



HAL
open science

Manipulation of Documented Objects by a Walking Humanoid Robot

Sébastien Dalibard, Alireza Nakhaei, Florent Lamiroux, Jean-Paul Laumond

► **To cite this version:**

Sébastien Dalibard, Alireza Nakhaei, Florent Lamiroux, Jean-Paul Laumond. Manipulation of Documented Objects by a Walking Humanoid Robot. 10th IEEE-RAS International Conference on Humanoid Robots, Dec 2010, Nashville, United States. pp. 518-523. hal-00518187

HAL Id: hal-00518187

<https://hal.science/hal-00518187>

Submitted on 28 Sep 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Manipulation of Documented Objects by a Walking Humanoid Robot

Sébastien Dalibard, Alireza Nakhaei, Florent Lamiraux and Jean-Paul Laumond

Abstract—This paper deals with manipulation task planning for a humanoid robot while stepping. It introduces the concept of “documented” objects, i.e. objects that provide information on how to manipulate them. The planning phase is decoupled into two parts. First a random motion planner uses the documentation of the object to quickly plan a collision free motion for a simplified model of the robot manipulating the object. Then an inverse kinematics solver animates the whole set of the robot’s degrees of freedom by converting the simplified path into time parametrized tasks. Several examples show the generalization of the method.

I. INTRODUCTION

Humanoid robots are highly redundant and complex systems. As such, they are a very challenging field of motion planning research: while their many degrees of freedom (DoFs) provide them with great capacities for both navigation and manipulation, they make the computational complexity of classic motion planners explode.

Indeed, the complexity of motion planning is strongly related to the dimension of the spaces to explore. Beyond the intrinsic difficulty due to a humanoid robot’s many DoFs, two factors account for a growth of complexity:

- the manipulation of an object by the robot, since the DoFs of that object must also be taken into account,
- the possibility -or necessity- of stepping during task execution. The positions of the footsteps then become new variables that require planning.

However, in many everyday tasks, a lot of planning time can be avoided by the use of a little knowledge about the robot or the environment. When opening a door or picking an object on a table for instance, the way the robot has to execute its manipulation task follows a known pattern, and often the exploration of the configuration space of the system (robot, object) does not need to be exhaustive.

In addition, for simple tasks, the collision avoidance computation can be done at a simplified level, without including all the DoFs of a humanoid robot. It is a classic assumption for navigation planning in graphic animation or robotics ([20], [25]).

This paper proposes a framework for manipulation task planning with steps for humanoid robots based on the simplifying hypotheses that

- the robot is provided with knowledge on how to manipulate objects,

The authors are with CNRS ; LAAS ; 7 Avenue du Colonel Roche, F-31077 Toulouse, France {sdalibar, anakhaei, florent, jpl}@laas.fr

The authors are with Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS ; F-31077 Toulouse, France.

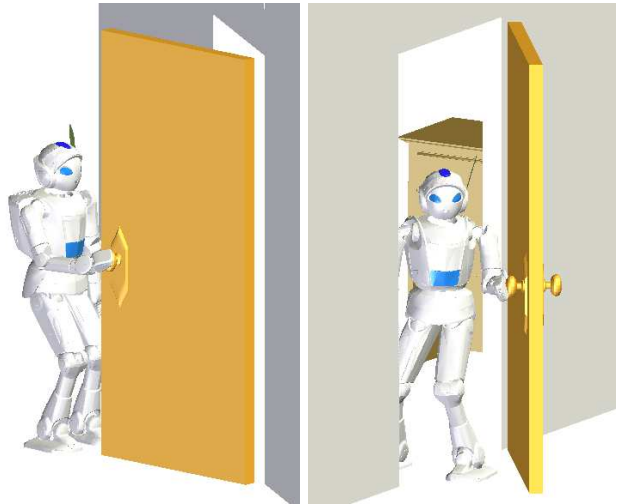


Fig. 1. Humanoid robot manipulating a door with both hands while stepping.

- collision avoidance can be computed on a simplified model of the robot.

II. RELATED WORK AND CONTRIBUTION

Our work is based on two fields of robotics research: first the inverse kinematics approach to humanoid path planning and second randomized motion planning and more specifically its application to manipulation planning.

During the last few years, random motion planning and inverse kinematics techniques have already been coupled successfully on manipulator and humanoid motion planning ([23], [3], [5]). These studies all present generic, probabilistic complete, task motion planners. To our knowledge, they are not suited to plan for steps for a humanoid robot.

A. Whole-Body Task Motion Planning

The classic formulation of inverse kinematics is to compute a joint motion to achieve a desired end-effector pose. When dealing with a humanoid robot, i.e. a highly redundant system, one wants to take advantage of this redundancy by specifying multiple tasks, potentially with different priorities. This problem has been widely studied in robotics planning and control literature, and many jacobian-based solutions have been proposed, for example: [19], [21], [2], [12]. Step generation itself can be part of such a local optimization routine, as presented in [10]. The main limitation of these methods is the difficulty in using them in the presence of obstacles. Local collision avoidance can be seen as a task constraint [8], but is often very costly. [5] presents

a comparison between local collision avoidance and global motion planning for humanoid whole-body task planning. In our work, we will use a jacobian-based framework for whole-body motion generation without dealing with obstacles. The collision avoidance will be guaranteed by a first process of randomized motion planning.

B. Manipulation Planning

In the past few decades, successful works have tried to tackle the general problem of motion planning with randomized algorithms. Following the configuration space paradigm [18], sampling ([11]) and diffusion ([13]) techniques have been proposed, the most famous probably being PRM and RRT. An overview of motion planning methods can be found in [4], [14] and [16].

Powerful as they are, these techniques are not perfectly suited for manipulation planning. A robot manipulating an object is subject to constraints. In terms of configuration space planning, these constraints mean that the path to find lies in a sub-manifold of null volume of the total space. Thus, naive sampling will never generate configurations on that sub-manifold, and will fail to solve any manipulation problem. A lot of work has been done to overcome that issue in a general way, see: [1], [17], [22], [24], [6] for good examples.

In our work, we will not try to tackle the manipulation problem in its general formulation. Instead, we will assume that when manipulating an object, the robot is provided with some information on how to manipulate it. This information can be, for instance, the hands the robot has to use or the position from which the robot should manipulate the object. These instructions will be, formally, projectors in the configuration space of the system (robot,object). We will not focus on the problem of grasp planning, and will assume that our local inverse kinematics solver generates adequate grasping configurations. This work does not deal either with the dynamic door opening problem, as it has been intensively studied in literature, but only geometric motion planning.

C. Contribution

This paper presents a software architecture for manipulation planning based on the notion of “documented” object, i.e. objects that come with a effective user manual for the robot to manipulate them. The planner computes a path for a simplified model of the robot manipulating the object, then animates all the DoFs of the robot by converting this path into time parametrized tasks and passing them to a inverse kinematic solver. This architecture is similar to what was presented in [27], but generalizes it to any manipulation task.

D. Paper Outline

Next sections will detail the different steps of our method, using the example of a robot going through a door. Section III is dedicated to the manipulation planner and section IV to the inverse kinematics solver. Section V presents several examples where our planner is used to manipulate rotating or translating objects.

III. MANIPULATION OF “DOCUMENTED” OBJECTS

This section describes the first component of our task planner. It consists in computing a collision free motion for a system composed by both a simplified model of the humanoid robot and the manipulated object. Depending on the object and the state of the robot, several constraints can be applied to the system. These constraints may vary along the execution of a single task.

For instance to pass through a door without releasing it during its motion, one has to:

- 1) Grab it with one hand,
- 2) Open the door,
- 3) Grab the handle on the other side with the other hand,
- 4) Pass through to the other side while closing the door.

The graph of possible transitions between these constraints is shown on Fig. 2.

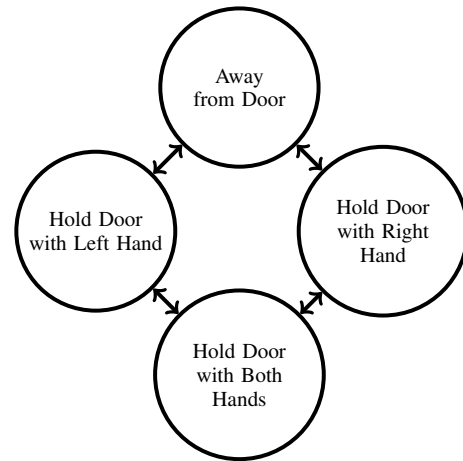


Fig. 2. Graph of successive constraints when opening and closing a door using two hands.

One or a family of sub-manifolds in the system configuration space correspond to each of these constraints. A transition between two constraints is the intersection of such sub-manifolds. A graph such as Fig. 2, as well as the operational expression of the constraints, i.e. the projectors on the corresponding sub-manifolds in the configuration space, are an input of the planner. The information on how to manipulate the object is in sense a user manual for the robot. From now, we will refer to objects coming with such a manual as “documented objects”.

Fig. 3 shows configuration examples corresponding to the different constraints of the door opening problem. In this figure, the simplified model of the robot is a blue box that moves in a 2D plane, with three DoFs (x, y, θ) . The workspaces of its left and right hands are approximated by circles, shown as red dashed lines in the figures. The projector corresponding to the constraint “Hold Door with Right Hand”, for instance (Fig. 3,b), ensures that the outer handle of the door lies within the circle corresponding to the right hand workspace.

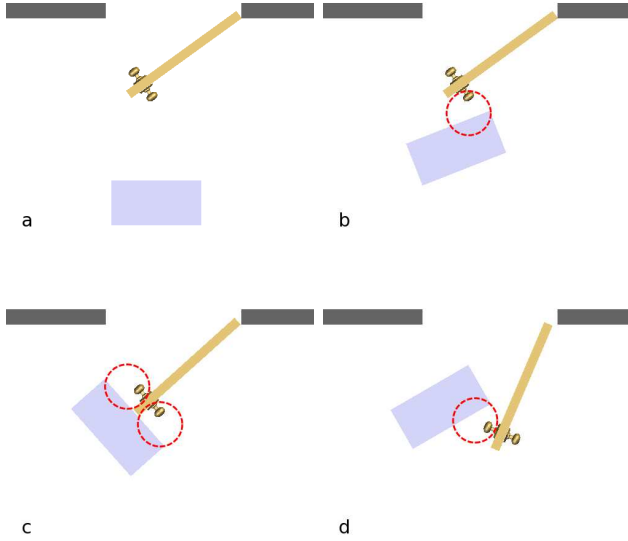


Fig. 3. Example of constraints applied to the system when passing through a door. The simplified model of the robot is a blue box with three DoFs: (x, y, θ) . The workspaces of its left and right hands are approximated by circles and are shown in red dashed lines. Fig. a represents the constraint “Away from Door”, b is “Hold Door with Right Hand”, c is “Hold Door with Both Hands” and d is “Hold Door with Left Hand”.

Given the documentation of an object, we plan a path for the system (robot,object) using randomized motion planning techniques.

A. Adaptation of randomized planning

Randomized motion planning techniques have been widely investigated and used during the last decades. They aim at capturing the topology of a configuration space by sampling collision-free configurations and then connecting them. The family of algorithms on which we have based our planner is Rapidly exploring Random Trees (RRT). Some adjustments had to be made to use that technique in our framework.

A. Representation of the system

The system we plan a path for is composed of a simplified model of the robot and a manipulated object. The state of the system can be described by the DoFs of both and an extra discrete DoF that represents the constraint applied to the system. For instance on Fig. 2 that extra DoF can take four different discrete values, each one corresponding to a node of the graph.

Every time a configuration is created or modified, the projector corresponding to that extra DoF is called and changes the other DoFs of the system accordingly.

B. Interpolation between two configurations

Since a configuration contains the constraint the system must respect, not every connection is possible between configurations, and this for two possible reasons:

- The connection is not authorized by the documentation. Recalling the example of Fig. 2, it is not possible to

directly connect a configuration where the robot does not hold the door to one where it holds it with both hands.

- The manipulated object cannot move if the robot does not hold it. Random sampling will generate configurations of the system with various values for the object DoFs. However, a configuration where the robot does not hold the object can not be directly linked to one with a different set of DoF for the object.

When a direct connection is possible between two configurations with different extra DoFs, the constraint to apply to any interpolated configuration is the weakest of the two constraints at the ends of the direct path. For instance, when linking a configuration where the robot does not hold the object to one where it holds it with its right hand, the linear path connecting them is made of configurations where the robot does not hold the object.

C. Distance

An RRT algorithm is very sensible to the metric used in the configuration space. The distance between two collision free configurations must reflect the likelihood of these configurations seeing each other, i.e. the straight line connecting them is collision free as well. In our case, the distance function we use should also reflect the fact that this straight line exists regarding the constraints. If no direct path exists between two configurations (because of one of the reasons mentioned in the previous paragraph), the distance function returns $+\infty$.

D. Constrained sampling

As said previously, a constraint on the system (robot,object) is formally a sub-manifold of the configuration space. Switching from a constraint to another requires to sample a configuration at the intersection of such sub-manifolds.

Still using the example of Fig. 2, to go from a configuration q_1 where the robot does not hold the door to one q_2 where it holds it with its right hand, the motion planner has to generate a configuration where the door does not move relatively to q_1 and the constraint “Hold the door with Right Hand” is activated. The probability of sampling such a configuration is 0. To overcome this issue, we do not uniformly sample the configuration space. Instead, we specifically sample the intersections of sub-manifolds corresponding to different constraints. The sub-manifolds of interest correspond to the configurations already in the tree. This specific sampling uses the documentation of the manipulated object as well.

To summarize, the documentation of the object consists in:

- 1) a method that takes as input a model of the robot and builds a system (robot,object,constraint)
- 2) the projectors in the system configuration space corresponding to all possible constraints
- 3) the graph of possible transitions between constraints

Given the documentation, a classic motion planner can make the adjustments detailed above. Fig. 4 shows a solution path

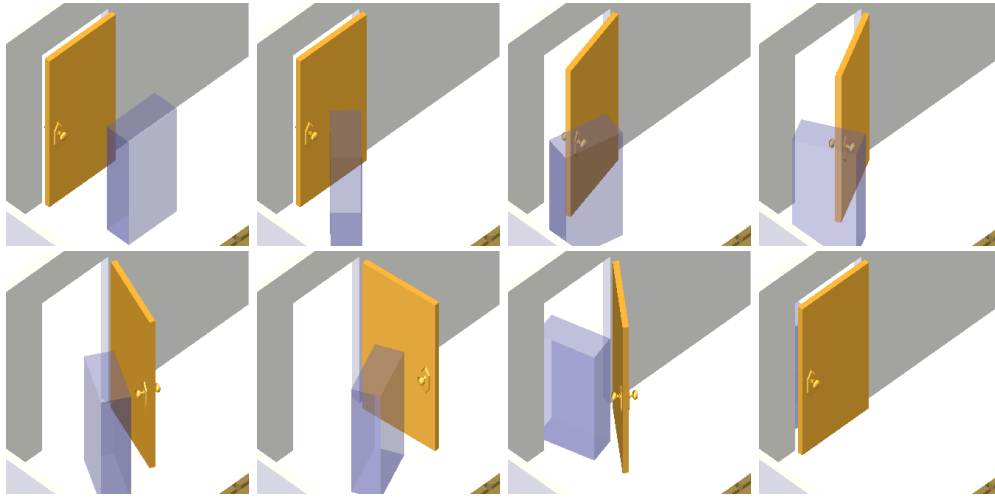


Fig. 4. Solution path for a 2-D box computed by an RRT algorithm to pass through a door. The box comes near the door, attaches its right side to the outer handle, changes sides, attaches its left side to the inner handle, and closes the door.

found for a 2-D box model of the robot passing through a door. This path was planned by an RRT algorithm applied on the system (box,door,constraint).

IV. INVERSE KINEMATICS WITH STEPPING

Once a path has been planned for the simplified model of the robot and the manipulated object, we need to convert it into a whole-body trajectory for the humanoid robot. To do so, we convert the configurations along that kinematic path into time parametrized tasks.

The simplified model of the robot we use is a bounding box around the robot and small spheres that approximate its hands workspaces. The sizes of the box and spheres ensure that no collision will appear once we animate the path with all the joint motions.

A. Generalized Inverse Kinematics Solver

To animate the path found in the previous section, we use a prioritized inverse kinematic solver that merges dynamic walk generation [7] with operational space task completion. It was first presented in [26], a exhaustive description of it can be found in [9].

B. Footsteps and Stability Constraints

The path of the bounding box around the robot is converted into a stack of footprints. The distance between two footprints, as well as the duration of each step depend on the controller parameters and capabilities of the humanoid robot. Dynamically stable walking is ensured by tasks

- 1) on the position of the center of mass of the robot,
- 2) on the position and orientation of the non-support foot.

These tasks are given the highest priority since they involve the equilibrium of the robot and can not be hindered by any other motion.

C. Hand Constraints

For every configuration along the first kinematic path, the set of constraints to apply to the hands of the robot depends on:

- the parameters of the door (axis position, opening angle, handle position on the door)
- the value of the extra DoF indicating the constraint applied to the simplified model of the robot

The time parametrization of these constraints is directly derived from that of the steps.

Once the set of time parametrized constraints is built, we pass it to the generalized inverse kinematic solver, which produces a whole-body, dynamically stable motion. Fig. 5 shows a whole-body path where the robot goes through the door.

As this example is the result of the parts presented in the two last sections, let us recall that the planner produced this motion with the following inputs:

- Initial and final configurations of the system (robot,object) (door closed)
- Documentation of the door

V. EXAMPLES

This section presents experimental results of our planner in simulation using the Toyota Partner Humanoid robot. Our algorithm uses KineoWorksTM([15]) implementation of RRT algorithm and collision checking. All the simulations were performed on a 2.13 GHz Intel Core 2 Duo PC with 2 GB RAM.

The contribution of this paper is the introduction of the formalism of “documented” objects, and its use in humanoid manipulation planning. Therefore, we chose not to focus on computation time evaluation on the examples presented here. Note that the implementation of the inverse kinematics solver runs in real time on the robot, so that part is not an issue as far as computation time is concerned. The manipulation

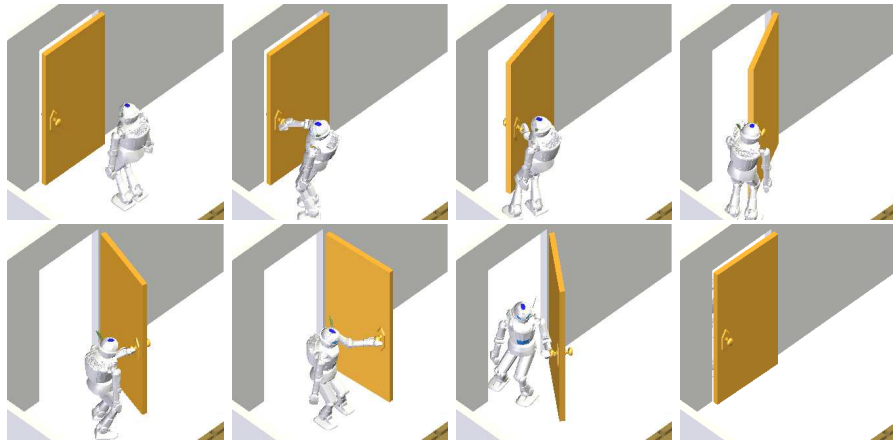


Fig. 5. Whole-body dynamically stable path for a humanoid robot passing through a door.

planning part took from tens of seconds to a few minutes for the problems presented here, depending on the difficulty of each problem.

A. Open and Close a Door

Fig. 5 shows the example of the robot opening and closing a door with both hands, with constraints as specified in section III. The only obstacle avoidance in this first example is between the robot and the door, and the robot and the walls. The two following examples are more sophisticated versions of this problem.

1) *With obstacles in the way of the door:* Fig. 6 shows the same example, but this time a chair is preventing the robot from opening the door wide. The random manipulation planner finds a narrower path, where the robot is very close to the wall and the door. The documentation of the door is the same as in the previous example.



Fig. 6. Opening and closing a door with two hands. A chair stands in the way of the door, so the robot has to go through a narrow passage between the door and the wall.

2) *Using only One Hand:* In this example (Fig. 7), the robot holds an object in its left hand. The documentation used by the robot is different from the previous one, since the robot can only use its right hand.

The new set of constraints is:

- 1) Robot away from door
- 2) Hold inner handle with right hand
- 3) Hold outer handle with right hand

The set of possible transitions is: $(1) \leftrightarrow (2)$, $(1) \leftrightarrow (3)$.

The robot has to open the door, release it, go on the other side and close the door with the same hand. Again, this motion was produced with the same framework, with the new documentation as an input.

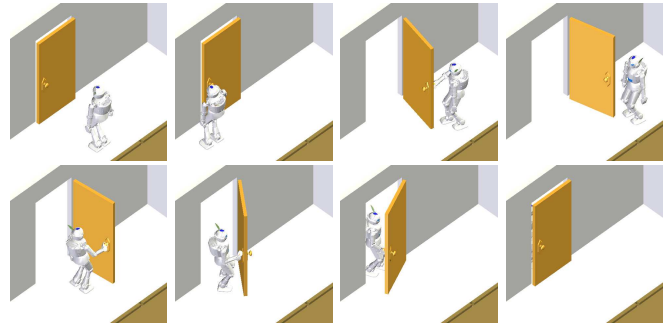


Fig. 7. Opening and closing a door with only one hand. The robot has an object in its left hand.

B. Open and Close a Sliding Door

Fig 8 shows an example similar to the first one, but where the robot has to go through a sliding door. The robot can use both its hands here. The set of constraints is the same as the first door example, except that the DoF of the door is in translation instead of rotation.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a framework for humanoid robot manipulation planning while stepping. The approach tries to break down the computational complexity of the problem by decoupling it into two parts:

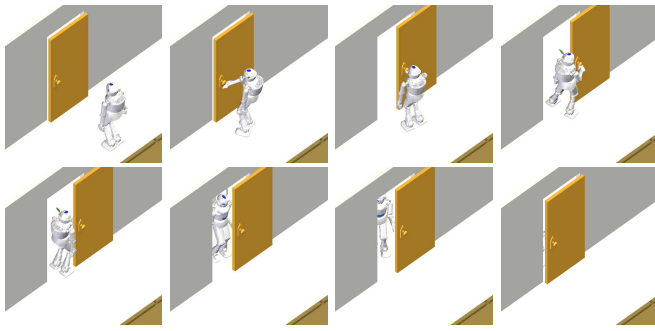


Fig. 8. Opening and closing a sliding door with both hands.

- a simplified manipulation problem, with an input documentation of the object given to the robot,
- whole-body motion generation by an inverse kinematics solver

We have tried that approach on a few examples of manipulation where stepping is needed, and it produced successful and realistic results.

Some future work could be envisaged, first would be to try this approach on other manipulation tasks, for instance on free-flyer objects coming with documentation. To do so, we only have to produce the objects documentation, and use them in our existing framework.

The second and more important point is the fact that for now, we do not check for collisions at the whole-body motion generation step. In our examples, since our representation of the simplified model of the robot was well suited, no collisions appeared, but we do not have a formal guarantee that it will always be the case. It could be worth investigating a hybrid approach between probabilistic complete whole-body planning approaches, such as [3], [5], and our work: if a collision appears when generating the whole-body motion, a local deformation of the path is planned using a probabilistic whole-body motion planner.

VII. ACKNOWLEDGMENTS

This work was partly supported by a LAAS-CNRS and Toyota Motor Europe NV/SA cooperation contract.

REFERENCES

- [1] NM Amato and Y. Wu. A randomized roadmap method for path and manipulation planning. In *1996 IEEE International Conference on Robotics and Automation, 1996. Proceedings.*, volume 1, 1996.
- [2] P. Baerlocher and R. Boulic. Task-priority formulations for the kinematic control of highly redundant articulated structures. In *1998 IEEE/RSJ International Conference on Intelligent Robots and Systems, 1998. Proceedings.*, volume 1, 1998.
- [3] D. Berenson, J. Chestnutt, S.S. Srinivasa, J.J. Kuffner, and S. Kagami. Pose-constrained whole-body planning using task space region chains. In *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, pages 181–187, 7-10 2009.
- [4] H. Choset, K.M. Lynch, S. Hutchinson, G.A. Kantor, W. Burgard, L.E. Kavraki, and S. Thrun. *Principles of Robot Motion: theory, algorithms, and implementation*. MIT Press, 2005.
- [5] S. Dalibard, A. Nakhaei, F. Lamiroux, 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*, pages 355–360, 7-10 2009.

- [6] M. GHARBI, J. CORTES, and T. SIMEON. Roadmap composition for multi-arm systems path planning. In *International Conference on Intelligent Robot and Systems (IROS 2009)*, pages 2471–2476, Saint Louis (Etats-Unis), 2009.
- [7] 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*, volume 2, pages 1620–1626. Citeseer, 2003.
- [8] F. Kanehiro, F. Lamiroux, O. Kanoun, E. Yoshida, and JP Laumond. A Local Collision Avoidance Method for Non-strictly Convex Polyhedra. In *2008 Robotics: Science and Systems Conference*, 2008.
- [9] O. Kanoun. *Contribution à la planification de mouvement pour robots humanoïdes*. PhD thesis, Institut National des Sciences Appliquées, Toulouse, 2009.
- [10] Oussama Kanoun, Jean-Paul Laumond, and Eiichi Yoshida. Planning Foot Placements for a Humanoid Robot: A Problem of Inverse Kinematics. *The International Journal of Robotics Research*, page 0278364910371238, 2010.
- [11] LE Kavraki, P. Svestka, J.C. Latombe, and MH Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *Robotics and Automation, IEEE Transactions on*, 12(4):566–580, 1996.
- [12] O. Khatib, L. Sentis, J. Park, and J. Warren. Whole body dynamic behavior and control of human-like robots. *International Journal of Humanoid Robotics*, 1(1):29–43, 2004.
- [13] 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, April 2000.
- [14] J.C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.
- [15] J.P. Laumond. Kineo CAM: a success story of motion planning algorithms. *Robotics & Automation Magazine, IEEE*, 13(2):90–93, 2006.
- [16] S. M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006. Available at <http://planning.cs.uiuc.edu/>.
- [17] SM LaValle, JH Yakey, and LE Kavraki. A probabilistic roadmap approach for systems with closed kinematicchains. In *1999 IEEE International Conference on Robotics and Automation, 1999. Proceedings*, volume 3, 1999.
- [18] Tomas Lozano-Perez. Spatial planning: A configuration space approach. *IEEE Transactions on Computers*, 32(2):108–120, 1983.
- [19] Y. Nakamura and H. Hanafusa. Inverse kinematic solutions with singularity robustness for robot manipulator control. *ASME, Transactions, Journal of Dynamic Systems, Measurement, and Control*, 108:163–171, 1986.
- [20] J. Pettré, J.P. Laumond, and T. Siméon. A 2-stages locomotion planner for digital actors. In *Proceedings of the 2003 ACM SIG-GRAPH/Eurographics symposium on Computer animation*, page 264. Eurographics Association, 2003.
- [21] B. Siciliano and J.J.E. 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*, pages 1211–1216, 1991.
- [22] T. Simeon, J.P. Laumond, J. Cortes, and A. Sahbani. Manipulation planning with probabilistic roadmaps. *The International Journal of Robotics Research*, 23(7-8):729, 2004.
- [23] M. Stilman. Task constrained motion planning in robot joint space. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pages 3074–3081. Citeseer, 2007.
- [24] Y. Yang and O. Brock. Elastic roadmaps: Globally task-consistent motion for autonomous mobile manipulation in dynamic environments. *Proceedings of Robotics: Science and Systems II*, 2006.
- [25] 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*, pages 1–6, 5-5 2005.
- [26] E. Yoshida, O. Kanoun, C. Esteves, and J.P. Laumond. Task-driven support polygon reshaping for humanoids. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 208–213, 2006.
- [27] E. Yoshida, M. Poirier, J.P. Laumond, O. Kanoun, F. Lamiroux, R. Alami, and K. Yokoi. Pivoting based manipulation by a humanoid robot. *Autonomous Robots*, 28(1):77–88, 2010.