Dynamically Balanced and Plausible Trajectories for Human-Like Robots

C. Park[1], S. Tonneau[2], J. S. Park[1], N. Mansard[2], F. Multon[3], J. Pettré[4], D. Manocha[1]

***Abstract*— We present a motion planning algorithm to automatically compute dynamically balanced and plausible trajectories for human-like, acyclic locomotion in cluttered environments. To address the combinatorial complexity of the problem, we decompose it into two non trivial subproblems : first, to generate a discrete, acyclic, sequence of balanced contact configurations that characterize the motion; then, to extend this sequence into a continuous, dynamically balanced trajectory. The contact planning phase is achieved using a low dimensional motion planner, that computes a guide path for the root of the robot. Thanks to a reachability condition, this path can be easily mapped into a sequence of balanced contacts, selected using heuristics that enhance the robustness and task compatibility of the postures. A trajectory optimization scheme is then used to extend the sequence into a collision free and dynamically balanced trajectory. Plausibility is further enhanced using relevant cost functions, such as jerk minimization. We highlight the performance of our planner which computes dynamically balanced and plausible motions on challenging human-like motion scenarios.**

I. Introduction

Our work is motivated by commonly used approaches based on *contacts-before-motion planning* [1], [2]. Instead of computing solutions that can simultaneously satisfy all the constraints using a single framework, the underlying planning algorithms find the solutions using multiple steps, each taking into account different constraints [2], [3]. These approaches first compute a kinematically feasible sequence of discrete contact configurations or stances from the initial position to the goal position. In the second step, a continuous trajectory is computed that interpolates those contact configurations and satisfy other constraints. These decoupled planning approach tend to be more efficient as compared to methods that take into account all the constraints simultaneously.

The first step solves the problem of acyclic contact planning, which is a particular problem of motion planning. It computes a sequence of poses which are in contact with the environment that can satisfy the balance of the system. While there has been a lot of work for planning cyclic contacts for bipedal motions [4], [5], the problem remains open for more generic acyclic contacts due to its difficulty. Planning acyclic contacts is proven to having two simultaneous problems [6]: $\mathcal{P}_1$: the planning of a guide trajectory for the robot root; and $\mathcal{P}_2$: the planning of a discrete sequence of contact configurations along the guide trajectory. We reduce the complexity of the acyclic contact planning by decoupling of the problems $\mathcal{P}_1$ and $\mathcal{P}_2$ [7].

The second step computes a continuous trajectory that satisfies equilibrium and plausibility constraints from the contact sequences. The balance constraints are important in terms of guaranteeing equilibrium of the computed postures along the trajectory for the robot are balanced. Another key issue is to compute trajectories that are natural-looking or plausible. The notion of natural-looking implies that the motion trajectory closely matches the motions observed in the nature. While there is no simple and obvious mathematical metric to define plausible motion, we borrow ideas from biomechanics, neuroscience, and computer animation to characterize plausible motion using energy-based formulations [8].

Main Results: In this paper, we present a novel motion planning algorithm that uses multi-level decoupled planning to compute trajectories which are collision-free, dynamic balanced, and plausible. Our approach is based on high-dimensional trajectory optimization techniques. Our acyclic contact generation method is based on a probabilistic roadmap (PRM), which computes guide trajectories that are approximately truly feasible in offline. In runtime, the planner resolves contact sequences by refining the guide trajectory. The trajectory optimization proceeds in three phases in order to handle the high-dimension of the configuration space and various constraints. We compute an initial trajectory from the discrete configurations. Secondly, we compute collision-free trajectory using a novel reduced DOF optimization algorithm. Finally, we perform a full DOF optimization that takes into account balancing and plausibility constraints. We have applied our approach for complex benchmark scenarios and our planner is able to generate the trajectories in a few seconds, which is significantly faster than prior methods.

II. Contact Planning using a Reachability-based Planner

In this section, we provide the details of our guide trajectory computation using the reachability computation, and the generation of the discrete sequence of contact configurations [9].

[1]Chonhyon Park and Jae Sung Park and Dinesh Manocha are with the Department of Computer Science, University of North Carolina at Chapel Hill. E-mail: {chpark, jaesungp, dm}@cs.unc.edu.

[2]Steve Tonneau and Nicolas Mansard are with LAAS CNRS. E-mail: {stonneau, nmansard}@laas.fr.

[3]Franck Multon is with Unversité Rennes II. E-mail: {fmulton}@irisa.fr.

[4]Julien Pettré is with INRIA. E-mail: {julien.pettre}@inria.fr.

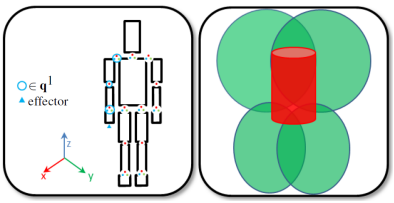

Fig. 1: Left: Robot in a rest configuration. Right: The red geometry denotes the volume of the trunk $\mathbf{W}^0$ and must remain collision-free. The green spheres are the range of the motion $\mathbf{W}^k$.

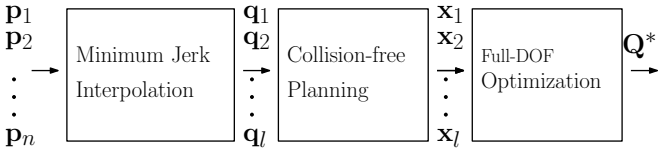

Fig. 2: We highlight different components of our multi-phase trajectory optimization. The input configurations $\mathbf{p}_1$, ..., $\mathbf{p}_n$ are interpolated to generate the initial trajectory $\mathbf{Q} = \mathbf{q}_1, ..., \mathbf{q}_l$. The first phase computes a collision-free trajectory, and subsequent phases optimize it with all constraints, including balancing and plausible motion.

A. Guide Trajectory Computation using Reachability

We first consider the problem of computing a relevant guide trajectory $\mathbf{q}^0(t) : [0, 1] \rightarrow SE(3)$ for the geometrical root of a human-like robot, connecting start and goal configurations. The objective is to enforce that any configuration $\mathbf{q}^0$ of the guide trajectory is *truly feasible*, i.e. can be mapped to a configuration with feasible contacts. We denote the set of *truly feasible* root poses by C_{reach}. For a configuration $\mathbf{q}^0$ to be in contact with an environment object O_i, O_i necessarily intersects with the range of motion $\mathbf{W}(\mathbf{q}^0)$ (Fig. 1). Furthermore, if $\mathbf{q}_0$ is *truly feasible*, then the trunk of the robot $\mathbf{W}^0(\mathbf{q}^0)$ is not colliding with the set of environment objects O. This necessary condition $C_{reach} \subset C^1_{reach}$ can be defined as:

$$C^1_{reach} = \{\mathbf{q}^0 : (\mathbf{W}(\mathbf{q}^0) \cap O \neq 0) \wedge (\mathbf{W}^0(\mathbf{q}^0) \cap O \neq 0)\} \quad (1)$$

The condition C^1_{reach} is a necessary condition that might include configurations not be truly feasible. On the other hand, replacing $\mathbf{W}^0$ in (1) with a bounding volume B^{max} of the whole robot in a given pose is a trivial sufficient condition for the completeness of the planning. The analytic formulation of a necessary and sufficient condition is not addressed in this work, but approximated in the following way. We define $\mathbf{W}^0_s$ as the scaled volume of $\mathbf{W}_0$ by a factor of s. Then we denote C^s_{reach} as:

$$C^s_{reach} = \{\mathbf{q}^0 : (\mathbf{W}(\mathbf{q}^0) \cap O \neq 0) \wedge (\mathbf{W}^0_s(\mathbf{q}^0) \cap O \neq 0)\}. \quad (2)$$

The parametrization of s allows to find a trade-off between the completeness and the efficiency of the planner.

For a fixed value of s, we implement RB-PRM, which is based on Visibility-PRM [10]. RB-PRM generates non-redundant samples in C_{reach} instead of C_{free}. To sample more efficiently configurations of C_{reach}, we bias the sampling process to generate near obstacles configurations.

B. Generation of Contact Configuration Sequences

In runtime, the guide trajectory $\mathbf{q}^0(t)$ is extended into a sequence of contact configurations. We first discretize $\mathbf{q}^0(t)$ into a configuration sequence, and optimize using the following criteria.

1) Maintaining a contact: A contacting limb at time $i-1$ remains in contact at time i until the inverse kinematics (IK) solver fails to find a collision free limb configuration. The IK solver also handles joint limits. If the solver fails to find a contact configuration, a collision free non-contact configuration is assigned to the limb.

2) Creating a contact: We create a contact with a limb which does not contact or the limb that made a contact first, to enforce quasi-static balance to a configuration which violates the balance. We project the configuration of this limb into a contact that enhances balance. If balance is not achieved with a single contact, we repeat creating new contacts until the balance is achieved.

3) Extended Force Transmission ratio(EFORT): If the planner only relies on the random sampling to create new contacts, the resulting postures will tend to look inefficient and unnatural. The Extended force transmission ratio (EFORT) [11] allows to choose configurations by measuring the task efficiency for exerting a force compatible with the direction of motion. The EFORT is measured as

$$\alpha_{EFORT}(\mathbf{q}^k, \mathbf{m}) = \frac{\mathbf{v_0 n^T m}}{\sqrt{\mathbf{m}^T \mathbf{J} \mathbf{J}^T \mathbf{m}}}, \quad (3)$$

where $\mathbf{J}$ is the Jacobian matrix of the k-th limb in configuration $\mathbf{q}^k$; $\mathbf{v}_0$ is the friction coefficient of the contact surface; $\mathbf{n}$ is the normal of the contact surface; and $\mathbf{m}$ is the direction opposite to the motion.

III. Multi-Step Trajectory Optimization

In this section, we present the details of our multiple-phase optimization to compute collision-free, balanced and natural-looking trajectories.

A. Initial Trajectory Generation

Given the n contact configuration sequences, $\mathbf{p}_1, ..., \mathbf{p}_n$, we generate an initial trajectory $\mathbf{Q}$ that is used by our multi-stage optimization algorithm. We set the velocities as zeros for the input configurations, and add m discretized configurations between each adjacent configuration pair. The m discretized configurations are interpolated using a minimum jerk trajectory [12]. The contact forces for the input configurations that satisfy the quasi-static balance are interpolated along the initial trajectory.

The initial trajectory generated has $(mn + n - m)$ discretized poses. Although the n input configurations may be feasible in terms of collision and stability constraints, the additional $m(n-1)$ configurations may violate some of these constraints. We use a trajectory optimization framework

to compute a feasible trajectory in multiple phases, that consider different constraints during each stage (Fig. 2).

B. Collision-free Planning of Reduced-DOF Motions

In the first trajectory optimization phase, we compute a collision-free trajectory. We optimize the joint positions of the configurations except for the configurations $\mathbf{q}_i$ that correspond to the initial input configurations $\mathbf{p}_j$, i.e., $i = kj$, which are already collision-free. Instead of optimizing the entire DOF of the robot, we compute solutions in a reduced DOF. For a trajectory segment $[\mathbf{q}_{kj}, ..., \mathbf{q}_{(k+1)j}]$ that is generated by the interpolation of $\mathbf{p}_j$ and $\mathbf{p}_{(j+1)}$, some end-effectors may have active contacts that exert contact forces on the robot, and their positions are not changed in $[\mathbf{q}_{kj}, ..., \mathbf{q}_{(k+1)j}]$. Therefore, we only optimize the joint positions that lie on the kinematic chains of end-effectors that have inactive contacts. We formulate the objective function for this computation as

$$\mathbf{Q}^* = \arg\min_{\mathbf{Q}} \sum_t (C_{col}(\mathbf{q}_t) + w\ddot{\mathbf{q}}_t), \tag{4}$$

where $\mathbf{q}_t$ corresponds to the interpolated configurations from trajectory keyframes $\mathbf{Q}$ and w is the weight for the trajectory smoothness cost that is used to avoid jerky motions.

$C_{col}(\mathbf{q}_t)$ corresponding constraint cost function for collisions in our optimization function. It is computed using the mesh objects $BV(\mathbf{q}_t)$ that represent the robot's body located at configuration $\mathbf{q}_t$. The constraint cost $C_{col}(\mathbf{q}_t)$ has to be $\mathbf{0}$ in order to avoid collisions with environment objects or self-collisions for the robot. We formulate this constraint as:

$$\begin{aligned} C_{col}(\mathbf{q}_t) = &\sum_{\substack{BV \in A \\ O_i \in O}} PD(BV(\mathbf{q}_t), O_i)^2 \\ &+ \sum_{\substack{BV_i, BV_j \in A \\ BV_i \neq BV_j}} PD(BV_i(\mathbf{q}_t), BV_j(\mathbf{q}_t))^2, \end{aligned} \tag{5}$$

where $PD(B_1, B_2)$ is the penetration depth between objects B_1 and B_2, which refers to the extend of inter-penetration between two overlapping objects.

In order to compute collision-free solutions using the constraint function $C_{col}(\mathbf{q}_t)$, simple local optimization methods may not work well and can get stuck in a local minima. Instead, we use a parallel stochastic optimization algorithm that computes multiple trajectories [13] to improve the performance and the probability of finding the global minima.

C. Full-DOF Optimization with Constraints

In the second phase of the optimization, we use a constrained optimization to optimize the trajectory $\mathbf{Q}$ with cost functions that correspond to constraints corresponding to collision-free motion, dynamic stability, and plausibility.

While the optimization for collision-free planning only requires the configuration $\mathbf{q}$, the optimization of our second phase requires additional parameters, $\mathbf{f}_i$, which corresponds to the contact forces. With these parameters, the optimization variables for a keyframe i are defined as follows:

$$\mathbf{x}_i = \left[\mathbf{q}_i, \dot{\mathbf{q}}_i, \mathbf{f}_i, \dot{\mathbf{f}}_i\right]. \tag{6}$$

The objective function of for this optimization is given as

$$\min_{\mathbf{x}_1, ..., \mathbf{x}_l} \sum_t C(\mathbf{x}_t), \tag{7}$$

where $C(\mathbf{x}_i)$ represents the cost function for the parameter $\mathbf{x}_i$ at time t. This cost function is decomposed as:

$$f(\mathbf{x}_i) = C_{col}(\mathbf{q}_i) + C_{ds}(\mathbf{x}_i) + C_{pla}(\mathbf{x}_i), \tag{8}$$

where $C_{col}(\mathbf{q}_i)$, $C_{ds}(\mathbf{x}_i)$, $C_{pla}(\mathbf{x}_i)$ represent the costs for the collision-free constraint, dynamic stability constraint, and plausible motion constraint, respectively.

1) Collision-free Constraint: We use the same constraint function $C_{col}(\mathbf{q}_i)$ defined above to ensure that the configurations are still collision-free during this optimization phase.

2) Dynamically Balanced Constraint: The stability constraints can be evaluated using Newton-Euler equation [14], based on external forces (gravity force, reaction force, etc.) and internal forces (joint forces, inertial force, etc.) that are exerted on the body. A pose is dynamically balanced if all the exerting forces and torques result in an equilibrium. The reaction forces can exist only if the corresponding end-effector makes an active contact. An end-effector contact is approximated by multiple contacts in a plane, and each contact exerts its own contact force. Therefore the stability cost function $C_{ds}(\mathbf{x}_i)$ is formulated as follows:

$$C_{ds}(\mathbf{x}_i) = \| \sum^{J} w_c(\mathbf{x}_i) + w_g(\mathbf{q}_i) + w_i(\mathbf{q}_i) \|^2, \tag{9}$$

where J is the total number of contact points, $w_c(\mathbf{x}_i)$ is the contact wrench for $\mathbf{x}_i$, and $w_g(\mathbf{q}_i)$ and $w_i(\mathbf{q}_i)$ correspond to the gravity and inertia wrenches for a configuration $\mathbf{q}_i$, respectively. $w_c(\mathbf{x}_i)$ is set to 0 for non-active contacts.

3) Plausible Motion Constraint: We use the torque minimization constraint to compute plausible motions. In particular, we use the inverse dynamics to compute the joint torque for the configuration $\mathbf{q}_t$ and the contact forces $\mathbf{f}_t$, and formulate the constraint cost as the squared sum:

$$C_{pla}(\mathbf{x}_i) = \sum_j w(j) \cdot \tau_j(\mathbf{x}_t), \tag{10}$$

where $\tau_j(\mathbf{x}_t)$ is the joint torque of the j-th joint. We solve the non-linear optimization problem corresponding to (7) using the L-BFGS algorithm using numerical derivatives with box constraints of the variables, which correspond to the joint position and velocity limits.

IV. Results

In this section, we highlight the performance of our algorithm on different benchmark scenarios.

A. Trajectory Computation with RB-PRM

The robot used in the experiments are shown in Fig. 1. The human-like robot has 34 DOFs, which is decomposed as 6 root DOFs and has 7 DOFs for each limb. We evaluate the performance of our planner in three challening benchmark scenarios.

	RB-PRM Time (s)	# of input configurations	Trajectory Optimization Time (s)
Climbing	25	17	34.1
Crouching	73	11	23.9
Crawling	5	284	128.1
MoCap	N/A	4	15.8
Mocap Lifting	N/A	4	35.5

TABLE I: Planning results for our benchmarks. We measure the planning time for the RB-PRM and the trajectory optimization.

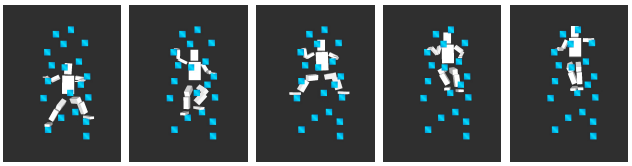

Fig. 3: The computed dynamically balanced and plausible trajectory of climbing benchmark. The combination of balancing and plausibility constraints results in improved trajectories.

1) **Climbing**: The input configuration is climbing on a wall using several blocks on the wall. We compute the the trajectory with the collision, balancing, and plausibility constraints. Fig. 3 highlight the computed trajectory with the plausible motion constraints, respectively.
2) **Crawling**: The robot crawls through the narrow front window of a truck.
3) **Crouching**: The robot goes from a standing to a crouching position to pass under an obstacle (i.e collision-free motion).

B. Trajectory Computation from Motion Capture Datasets

Our second set of benchmarks for the trajectory optimization use input poses from motion capture data. We extract four configurations that correspond double support with both feet from a walking motion capture date for the model. In order to validate the dynamic balancing and the plausiblity constraints of our approach, we computed the continuous trajectories from the poses. The model used in Fig. 4 is lifting a heavy object in its right hand. As shown in the figure, the right arm does not move much in the computed motion to reduce the joint torques on the right arm.

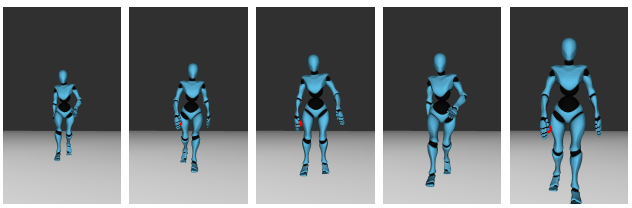

Fig. 4: The computed trajectory from Motion Capture datasets.

V. Conclusions and Future Work

In this paper we present a novel motion planning algorithm that compute dynamically balanced and stable trajectories. We use a reachability-based PRM and a multi-phased optimization strategy based on a decoupled planner and are able to compute collision-tree trajectories that satisfy all these constraints. As compared to prior methods, our approach is much faster and the resulting motion trajectories are dynamically balances as opposed to quasi static motions. We highlight the performance on complex human motion benchmarks.

Our future work will focus on a more accurate formulation of C_{reach} and the trajectory optimization, as they can still get stuck in local minima. In terms of future work, we would like to overcome these limitations. We would also like to integrate our algorithm with a humanoid and also evaluate the performance on high-DOF manipulators.

References

[1] A. Escande, A. Kheddar, and S. Miossec, "Planning contact points for humanoid robots," *Robotics and Autonomous Systems*, vol. 61, no. 5, pp. 428–442, 2013.

[2] J. J. Kuffner Jr, 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.

[3] K. Hauser, "Fast interpolation and time-optimization with contact," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1231–1250, 2014.

[4] 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*, 2003, pp. 1620–1626.

[5] L. Baudouin, N. Perrin, T. Moulard, F. Lamiraux, O. Stasse, and E. Yoshida, "Real-time replanning using 3d environment for humanoid robot," in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*. IEEE, 2011, pp. 584–589.

[6] T. Bretl, S. Rock, J.-C. Latombe, B. Kennedy, and H. Aghazarian, "Free-climbing with a multi-use robot," in *Experimental Robotics IX*. Springer, 2006, pp. 449–458.

[7] A. Escande, A. Kheddar, S. Miossec, and S. Garsault, "Planning support contact-points for acyclic motions and experiments on hrp-2," in *Experimental Robotics*. Springer, 2009, pp. 293–302.

[8] O. Khatib, J. Warren, V. De Sapio, and L. Sentis, "Human-like motion from physiologically-based potential energies," in *On advances in robot kinematics*. Springer, 2004, pp. 145–154.

[9] S. Tonneau, N. Mansard, C. Park, D. Manocha, F. Multon, and J. Pettré, "A reachability-based planner for sequences of acyclic contacts in cluttered environments," in *Proceedings of International Symposium on Robotics Research*, 2015.

[10] C. Nissoux, T. Siméon, and J.-P. Laumond, "Visibility based probabilistic roadmaps," in *Intelligent Robots and Systems, 1999. IROS'99. Proceedings. 1999 IEEE/RSJ International Conference on*, vol. 3. IEEE, 1999, pp. 1316–1321.

[11] S. Tonneau, J. Pettré, and F. Multon, "Using task efficient contact configurations to animate creatures in arbitrary environments," *Computers & Graphics*, vol. 45, pp. 40–50, 2014.

[12] N. Hogan, "Adaptive control of mechanical impedance by coactivation of antagonist muscles," *IEEE Transactions on Automatic Control*, vol. 29, no. 8, pp. 681–690, 1984.

[13] C. Park, J. Pan, and D. Manocha, "ITOMP: Incremental trajectory optimization for real-time replanning in dynamic environments," in *Proceedings of International Conference on Automated Planning and Scheduling*, 2012.

[14] J. C. Trinkle, J.-S. Pang, S. Sudarsky, and G. Lo, "On dynamic multi-rigid-body contact problems with coulomb friction," *ZAMM-Journal of Applied Mathematics and Mechanics*, vol. 77, no. 4, pp. 267–279, 1997.