Multi-contacts stances planning for humanoid locomotion and manipulation

Karim Bouyarmane, Abderrahmane Kheddar

Multi-Contacts Stances Planning for Humanoid Locomotion and Manipulation

*Karim BOUYARMANE (CNRS-LIRMM), Abderrahmane KHEDDAR (CNRS-LIRMM)

CNRS-AIST Joint Robotics Laboratory (JRL), UMI3218/CRT, Tsukuba, Japan

Abstract—We present an algorithm that plans a sequence of multi-contacts stances in order to solve the locomotion and manipulation planning problems for humanoid robots and equivalently for dexterous hands. The algorithm uses a Best-First planning approach to explore the stances space and relies on an inverse statics stance solver to generate postures in the configuration space of the system realizing a given stance. Results show that locomotion and manipulation motion planning problems can be solved within the same framework in a unified way.

Key Words: Contact planning, humanoid robots, acyclic gaits, locomotion and manipulation planning.

1. Introduction

Contact-before-motion [1][2] is a motion planning approach that first plans the footprints of a legged system before planning the continuous motion that follows these footprints. By not restricting the possible contacts only to the feet, and thus allowing the humanoid robot to take support on the environment with its hands or any other body part, we can plan complex motions that solve highly constrained situations in unstructured environment [3][4]. This is the main objective of acyclic motion planning. This approach can be extended for solving other classes of motion planning problems beyond the "legged locomotion for a single humanoid robot" problem. These other classes of problems include dexterous manipulation [5][6][7], collaborative manipulation, whole-body manipulation, and legged locomotion for multiple agents [8], among other problems.

After introducing the notations and formulation of the problem (Sect. 2), we present the main algorithm we use to perform this planning which is a Best-Fist search algorithm (Sect. 3) that relies heavily on an independent module we call the inverse stance solver (Sect. 4). Results obtained using this algorithm are demonstrated in Sect. 5.

2. Problem Formulation

We consider a system made of N entities $r \in \{1, \ldots, N\}$. Each entity is a kinematic tree with either fixed root or free-floating root, and with either a single link or multiple links.

	multiple links	single link
free-floating	humanoid robot	manipulated obj.
fixed	dexterous hand	the environment

Each of these entities $r \in \{1, \ldots, N\}$ generates a configuration space $\mathscr{C}_r$ that captures the topology of the corresponding kinematic tree. The configuration space of the whole system is the Cartesian product of these individual configuration spaces

$$\mathscr{C} = \prod_{r=1}^{N} \mathscr{C}_r \, .$$

On each entity $r \in \{1, \ldots, N\}$ we define a set of m_r contact surfaces $s \in \{1, \ldots, m_r\}$. A contact is then a 7-tuple

$$c = (r_1, s_1, r_2, s_2, x, y, \theta) \in \mathbb{N}^4 \times \mathbb{R}^2 \times \mathbb{S}^1 \, ,$$

where

- r_1 is the first entity in contact,
- s_1 is a contact surface defined on the entity r_1,
- r_2 is the second entity in contact,
- s_2 is a contact surface defined on the entity r_2,
- (x, y, θ) is the relative position and orientation of the two contact .

A stance is a set of contacts

$$\sigma \in 2^{\mathbb{N}^4 \times \mathbb{R}^2 \times \mathbb{S}^1}$$

The set of all stances is denotes Σ.

We define a forward kinematic mapping

$$p_\Sigma : \mathscr{C} \to \Sigma$$

that maps every configuration to the set of contacts of the entities when put in that configuration, and the inverse mapping

$$\mathscr{Q}_\sigma = p_\Sigma^{-1}(\{\sigma\})$$

which defines an inverse kinematics submanifold of the configuration space consisting of all the configurations that realize the stance σ. On $\mathscr{Q}_\sigma$ we isolate a subspace

$$\mathscr{F}_\sigma \subset \mathscr{Q}_\sigma$$

in which we only keep the configurations that are physically valid, i.e. configurations that are in static equilibrium, collision-free, and within the joint angles and torques bounds. The search algorithm basically starts from a stance, explores the adjacent stances and then iterates the search on each adjacent stance. Therefore we need a definition of the adjacency relationship. For a stance σ we define the following adjacency sets:

- $\sigma' \in \mathrm{Adj}^+(\sigma)$ if $\sigma' \supset \sigma$ and $\mathrm{card}(\sigma') = \mathrm{card}(\sigma) + 1$.
- $\sigma' \in \mathrm{Adj}^-(\sigma)$ if $\sigma' \subset \sigma$ and $\mathrm{card}(\sigma') = \mathrm{card}(\sigma) - 1$.
- $\mathrm{Adj}(\sigma) = \mathrm{Adj}^+(\sigma) \cup \mathrm{Adj}^-(\sigma)$.

On $\mathrm{Adj}^+(\sigma)$ we define an equivalence relation $\sim_\sigma$ that characterizes only the pair of surfaces in contact without regard to their relative position. Denoting the canonical projection $p_{\mathbb{N}^4} : \mathbb{N}^4 \times \mathbb{R}^2 \times \mathbb{S}^1 \to \mathbb{N}^4$, we define $\sigma_1 \sim_\sigma \sigma_2$ if $\sigma_1 = \sigma \cup \{c_1\}$ and $\sigma_2 = \sigma \cup \{c_2\}$ and $p_{\mathbb{N}^4}(c_1) = p_{\mathbb{N}^4}(c_2)$. Each equivalence class $[\sigma']$ is isomorphic to $\mathbb{R}^2 \times \mathbb{S}^1$. The quotient space is denoted $\mathrm{Adj}^+(\sigma)_{/\sim_\sigma}$.

A feasible sequence of stances is a sequence

$$(\sigma_1, \ldots, \sigma_n) \in \Sigma^n$$

such that $\forall i \in \{1, \ldots, n-1\}$ $\sigma_{i+1} \in \mathrm{Adj}(\sigma_i)$ and $\mathscr{F}_{\sigma_i} \cap \mathscr{F}_{\sigma_{i+1}} \neq \varnothing$. We can now formulate our problem: Given $\sigma_{\text{start}}, \sigma_{\text{goal}} \in \Sigma$ find a feasible sequence of stances $(\sigma_1, \ldots, \sigma_n)$ such that $\sigma_1 = \sigma_{\text{start}}$ and $\sigma_n = \sigma_{\text{goal}}$.

3. Planning Algorithm

The algorithm used to explore Σ is a Best-First search algorithm [9][10], depicted in Alg. 1.

Algorithm 1 FIND_SEQ._OF_STANCES($\sigma_{\text{start}}, \sigma_{\text{goal}}$)

1: initialize priority queue $Q \leftarrow \{\sigma_{\text{start}}\}$
2: **repeat**
3: pop best stance σ from Q
4: **for all** $[\sigma'] \in \mathrm{Adj}^+(\sigma)_{/\sim_\sigma}$ **do**
5: call the inverse stance solver to find q in $\mathscr{F}_\sigma \cap \mathscr{F}_{[\sigma']}$
6: upon success push $\sigma' = p_\Sigma(q)$ into Q
7: **end for**
8: **for all** $\sigma' \in \mathrm{Adj}^-(\sigma)$ **do**
9: call the inverse stance solver to find q in $\mathscr{F}_\sigma \cap \mathscr{F}_{\sigma'}$
10: upon success push σ' into Q
11: **end for**
12: **until** σ is close enough to σ_{goal}

4. Inverse Stance Solver

Lines 5 and 9 of Alg. 1 make a call to an external module called the inverse stance solver. This module is an inverse kinematics solver that finds configurations which satisfy the geometric contact constraints for a given stance, in addition to physical constraints such as the static equilibrium constraints, collision avoidance, and joint angles and torques limits. To achieve this we write a non-linear constrained optimization problem [11][12] in the geometric variables (the configuration q) and the statics variables (the contact forces λ).

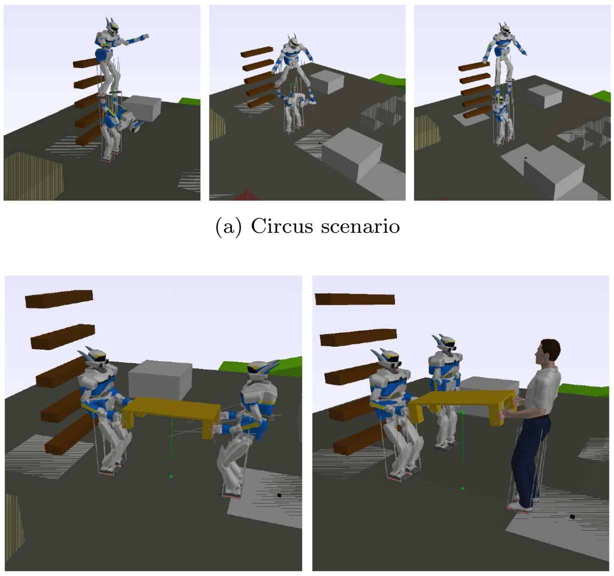
$$\min_{\lambda, q \in \mathscr{F}_\sigma} \mathrm{obj}(q, \lambda)\,,$$

where the objective function is used either to track a goal configuration q_{goal} corresponding to the goal stance σ_{gooal} or used to track a guide path given by a collision-free path planner on a simplified reduced model of the configuration space [13].

When adding a contact to a stance, the position of the contact (x, y, θ) can also be included as an optimization variable and left to be decided by the optimizer,

$$\min_{(x,y,\theta), \lambda, q \in \mathscr{F}_\sigma} \mathrm{obj}(q, \lambda, x, y, \theta)\,.$$

Examples of solutions given by the inverse stance solver are depicted in Fig. 1.



(a) Circus scenario

(b) Collaborative manipulation scenario

Fig.1: Examples of inverse stances solving problems for complex multiple agents systems.

5. Results

Figs. 2, 3 and 4 shows results obtained by applying Alg. 1 to some examples of stances planning for three classes of problems solved by the planner.

6. Conclusion

We have solved the stances planning problem for multi-contacts based acyclic motions of different types of robotics systems. The same framework allows for solving both manipulation and locomotion problems within one framework and using the exact same algorithm for different models of articulated robots, with

Fig.2: Locomotion over irregular terrain using Coulomb friction.

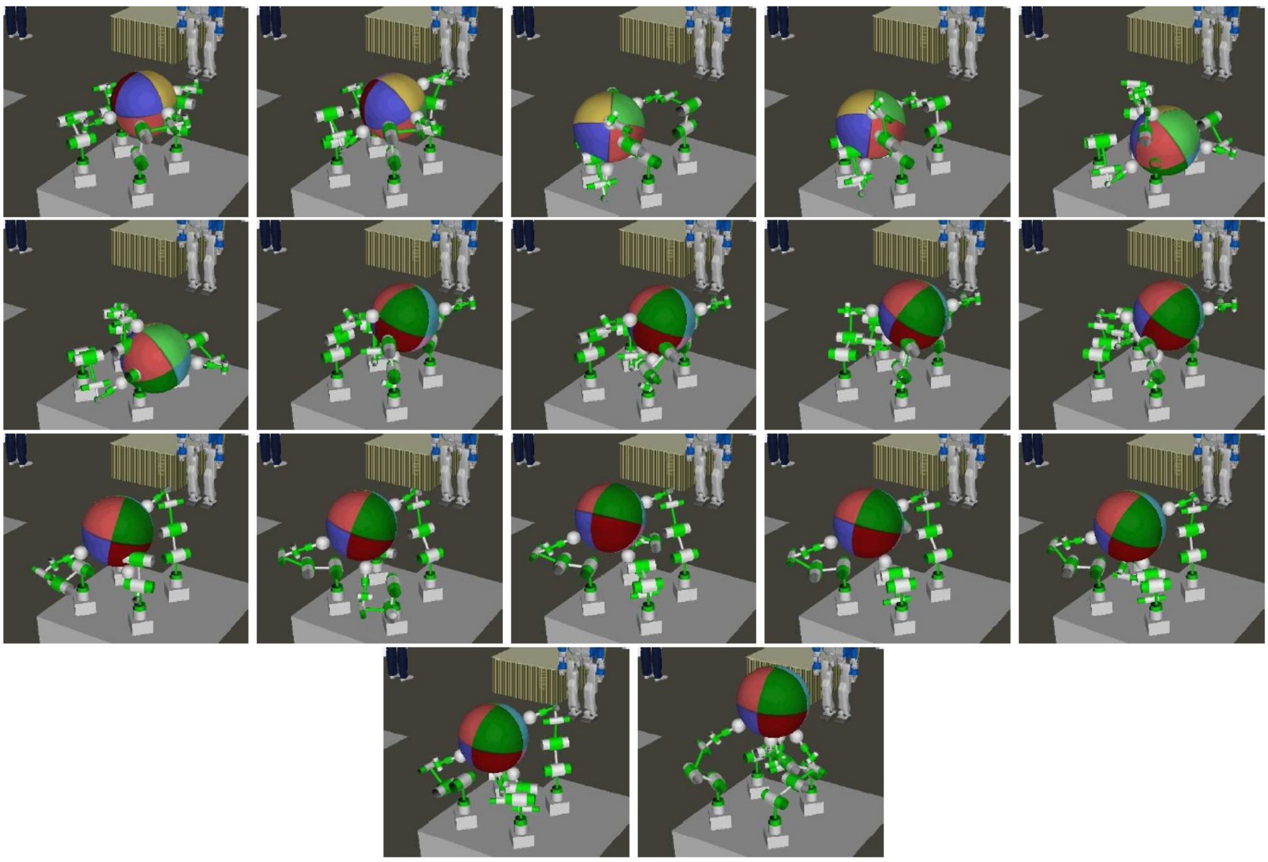


Fig.3: Dexterous manipulation in which the objective is to rotate the ball upside-down using only frictional contacts between the fingertips and the ball.

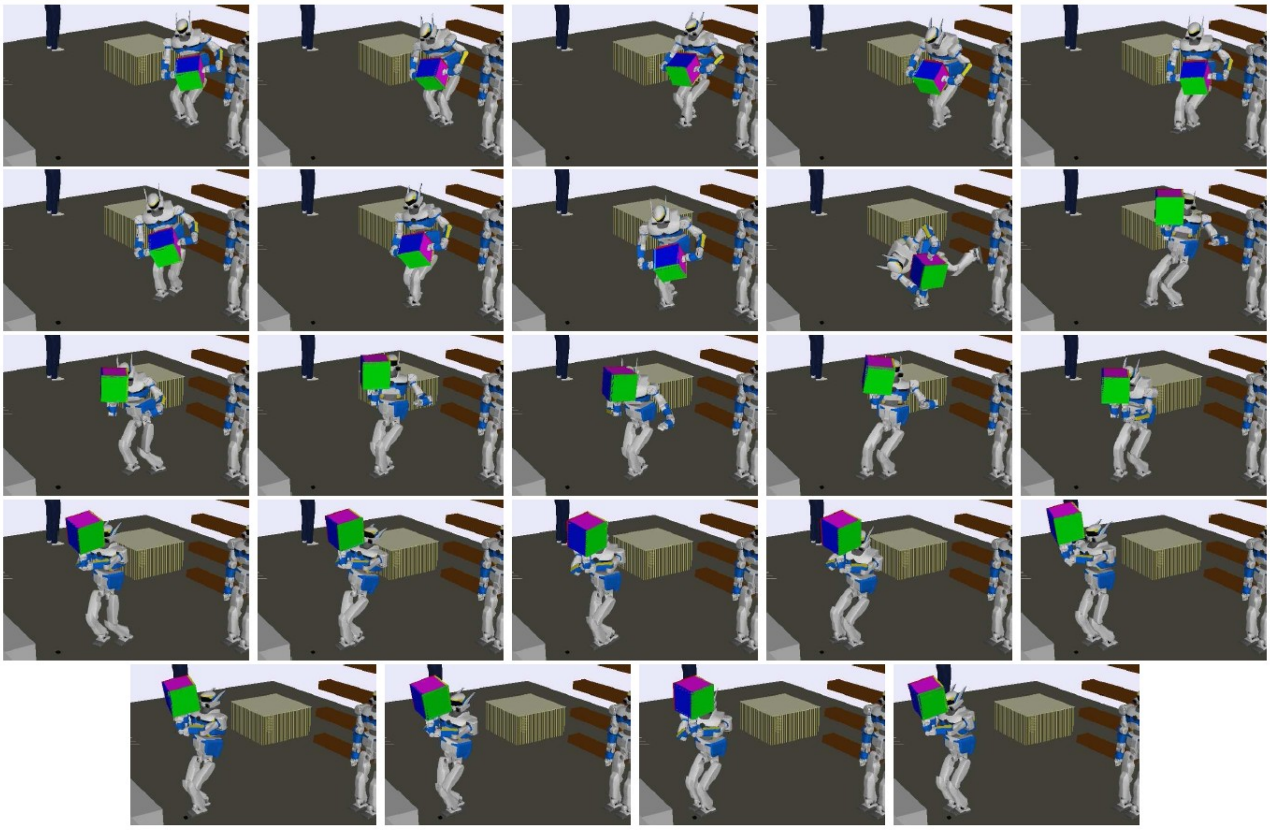


Fig.4: Simultaneous locomotion and manipulation in which the objective is to advance and at the same time rotate the box.

no specific implementation for the particular instances of the problems.

[1] K. Hauser, "Motion planning for legged and humanoid robots," Ph.D. dissertation, Stanford University, December 2008.

[2] A. Escande, "Contact planning for acyclic motion with application to humanoids," Ph.D. dissertation, University of Evry-Val d'Essone, December 2008.

[3] A. Escande, A. Kheddar, and S. Miossec, "Planning support contact-points for humanoid robots and experiments on hrp-2," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.

[4] A. Escande, A. Kheddar, S. Miossec, and S. Garsault, "Planning support contact-points for acyclic motions and experiments on hrp-2," in *Proceedings of the International Symposium on Experimental Robotics*, 2008.

[5] J.-P. Saut, A. Sahbani, S. El-Khoury, and V. Perdereau, "Dexterous manipulation planning using probabilistic roadmaps in continuous grasp subspaces," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.

[6] J. Xu, J. Koo, and Z. Li, "Finger gaits planning for multifingered manipulation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.

[7] M. Yashima, Y. Shiina, and H. Yamaguchi, "Randomized manipulation planning for a multifingered hand by switching contact modes," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2003.

[8] C. Esteves, G. Arechavelata, J. Pettré, and J.-P. Laumond, "Animation planning for virtual characters cooperation," *ACM Transactions on Graphics*, vol. 25, no. 2, pp. 319–339, April 2006.

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

[10] S. M. LaValle, *Planning Algorithms.* Cambridge University Press, 2006.

[11] C. T. Lawrence and A. L. Tits, "Nonlinear equality constraints in feasible sequential quadratic programming," *Optimization Methods and Software*, vol. 6, pp. 265–282, 1996.

[12] A. Wachter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, pp. 25–57, 2006.

[13] K. Bouyarmane, A. Escande, F. Lamiraux, and A. Kheddar, "Potential field guide for multicontact humanoid motion planning," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2009.