Kinodynamic Motion Planners based on Velocity Interval Propagation

Stéphane CARON, Yoshihiko NAKAMURA, Quang-Cuong PHAM
Department of Mechano-Informatics, University of Tokyo, Japan

1. Introduction

Humanoid robotics has spawned several fields of active research. When it comes to dynamic motion control, three lines of work stand out: reduced models (combined with inverse kinematics), local controllers and global planning.

The concept of Zero-Moment Point (ZMP) [17] allows for simplified mathematical models of the complex kinematic structure of humanoid robots; in particular, to generate walking patterns of humanoid robots, [3] designed the "cart-table" model. Once a walking trajectory has been decided under this reduced model, inverse kinematics is used to compute back the whole-body posture. This approach enables planning of dynamic trajectories in real-time, yet it does not provide any completeness guarantee because of the surjection to the simplified model.

Local controllers stem from the redundancy of humanoid robots as articulated systems: objectives such as achieving a given end-effector pose have infinitely many solutions, the free space of which can be used to specify sub-objectives (obstacle avoidance, energy consumption, etc.) This formulation has been well studied in the literature and Jacobian-based solutions designed for both inverse kinematics [9, 15] and dynamics [5]. These methods yield smooth solutions and can be used in real-time for whole-body motion planning; they are, however, prone to fall into local minima and present no completeness guarantee, grounding the need for more global planners.

Randomized motion planning algorithms have been developped and successfully applied to systems with a fair amount of DoFs within the past twenty years. The two most prominent algorithms in this field are Probabilistic Roadmaps (PRM) [4] and Rapidly-exploring Random Trees (RRT) [8]. Both result from the same approach: sample robot states randomly and try to connect them to an existing roadmap/tree using local steering methods. They provide theoretical guarantees such as *probabilistic completeness* (if there is a solution, running the algorithm forever will eventually find it) [8], but suffer from the curse of dimensionality: their computational complexity is exponential in the dimension of the explored space. As such, they are not fit for real-time planning.

Successful whole-body motion planning of humanoid robots using RRTs appeared for the first time in [6]. In this work, planning is performed in the configuration space and a dynamics filter is applied to time-parametrize the returned path into a dynamically-feasible trajectory. However, to ensure that such time parametrization is possible, all nodes in the tree are restricted to statically-stable configurations, which is a severe limitation since many humanoid motions, including walk, are dynamically balanced without being statically stable. Another approach advocated in [8] is to perform planning in the state space (*i.e.*, adding velocity coordinates), yet it doubles the space's dimension, which is intractable for high-DoF systems such as humanoid robots.

In the present paper, we present and further develop a motion planning approach recently proposed in [10], which allows planning with dynamics constraints while staying in the configuration space; thus avoiding the complexity explosion mentioned above. We briefly discuss how this approach can be extended to handle ZMP constraints, which may give rise to a new family of efficient motion planners for humanoid robots.

The outline of the paper is as follows: after defining the problem and terminology in Section 2., we integrate the Velocity Interval Propagation algorithm from [10] into RRT in Section 3.. Experimental analysis of our solution is provided in Section 4., and we finally describe in Section 5. how ZMP balance constraints can be integrated into our framework to develop a complete kinodynamic planner for humanoid robots.

2. Problem statement

We denote by $\mathcal{C}$ the *configuration space* of the robot, *i.e.*, the space of vectors $\mathbf{q} = (q_1, \ldots, q_n)$ where q_i represents the joint-angle of the robot's i^{th} degree of freedom. We call *state space* the set $\mathcal{X}$ of vectors $\mathbf{x} = (\mathbf{q}\ \dot{\mathbf{q}})^{\mathrm{T}}$ including configurations and velocity coordinates. We assume the system is driven by the non-linear relation

$$\dot{x} = f(x, u), \tag{1}$$

where u is a control vector belonging to some compact input space $\mathcal{U}$.

We call *path* of the system any continuous curve $P \subset \mathcal{C}$ of the configuration space. Paths carry all the kinematic information of a motion. We call *trajectory* a time-parametrized path, *i.e.*, a function $q : [0, T] \to P$ such that $q(0)$ and $q(T)$ are both extremities of the underlying path P. We will denote by $\mathsf{first}(q) := q(0)$ and $\mathsf{last}(q) := q(T)$ these two extremities. We say that a path is *traversable* when it admits a time-parametrization satisfying the system dynamics (1),

i.e., when there exists a control $u : [0, T] \to \mathcal{U}$ such that both $\forall t \in [0, T], (\dot{q}, \ddot{q}) = f((q, \dot{q}), u)$.

3. Velocity Interval Propagation

3.1 RRT

The RRT algorithm was introduced in [7], further analyzed in [8] and applied to dynamically-stable motion planning of humanoid robots in [6]. In the present section, we will suppose an RRT growing in the state space $\mathcal{X}$, *i.e.*, the probabilistically complete planner described in [8]. It requires the following components:

Metric: a function $\sigma : \mathcal{X}^2 \to [0, +\infty)$ representing the distance between two states.

Trajectory checker: a function trajcheck ensuring that there is no collision along the path and that all differential conditions are satisfied.

Local steering: a method steer which may be any of the local methods mentioned in section 1.. During local steering from a state x_{start}, a "close" state x_{goal} is provided and a control driving the system from x_{start} to x_{goal} is sought after.

In what follows, we will distinguish two approaches to local steering:

- *using forward dynamics*, where a set of *control* functions $u : [0, T_u] \to \mathcal{U}$ is considered and the one driving the system closest to x_{goal} is retained (this approach can be found in [7, 2]);
- *using inverse dynamics*, where a set of *state* functions $x : [0, T_x] \to \mathcal{X}$ driving the system to x_{goal} is considered, and a suitable control is computed using inverse dynamics (this approach is the one used in [6].

The latter is useful when additional conditions (such as static stability [6]) are enforced on configurations or velocities, while they are difficult to achieve with a forward dynamics-based approach. However, using forward dynamics is much more efficient for *exploring* the system's (dynamically) reachable space.

Algorithm 1 describes RRT planning in the state space.

Algorithm 1 RRT(x_{init}, N, σ, steer, trajcheck):

```
1:  (V, E) ← (x_init}, ∅)
2:  for N steps do
3:     x_rand ← new sample from X_free
4:     x_parent ← arg min{σ(x, x_rand) | x ∈ V}
5:     traj ← steer(x_parent, x_rand)
6:     if trajcheck(traj) then
7:        x_last ← last(traj)
8:        V ← V ∪ {x_last}
9:        E ← E ∪ {(x_parent, x_last)}
10:    end if
11: end for
12: return T = (V, E)
```

3.2 Extension to k Nearest Neighbors

We first considered the extension to k nearest neighbors out of necessity: kinodynamic feasiblity is a strong constraint imposed on local paths, which resulted in a high rejection rate of sampled states. We found that considering not one, but up to k nearest neighbors (according to the metric σ) resulted in a substantial performance increase. It turns out that this observation can be related to the distortion between σ and the ideal metric associated with steer, as discussed below.

kNN-RRT's pseudo-code can be obtained from Algorithm 1 by replacing the right operand of line 4 with $\arg\min_k\{\sigma(x, x_{\text{rand}}) \mid x \in V\}$, *i.e.*, the set of k nearest neighbors of x_{rand} in the tree. VIP-RRT includes this modification (see Algorithm 2).

The choice of the metric σ is crucial to the performance of RRTs. Euclidean distances are a default choice, but they turn out to be ill-suited for problems as simple as the single inverted pendulum [14]. Given a local steering method steer, [8] suggested that the ideal metric for RRT is the *cost-to-go* function associated with steer, which is such that $\arg\min\{\textit{cost-to-go}(x, x_{\text{rand}}) \mid x \in V\}$ would return the *best antecedent* for x_{rand} in the tree at line 4 of Algorithm 1. However, computing this metric is as difficult as solving the entire planning problem. The k nearest-neighbor heuristic approximates it. For $k = 1$, it relies only on the metric σ. For $k = |V|$ (size of the tree), it yields the optimal antecedent. The quality of the approximation depends on the distortion between σ and *cost-to-go*.

3.3 VIP Algorithm

Velocity Interval Propagation, which we introduced in [10], stems from the time-optimal control algorithm [1, 13, 16, 12]. Given a path in the configuration space, this algorithm detects whether a traversal of that path respecting differential constraints exists, and if so, returns the one of minimum duration. This approach is based on the derivation of Maximum Velocity Curves (MVCs) of the system. We showed in [10] how MVCs can be used to propagate intervals of *reachable velocities*, an operation that we coined "VIP" for Velocity Interval Propagation. Its prototype goes as follows:

Inputs: a path $P \subset \mathcal{C}_{\text{free}}$ and an interval $[v_{\min}^{\text{beg}}, v_{\max}^{\text{beg}}]$ of velocities available at the beginning of P;

Output: the interval $[v_{\min}^{\text{end}}, v_{\max}^{\text{end}}]$ of reachable velocities at the end of P.

Velocity intervals are *certificates of traversability*: when extending the tree from q_{parent} to q_{rand}, there exist a trajectory *from the root of the tree* to q_{rand} if the VIP call from q_{parent} was successful. This is a necessary and sufficient condition for single-valued MVC systems, and only a sufficient condition for multiple-valued MVC systems. In other words, the

global traversability property can be checked incrementally while growing the tree. Although the global traversability was ensured through constraining all tree nodes to statically stable configurations in [6], using velocity intervals allows for dynamically balanced trajectories.

3.4 VIP-RRT

As discussed in subsection 3.2, straightforward integration of VIP into a simple RRT does not yield satisfying results due to severe constraints on local steering. This can be worked around with the kNN heuristic. Pseudo-code for the resulting solution is shown in Algorithm 2, where $\nu := [v_{\min}, v_{\max}]$ denotes velocity intervals.

Algorithm 2 VIP-RRT(k, q_{init}, N, $\sigma_{\mathcal{C}}$, $\mathsf{steer}_{\mathcal{C}}$):

$(V, E) \leftarrow (\{x_{\text{init}}\}, \emptyset)$
for N steps **do**
  $q_{\text{rand}} \leftarrow$ new sample from $\mathcal{C}_{\text{free}}$
  nearest $\leftarrow \arg\min_k \{\sigma_{\mathcal{C}}(q, q_{\text{rand}}) \mid (q, \nu) \in V\}$
  for each $(q_{\text{near}}, \nu_{\text{near}}) \in$ nearest **do**
    path $\leftarrow \mathsf{steer}_{\mathcal{C}}(q_{\text{near}}, q_{\text{rand}})$
    $q_{\text{last}} \leftarrow \mathsf{last}$(path)
    $\nu_{\text{last}} \leftarrow \mathsf{VIP}$(path)
    if ν_{last} is not empty **then**
      $V \leftarrow V \cup \{(q_{\text{last}}, \nu_{\text{last}})\}$
      $E \leftarrow E \cup \{(q_{\text{parent}}, q_{\text{last}})\}$
      break the inner loop
    end if
  end for
end for
return $\mathcal{T} = (V, E)$

4. Simulation results

Preliminary simulations were run on a 2-DoF inverted pendulum with torque constraints. In this setting, the goal is to swing-up the pendulum from its stable, downward equilibrium to its unstable, upward one. The minimum torque required to perform the motion quasi-statically is $\tau_1 = 15.68$ Nm at the first joint. We set a torque constraint of $\tau_1^{\max} = 8$ Nm on this joint, and $\tau_2^{\max} = 4$ Nm on the second one. Although we already performed simulations with this pendulum in [10], the torque limits are lower here, which resulted in a different behavior that we discuss below.

Figure 1 shows the performances of all three algorithms (RRT, kNN-RRT and VIP-RRT) over forty runs of this system. We chose $k = 40$, as identified in the supplementary material of [10] for the pendulum. The left plot shows the fraction of successful planners when given more computation time. Note that the problem is quite difficult, as indicated by the fact that only one instance of the simple $\mathcal{X}$-space RRT found a solution. kNN-RRT yielded significant improvement with eight successful planners, which we interpret as a benefit of its local approximation of the ideal metric. VIP-RRT performed best in the lot with more than twenty successful planners, a substantial gain that we attribute to the ability to plan in configuration space (versus state space for RRT and kNN-RRT).

The right part of the Figure shows the distance to the goal area averaged over all runs. After making steep progress toward the goal, RRT and kNN-RRT's progress slows down to almost zero. This phenomenon did not appear in [10]. It results from the increased difficulty of the problem: compared to our previous setting, successful trajectories now require three pumping motions instead of two, which implies exponentially longer trajectories in the state space. Therefore, even though kNN-RRTs (resp. RRTs) find the first swing-ups in around 30 min (resp. 1 hr), it will take them exponentially longer to find the last one. Meanwhile, to discover an additional pumping motion, VIP-RRT only needs to sample its extremal configuration, thanks to sparse trees and the kNN heuristic.

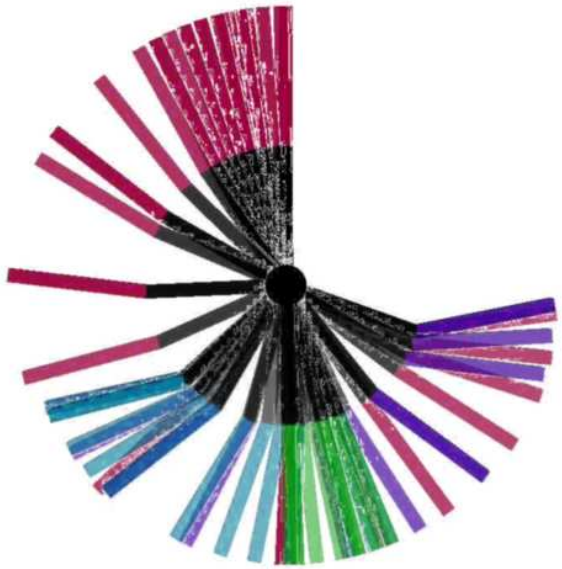

Fig.2 Successful trajectory planned with VIP-RRT for the simulated pendulum. Color changes indicate a change in the direction of movement: green corresponds to the first swing-up (to the right), then blue (to the left), then magenta (to the right) and finally red until the stand-up configuration.

5. Towards a complete kinodynamic planner for humanoid robots

For humanoid robots, balance constraints are of primary concern. A well-known condition for dynamic balance is that the ZMP stays within the convex hull of the ground contact points at any time instant [17]. [11] described how to perform time-optimal path parametrization for dynamic motions of humanoid robots: first, an MVC is derived from ZMP balance constraints, and then the time parametrization algorithm is applied to this MVC. Independently,

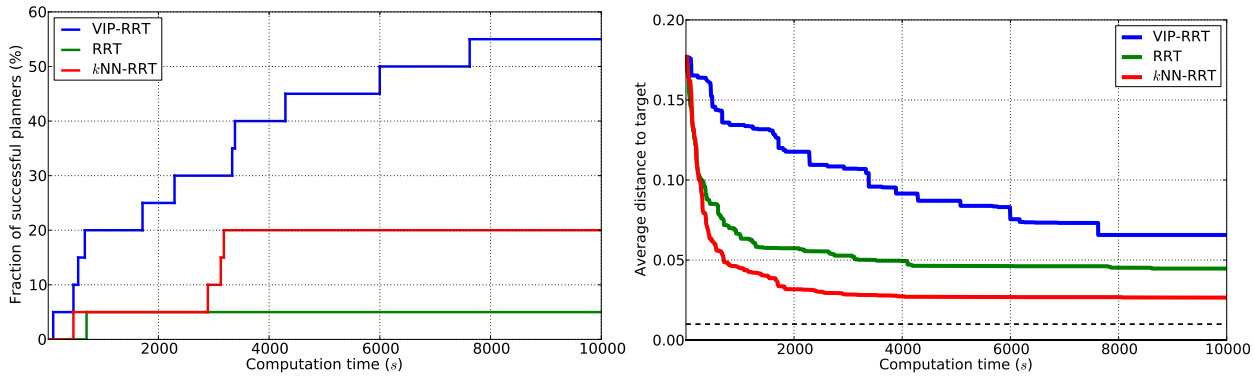

Fig.1 RRT, kNN-RRT and VIP-RRT ran over 40 instances of a double inverted pendulum with torque limits (8 Nm on the first joint and 4 Nm on the second one). The horizontal axis shows computation time, while the vertical axis represents the number of successful planners (left) and the distance to the goal area (right) averaged over all runs.

we described in [10] how intervals of reachable velocities can be propagated along MVCs.

Our plan is now to integrate the two lines of work, *i.e.*, to propagate velocity intervals along MVCs derived from ZMP balance constraints of humanoid robots. The resulting planner would, as Algorithm 2, plan in the configuration space. When steering from one configuration to another, local paths will be interpolated and checked using the VIP routine: if the propagation succeeds, we can parametrize them into trajectories respecting the robot's holonomic and balance constraints.

References

[1] James E Bobrow, Steven Dubowsky, and JS Gibson. Time-optimal control of robotic manipulators along specified paths. *The International Journal of Robotics Research*, 4(3):3–17, 1985.

[2] David Hsu, Robert Kindel, Jean-Claude Latombe, and Stephen Rock. Randomized kinodynamic motion planning with moving obstacles. *The International Journal of Robotics Research*, 21(3):233–255, 2002.

[3] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kiyoshi Fujiwara, Kensuke Harada, Kazuhito Yokoi, and Hirohisa Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 2, pages 1620–1626. IEEE, 2003.

[4] Lydia E Kavraki, Petr Svestka, J-C Latombe, and Mark H Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *Robotics and Automation, IEEE Transactions on*, 12(4):566–580, 1996.

[5] Oussama Khatib, Luis Sentis, Jaeheung Park, and James Warren. Whole-body dynamic behavior and control of human-like robots. *International Journal of Humanoid Robotics*, 1(01):29–43, 2004.

[6] James J Kuffner, Satoshi Kagami, Koichi Nishiwaki, Masayuki Inaba, and Hirochika Inoue. Dynamically-stable motion planning for humanoid robots. *Autonomous Robots*, 12(1):105–118, 2002.

[7] Steven M LaValle. Rapidly-exploring random trees a new tool for path planning. 1998.

[8] Steven M LaValle and James J Kuffner. Randomized kinodynamic planning. *The International Journal of Robotics Research*, 20(5):378–400, 2001.

[9] Yoshihiko Nakamura, Hideo Hanafusa, and Tsuneo Yoshikawa. Task-priority based redundancy control of robot manipulators. *The International Journal of Robotics Research*, 6(2):3–15, 1987.

[10] Quang-Cuong Pham, Stéphane Caron, and Yoshihiko Nakamura. Kinodynamic planning in the configuration space via velocity interval propagation. *Robotics: Science and System*, 2013.

[11] Quang-Cuong Pham and Yoshihiko Nakamura. Time-optimal path parameterization for critically dynamic motions of humanoid robots. In *IEEE-RAS International Conference on Humanoid Robots*, 2012.

[12] Zvi Shiller and Lu Hsueh-Hen. Computation of path constrained time optimal motions with dynamic singularities. *Journal of dynamic systems, measurement, and control*, 114(1):34–40, 1992.

[13] Kang Shin and N McKay. Selection of near-minimum time geometric paths for robotic manipulators. *Automatic Control, IEEE Transactions on*, 31(6):501–511, 1986.

[14] Alexander Shkolnik, Matthew Walter, and Russ Tedrake. Reachability-guided sampling for planning under differential constraints. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 2859–2865. IEEE, 2009.

[15] Bruno Siciliano and J-JE 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. IEEE, 1991.

[16] J-JE Slotine and Hyun S Yang. Improving the efficiency of time-optimal path-following algorithms. *Robotics and Automation, IEEE Transactions on*, 5(1):118–124, 1989.

[17] Miomir Vukobratovic, Branislav Borovac, and Dragoljub SurdiloviC. Zero-moment point–proper interpretation and new applications. In *Proceedings of the IEEE/RAS International Conference on Humanoid Robots*, volume 244. Piscataway, NJ, USA: IEEE, 2001.