays in a ball of size ϵ around (x_0, y_0) and generates a ZMP motion that reaches successively the centers of the feet. Let us note it $zmp_0(x_0, y_0, \theta_0)$:

$$zmp_0(x_0, y_0, \theta_0)(t) = \begin{pmatrix} x_0 - \epsilon(1 + \left(\frac{\omega}{\omega_0}\right)^2) \sin(\theta_0) \sin(\omega t) \\ y_0 + \epsilon(1 + \left(\frac{\omega}{\omega_0}\right)^2) \cos(\theta_0) \sin(\omega t) \end{pmatrix}$$

Let us now describe the CoM and legs motions starting from a static configuration q_i and reaching a configuration $q_f = (x_f, y_f, \theta_f)$. Without loss of generality, one can assume that $q_i = (0, 0, 0)$. We consider as given the transient state that initializes a walk in-place motion. At time $(t = 0)$, the robot is walking in-place and the CoM and ZMP are moving towards the left foot.

Let $f : [0, T] \to [0, 1]$ be an increasing function of class C^∞ such that $f(0) = 0$, $\dot{f}(0) = 0$, $\ddot{f}(0) = 0$, $f(T) = 1$, $\dot{f}(T) = 0$ and $\ddot{f}(T) = 0$. One can write explicitly such a function by using a degree 5 spline. In order to bound the variations of the ZMP trajectory due to variations of f, we also request the successive derivatives of f to be bounded, and require that for all $t \in [0, T]$:

$$\begin{aligned} \frac{1}{\omega_0^2}(|x_f| + |y_f|)|\dddot{f}(t)| &< \min(l_1/6, l_2/4) \\ \frac{2\epsilon\omega}{\omega_0^2}|\theta_f||\dot{f}(t)| &< l_1/6 \\ \frac{\epsilon}{\omega_0^2}|\theta_f||\ddot{f}(t)| &< l_1/6 \\ \frac{\epsilon}{\omega_0^2}\theta_f^2 \left(\dot{f}(t)\right)^2 &< l_2/4 \end{aligned}$$

Again, these inequalities can be guaranteed by choosing T large enough, i.e. by following the path slowly enough. The $\mathcal{CS}$ trajectory $f_{\mathcal{CS}} : [0, T] \to \mathcal{CS}$ such that for all $t \in [0, T]$, $f_{\mathcal{CS}}(t) = f(t)(q_f - q_i)$ goes from q_i to q_f in time T, while staying on the segment $[q_i, q_f]$. The CoM motion designed to follow this trajectory by walking is:

$$c(t) = c_0(f_{\mathcal{CS}}(t)) = \begin{pmatrix} f(t)x_f - \epsilon \sin(f(t)\theta_f) \sin(\omega t) \\ f(t)y_f + \epsilon \cos(f(t)\theta_f) \sin(\omega t) \end{pmatrix}$$

Note that the initial conditions on f guarantee the continuity of the command derivative and of the ZMP position at time $(t = 0)$. The desired ZMP trajectory is:

$$zmp_{ref}(t) = \begin{pmatrix} f(t)x_f - \epsilon(1 + \left(\frac{\omega}{\omega_0}\right)^2) \sin(f(t)\theta_f) \sin(\omega t) \\ f(t)y_f + \epsilon(1 + \left(\frac{\omega}{\omega_0}\right)^2) \cos(f(t)\theta_f) \sin(\omega t) \end{pmatrix}$$

Following this trajectory, at time $t_l^{(n)}$, the ZMP is located at the center of the left foot, the robot can lift its right foot and move it to position $foot_r(c(t_r^{(n)}))$ and at time $t_r^{(n)}$, the ZMP is located at the center of the right foot, the robot can lift its left foot and move it to position $foot_l(c(t_l^{(n+1)}))$. The real ZMP trajectory zmp_{real} differs from zmp_{ref} because of variations of f. One can compute zmp_{real} by calculating the successive derivatives of c. The error between the desired ZMP position and its real position at time t is given by the equation:

$$\begin{aligned} zmp_{ref}(t) - zmp_{real} = \quad & -\frac{1}{\omega_0^2}\dddot{f}(t) \begin{pmatrix} x_f \\ y_f \end{pmatrix} \\ & + \frac{2\epsilon\omega}{\omega_0^2}\theta_f \dot{f}(t) \cos(\omega t) \begin{pmatrix} \cos(f(t)\theta_f) \\ \sin(f(t)\theta_f) \end{pmatrix} \\ & + \frac{\epsilon}{\omega_0^2}\theta_f \ddot{f}(t) \sin(\omega t) \begin{pmatrix} \cos(f(t)\theta_f) \\ \sin(f(t)\theta_f) \end{pmatrix} \\ & - \frac{\epsilon}{\omega_0^2}\theta_f^2 \dot{f}(t)^2 \sin(\omega t) \begin{pmatrix} \sin(f(t)\theta_f) \\ \cos(f(t)\theta_f) \end{pmatrix} \end{aligned}$$

At time t, the vector $(\cos(f(t)\theta_f), \sin(f(t)\theta_f))$ follows the robot orientation. Using the bounds on the derivatives of f, one can check that the ZMP error along this vector is less than $l_1/2$. In the same way, the error along the direction orthogonal to the robot is less than $l_2/2$. Therefore, for all n, at time $t_l^{(n)}$, the ZMP is under the left foot and at time $t_r^{(n)}$ the ZMP is under the right foot. During the double support phases, the ZMP also stays inside the support polygon.

Overall, we have found a stable walk motion that goes from q_i to q_f while keeping the CoM within ϵ distance of the line segment between q_i and q_f. Once in q_f, a C^1 command can switch to a walk in-place trajectory then to a static configuration.

REFERENCES

[1] B. Siciliano and J. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," in *Advanced Robotics, 1991.'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, 1991, pp. 1211–1216.

[2] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *ASME, Transactions, Journal of Dynamic Systems, Measurement, and Control*, vol. 108, pp. 163–171, 1986.

[3] F. Kanehiro, F. Lamiraux, O. Kanoun, E. Yoshida, and J. Laumond, "A Local Collision Avoidance Method for Non-strictly Convex Polyhedra," in *2008 Robotics: Science and Systems Conference*, 2008.

[4] O. Kanoun, J.-P. Laumond, and E. Yoshida, "Planning Foot Placements for a Humanoid Robot: A Problem of Inverse Kinematics," *The International Journal of Robotics Research*, p. 0278364910371238, 2010. [Online]. Available: http://ijr.sagepub.com/cgi/content/abstract/0278364910371238v1

[5] J. Latombe, *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[6] L. Kavraki, P. Svestka, J. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 4, pp. 566–580, 1996.

[7] J. Kuffner and S. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA'2000), San Francisco, CA*, Apr. 2000. [Online]. Available: citeseer.ist.psu.edu/article/kuffner00rrtconnect.html

[8] D. Berenson, S. S. Srinivasa, and J. Kuffner, "Task space regions: A framework for pose-constrained manipulation planning," *The International Journal of Robotics Research*, 2011. [Online]. Available: http://ijr.sagepub.com/content/early/2011/03/15/0278364910396389.abstract

[9] S. Dalibard, A. Nakhaei, F. Lamiraux, and J.-P. Laumond, "Whole-body task planning for a humanoid robot: a way to integrate collision avoidance," in *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, 7-10 2009, pp. 355 –360.

[10] M. Vukobratovic and D. Juricic, "Contribution to the synthesis of biped gait," *Biomedical Engineering, IEEE Transactions on*, no. 1, pp. 1–6, 1969.

[11] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of zero-moment point," in *IEEE International Conference on Robotics and Automation*, vol. 2. Citeseer, 2003, pp. 1620–1626.

[12] J. Pettré, J. Laumond, and T. Siméon, "A 2-stages locomotion planner for digital actors," in *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2003, p. 264.

[13] E. Yoshida, I. Belousov, C. Esteves, and J.-P. Laumond, "Humanoid motion planning for dynamic tasks," in *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, 5-5 2005, pp. 1 –6.

[14] J. Chestnutt, M. Lau, G. Cheung, J. Kuffner, J. Hodgins, and T. Kanade, "Footstep planning for the Honda ASIMO humanoid," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 629–634.

[15] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, "Motion planning for humanoid robots," *Robotics Research*, pp. 365–374, 2005.

[16] N. Perrin, O. Stasse, F. Lamiraux, and E. Yoshida, "A biped walking pattern generator based on half-steps for dimensionality reduction," *Robotics and Automation, 2011. Proceedings. ICRA'11. IEEE International Conference on*, 2011.

[17] J. Laumond, *Robot motion planning and control*. Springer, 1998.

[18] E. Yoshida, O. Kanoun, C. Esteves, and J. Laumond, "Task-driven support polygon reshaping for humanoids," in *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, 2006, pp. 208–213.

[19] J.-P. Laumond, P. Jacobs, M. Taix, and R. Murray, "A motion planner for nonholonomic mobile robots," *Robotics and Automation, IEEE Transactions on*, vol. 10, no. 5, pp. 577 –593, oct 1994.

[20] J. Laumond, "Kineo CAM: a success story of motion planning algorithms," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 2, pp. 90–93, 2006.

[21] A. El Khoury, M. Taix, and F. Lamiraux, "Path optimization for humanoid walk planning: an efficient approach," *to appear in Int. Conf. Informatics in Control, Automation and Robotics (ICINCO) 2011*, 2011.

[22] F. Kanehiro, H. Hirukawa, K. Kaneko, S. Kajita, K. Fujiwara, K. Harada, and K. Yokoi, "Locomotion planning of humanoid robots to pass through narrow spaces," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 1. IEEE, 2004, pp. 604–609.