Maximize-perturb-minimize: a fast and effective heuristic to obtain sets of locally optimal robot postures

Martim Brandão, Kenji Hashimoto and Atsuo Takanishi

***Abstract*— Complex robots such as legged and humanoid robots are often characterized by non-convex optimization landscapes with multiple local minima. Obtaining sets of these local minima has interesting applications in global optimization, as well as in smart teleoperation interfaces with automatic posture suggestions.**

In this paper we propose a new heuristic method to obtain sets of local minima, which is to run multiple minimization problems initialized around a local maximum. The method is simple, fast, and produces diverse postures from a single nominal posture. Results on the robot WAREC-1 using a sum-of-squared-torques cost function show that our method quickly obtains lower-cost postures than typical random restart strategies. We further show that obtained postures are more diverse than when sampling around nominal postures, and that they are more likely to be feasible when compared to a uniform-sampling strategy. We also show that lack of completeness leads to the method being most useful when computation has to be fast, but not on very large computation time budgets.

I. INTRODUCTION

Finding sets of alternative locally optimal postures is an important problem in robotics. One of the motivations to explore this problem can be to increase robot efficiency by finding lower-cost postures. Another interesting application is teleoperation. See Figure 1 for an example. The idea is to provide these locally-optimal postures to an operator: to help jump-starting motion design, or for the operator to choose from according to his intuition. While posture generation algorithms already exist, they converge to a single local minimum and have problems with infeasible initializations [1], [2]. Exploration is usually done through perturbations around a nominal posture or through random uniform sampling, which can either lead to too few or too much variation in the produced outputs. This paper is an attempt to alleviate these issues. Our contributions are the following:

- We introduce a new heuristic algorithm to obtain sets of locally optimal robot postures, which consists of minimization with multi-starts around a local maximum
- We give an intuition for why the heuristic makes sense using the example of the robot WAREC-1 [3]
- We qualitatively and quantitatively evaluate the algorithm in terms of diversity of solutions, success rates and minimum costs obtained
- We discuss when the algorithm compares favorably to other exploration methods, as well as when it ceases to do so.

*This work was supported by ImPACT TRC Program of Council for Science, Technology and Innovation (Cabinet Office, Government of Japan).

M. Brandão is with the Research Institute for Science and Engineering, Waseda University: #41-304, 17 Kikui-cho, Shinjuku-ku, Tokyo 162-0044, JAPAN.

K. Hashimoto is with the Waseda Institute for Advanced Study, and is a researcher at the Humanoid Robotics Institute (HRI).

A. Takanishi is with the Department of Modern Mechanical Engineering, Waseda University; and the director of the Humanoid Robotics Institute (HRI), Waseda University.

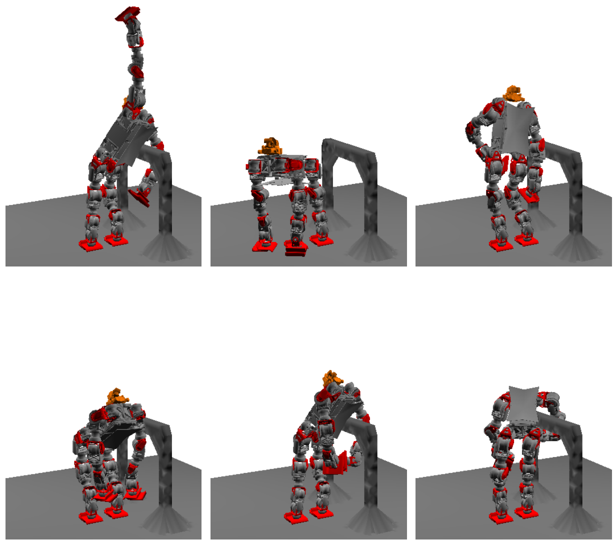

Fig. 1. Generating postures on different local minima of a cost function. Applications to global optimization and teleoperation interfaces (e.g. providing multiple low-cost alternative postures for a teleoperator to jump-start posture design).

II. RELATED WORK

The problem of generating multiple robot postures satisfying a task, for example for robot teleoperation interfaces, is closely related to the problem of character animation in computer graphics. The focus there is usually on making postures and motion human-like, by biasing inverse kinematics towards human postures obtained in motion-capture systems [4], [5]. For example, [5] learns a latent space of motion-capture motion which is sampled during motion generation. A more recent example is that of [6], where a manifold of human locomotion is learned using a convolutional auto-encoder and then used for motion synthesis and to constrain an animator's motion to look natural. On the other hand, in robotics there are high requirements on the energetic cost of the motion, and thus even if a certain posture appearance is desired, a cost function such as torques, battery consumption or other usually needs to be locally minimized [7]. With this in mind, in this paper we take a slightly different look at the problem of generating postures for a task, compared to the graphics literature, which is to obtain postures on local minima of the desired cost function within some distance of a nominal posture. So while a teleoperator may want "natural" postures, we focus on sampling those which are locally optimal in some energetic sense. Furthermore, recent findings show that optimizing energy consumption in humanoid robots may actually lead to human-like motion [7], which suggests that our approach of generating postures on local minima might coincidently lead to human-like postures on humanoids. We will qualitatively see that this is the case in Section IV-A.

In the robotics community, several optimization algorithms for posture generation and motion planning have been proposed. These algorithms are local, based on sequential quadratic programming [1], projected gradient descent [2] and others. Since they are only local, they are often prone to get stuck at infeasible local minima or not to explore other lower cost alternatives. Some algorithms [2], [8], add noise to the optimization process to promote exploration. Others use multi-starts (e.g. random restarts) from different seeds and pick the lowest-cost trajectory [1], [9]. Similarly, in sampling-based motion planning such as RRTs [10] and PRM variants [11], robot configurations are typically sampled uniformly within joint limits or close to a nominal posture. Uniform sampling can take too long to find a feasible posture, however, and sampling around a nominal posture may be too conservative or require much tuning until the right amount of exploration is obtained. Our approach to the problem of generating multiple postures at local minima tries to do so within close distance of nominal postures, while removing the need of tuning exploration radius for decent performance, and still doing it quickly. We do so by sampling from a local maximum, which has the advantage that any small perturbation is enough to produce a descent. On highly-nonlinear functions such as those in legged and humanoid robots, our sampling strategy will usually produce a different direction of descent each time, and thus lead to discovering more local minima within the neighborhood of preferred postures. Our algorithm is therefore more likely to quickly find diverse postures for a teleoperator, or for reducing final cost on a motion planner.

III. METHOD

A. Problem

In this paper we consider posture generation problems of the form

$$\begin{aligned} \underset{q \in \mathbb{R}^K}{\text{minimize}} \quad & f(q) \\ \text{subject to} \quad & g(q) = 0 \\ & h(q) \leq 0, \end{aligned} \tag{1}$$

where $q \in \mathbb{R}^K$ are the degrees of freedom (DOF) of the robot, and the cost function $f(q)$ is non-convex with multiple local minima. For convenience we can equivalently write (1) as

$$\underset{q \in C}{\text{minimize}} \quad f(q), \tag{2}$$

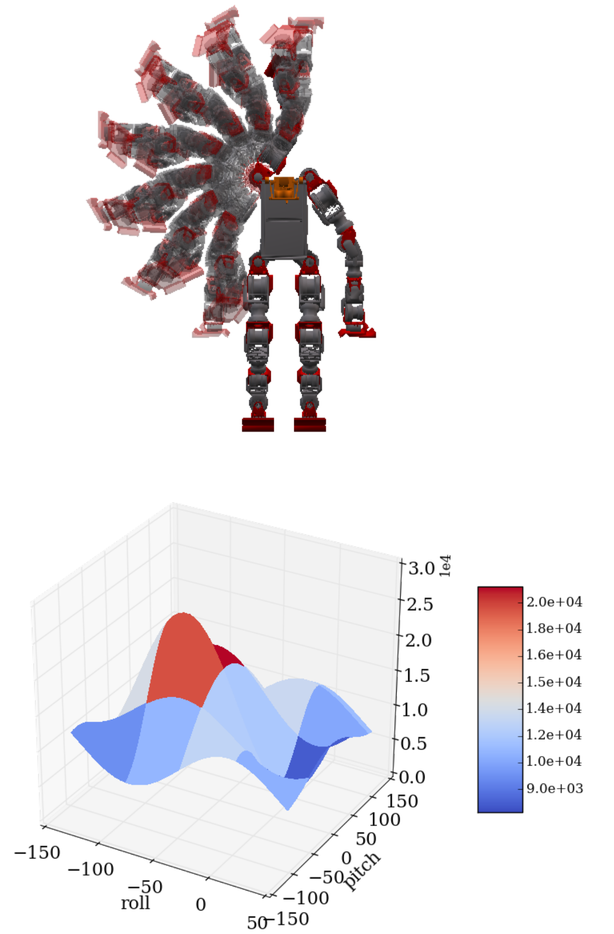

Fig. 2. Optimization landscape around a nominal posture, when varying the shoulder roll and pitch joint angles. X-Y axes are in degrees. f is the sum of squared joint torques.

where the set $C \subseteq \mathbb{R}^K$ is the intersection of $\mathbb{R}^K$ with constraints g and h.

We wish to find multiple local minima of (2), in order to either use the lowest-cost minimum found or to display the different solutions for an operator to choose from.

B. Maximize-perturb-minimize

To motivate our algorithm, take for example the robot WAREC-1 [3], shown in Figure 2. The robot is standing on its nominal posture, and its shoulder roll and pitch joints are moved within their limits. Using a typical sum-of-squared-torques cost function, $f(q) = \sum_i \tau_i(q)^2$, the figure shows one local minimum and one local maximum in the interior of the set (corresponding to vertical arm pointing up, and horizontal arm, respectively). There are also other local minima and maxima at the boundaries of the set (e.g. vertical arm pointing down).

Note that depending on the size of the basin of attraction, looking for solutions around a nominal posture might lead to always finding the same local minimum or another minimum immediately next to it. However, around a local maximum different directions point towards different local minima, even for the slightest perturbation. This observation motivates the following algorithm: to sample points around local maxima as initializations for (2) in order to obtain different local minima.

We then propose to solve (2) using the following Algorithm 1.

Algorithm 1 Maximize-perturb-minimize

input: q_0, σ^2
Solve $\tilde{q} \leftarrow \underset{q \in C}{\operatorname{argmax}} \quad f(q)$, initialized from q_0
for SamplingIteration $s \leftarrow 1, 2, ..., N$ **do**
  Sample $q_s \leftarrow \tilde{q} + \mathcal{N}(0, I\sigma^2)$
  Solve $q_s^* \leftarrow \underset{q \in C}{\operatorname{argmin}} \quad f(q)$, initialized from q_s
end for
output: $q_1^*, ..., q_N^*$

Note that Algorithm 1 involves only one extra maximization in the beginning, when compared to a strategy such as multi-starts around a nominal posture. In addition to that, any $\sigma^2 > 0$ will suffice to find different directions of descent and hence different critical points. On the other hand, sampling around a nominal posture requires more tuning such as to make σ^2 large enough to promote exploration away from the closest local minimum.

One assumption of this algorithm is that a local maximum exists on a path between local minima. Note that this need not be the case, since multivariable functions in general can have multiple minima and no other critical points[1]. In practice, however, we did not observe such problems in our experiments.

C. Baselines for comparison

We compare our algorithm with common sampling strategies for optimization with multi-starts:

1) Nominal-posture-sampling: $q_s \leftarrow q^{\text{nominal}} + \mathcal{N}(0, I\sigma^2)$
2) Uniform-sampling: $q_s \leftarrow \text{unif}(q_{\text{min}}, q_{\text{max}})$

where σ^2 is a parameter, and q_{min} and q_{max} are joint limits.

D. Implementation

We solve (2) with a trust-region-based Sequential Quadratic Programming (SQP) algorithm, TrajOpt [1], which is openly available[2]. We use Gurobi as the QP solver.

IV. RESULTS

A. Teleoperation application: qualitative evaluation of postures

We first evaluate our algorithm qualitatively, by checking whether it can find multiple local minima with qualitatively diverse postures. The application we have in mind is teleoperation, such as software that will automatically suggest alternative low-cost postures for an operator to choose or start editing from. We assume the operator has given a nominal posture which indicates a usual, preferred or reference configuration of the robot. This is to guide the generation procedure, so that postures are still sensible (e.g. standing, not crossing arms or legs, not upside down) and look natural.

We choose a typical cost function, the sum of squared joint torques $f(q) = \sum_i \tau_i^2(q)$, on the WAREC-1 robot model using the nominal posture shown in Figure 2: standing with both arms down.

We first run our algorithm and the baselines on a simple scenario with no obstacles. The constraints in the optimization problem are: joint limits, two (6D) feet pose constraints, no-self-collision and ZMP inside of the support polygon. We use a signed-distance function for collision avoidance computed on the robot's links' meshes using Bullet [13] as implemented in TrajOpt [1]. To make evaluation fair, we run all algorithms for the same amount of time, 30 seconds, and then collect the feasible postures which were generated by each during that interval. The results of nominal-posture-sampling are quite sensitive to the perturbation parameter σ, and so we report results with a value that gave us best results on average: $\sigma = 20$ degrees.

Figure 3 shows 10 of the first solutions obtained by each algorithm on a 30 second budget. The figure shows that results obtained with our algorithm, to which we will call Max-σ-Min for short, are visually more diverse than nominal-posture-sampling. They are visually close to nominal posture (see Figure 2), but the arms are raised and dropped in different ways such as to reduce torque consumption. Remember that static torques are minimum when joint motion around that posture will lead to minimum mass displacement on the direction of gravity (i.e. they will be "relaxed" postures). Nominal-posture-sampling finds visually similar postures, where differences are mainly in yaw rotation of hands, bending the trunk to one side or to the other, etc. Note that the reason why postures are bent down is that there is one very-low-cost posture which is quadruped with all arms vertically pointing down, and the gradient of torques from the nominal posture points in that direction. However, the task we chose is bipedal and the ZMP constraint makes such a stretched quadrupedal posture infeasible (i.e. the COM would go out of the support polygon). Our algorithm also discovers this posture, but is not limited to it. The uniform-sampling algorithm also finds diverse postures since it samples within joint limits. While qualitatively it is difficult to say which postures would be most useful to a teleoperator, our algorithm's are closer to nominal and more natural (they look like stretches and crouches) than uniform-sampling - which looks random.

Another difference between algorithms is the number of postures found within the 30 second budget. Max-σ-Min obtained 28 feasible local minima, nominal-posture-sampling obtained 30, and uniform sampling obtained 10. This shows a clear disadvantage of uniform sampling: that it is difficult for it to find feasible postures on a limited time budget. This is intuitive, since sampling anywhere from the joint

[1] If a single variable function has two local minima then a local maximum must exist. However, that is not the case for multivariable functions [12]. It can be proven by counter-example: $f(x, y) = (x^2 - 1)^2 + (x^2 y - x - 1)^2$ has two local minima and no other critical points.

[2] http://rll.berkeley.edu/trajopt/

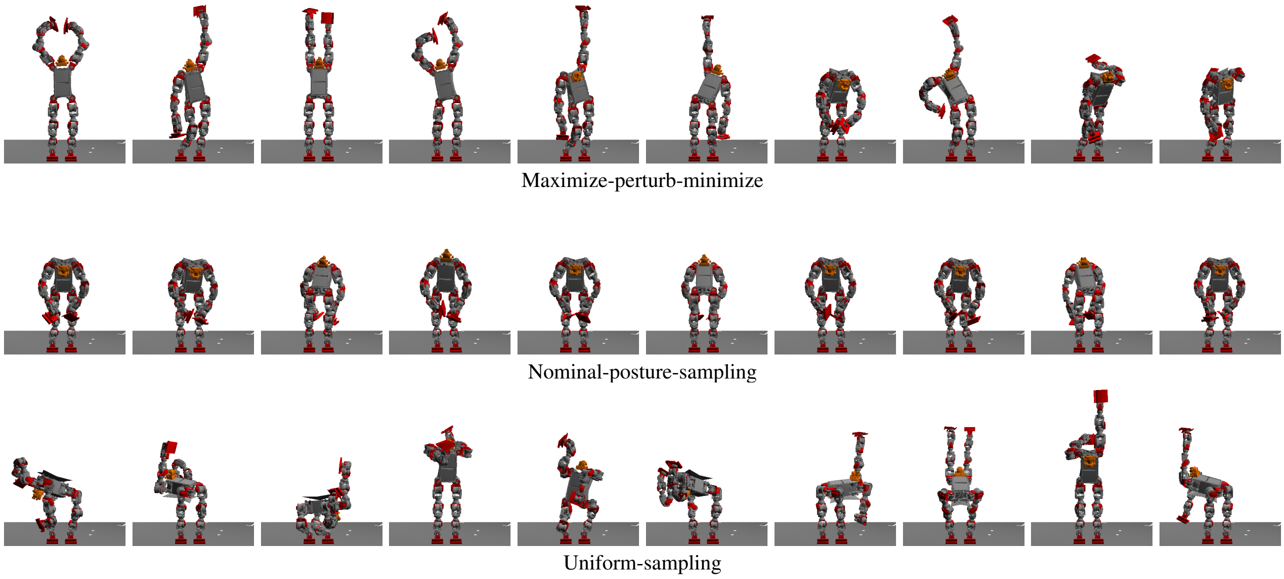

Fig. 3. Ten of the first postures generated by our algorithm and two baselines, on a 30 second time budget. Simple scenario without obstacles. Cost is sum-of-squared-torques, constraints are feet poses, joint limits, no-self-collision, ZMP.

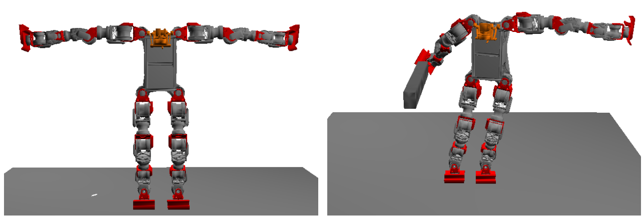

Fig. 4. Postures corresponding to a local maximum of the sum-of-squared-torques. Two scenarios: no obstacles and one obstacle (a bar).

limits can lead to postures which are too far away from the task constraints for SQP to find a feasible solution. For example, sampled postures can be in self-collision, and some constraints may have gradient directions that cancel out. Our algorithm and nominal-posture-sampling both generate a similar amount of postures since they both sample close to nominal (although ours samples from the local maximum closest to nominal). Our algorithm generates two less postures since it has to spend some computational time in the beginning to do one maximization.

We show the result of the maximization in Figure 4: it is a posture with arms wide-open. The figure makes it clear why qualitatively different postures are obtained: small perturbations around the posture may lead one or both arms in different directions which will then be attracted by different minima: some to the top and some to the bottom of the robot.

Next, we consider a scenario with an obstacle on one of the sides of the robot, such that one of the arms will have to avoid it during the maximization stage (and thus a different local maximum is sampled from). We show this maximum on Figure 4 as well. The posture still raises both arms although one is not raised as high as before. We ran our algorithm and baselines on this scenario for a budget of 30 seconds, and we show 10 of the postures on Figure 5.

In this second scenario, our algorithm once again finds qualitatively different postures, when compared to nominal-sampling, although the "both arms up" posture is not found even though it is possible - because it is too far from the local maximum which we use for sampling. Still, the algorithm discovers arm-waiving, crouching, and a posture where the obstacle is "embraced". Nominal-sampling was again low-variance, while uniform-sampling was high variance but often unnatural. Uniform-sampling actually produced similar obstacle-embracing postures which could be of use to a teleoperator. Its main disadvantage is that it unfortunately relies on getting "lucky" to obtain interesting postures. One important observation from this figure is that all methods discovered local minima which embrace the obstacle with the arm. This observation further motivates our intuition that using local minima could be an effective way to suggest robot postures to an operator on a teleoperation interface. The obstacles impose new local minima on the feasible space which change the postures that optimization methods are attracted to. This reminds us of the concept of object affordances [14], which says that objects call or attract us to do certain actions.

B. *Cost minimization*

We now turn to analyzing our algorithm in terms of usefulness in global optimization, which corresponds to picking out the least-cost solution out of all solutions found with multi-starts.

To evaluate algorithm performance we computed the minimum cost over all solutions obtained on a time budget. We varied the maximum allowed computation time from 5 to 90 seconds, and for each choice of computation time we ran 50 experiments with different random number generator

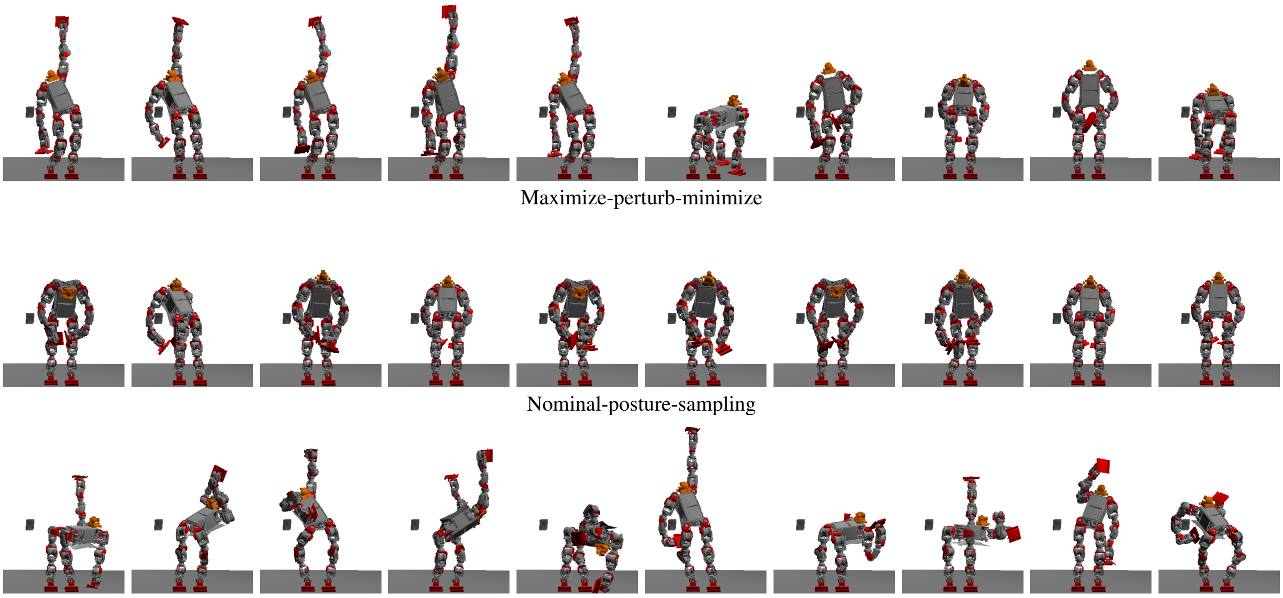

Fig. 5. Ten of the first postures generated by our algorithm and two baselines, on a 30 second time budget. Scenario with one obstacle on the side of the robot. Cost is sum-of-squared-torques, constraints are feet poses, joint limits, no-collision, ZMP.

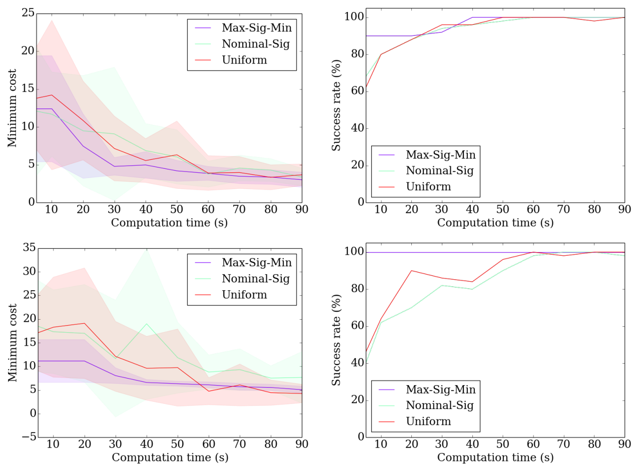

Fig. 6. Average cost and success rate (of finding feasible local minima) as a function of computation time budget. Each point is an average over 50 randomized experiments.

seeds. Figure 6 shows the averaged results on both the simple and obstacle scenario. The graphs show that our algorithm is advantageous for short time budgets. Up to around 50 seconds our algorithm obtains the lowest cost on average. The success rate (number of times that a feasible posture was found) was also higher until around 50 seconds. Surprisingly nominal-posture-sampling also had lower success rates than our algorithm, which shows that a lack of exploration can lead to consistently finding infeasible postures. After around 50 seconds, however, the advantages of our method start to disappear as all algorithms eventually are able to find both feasible and similarly-low-cost minima in the long run. In a context of large computation time budgets, uniform sampling is probably a better option than our algorithm due to its high exploration capabilities - both for global optimization and varied suggestions for teleoperators.

V. CONCLUSIONS

Finding multiple local minima in posture generation has the potential of improving robot performance through cost minimization, as well as interesting applications in robot teleoperation and animation.

In this paper we proposed a method to quickly generate postures lying on local minima of a cost function. The method is simple yet efficient in terms of variability of obtained postures and lowest obtained cost. It consists on first maximizing a cost function from a nominal state, and then running the original optimization algorithm with multi-starts around the local maximum. It relies on the intuitive idea that around a local maximum, even small perturbations can lead to gradients with very different directions. The postures generated are diverse yet still qualitatively close to given nominal postures. Quantitatively, the method obtains lower-cost postures in shorter computation times than other reasonable baselines such as optimization with multi-starts from samples around a nominal posture, or also from uniform samples within joint limits.

In our experimental results we concluded that our algorithm's advantages are stronger when there are short time constraints for the optimization. The advantages in terms of success rate and minimum cost found disappear as computation time budgets increase to around 1 minute. From then on, sampling from a uniform distribution might still be a better choice for obtaining better local minima or qualitatively diverse postures for teleoperation. However, at least for teleoperation purposes, some other algorithm should be used together with uniformly-sampled-multistarts such as to reduce the amount of postures to show to an operator, for example through clustering.

Interestingly, we found that local minima may make for natural posture suggestions for teleoperation. On a simple scenario without obstacles our algorithm obtained diverse natural looking postures such as waving, crouching, and both-arms-up. When we introduced an object, a wide local minima was created consisting of postures embracing it (which was discovered by all algorithms). This reminded us of the affordance theory [14] of objects which call for certain actions, hence being a potentially interesting tool for automatic suggestion of postures and actions for robot operators.

Two most related fields to this paper are that of global optimization and optimization with metaheuristics. The method we use here, random restarts, has a long history and is widely studied [15], but many other global and heuristic algorithms have been proposed in the literature, from simulated annealing to Monte-Carlo [2], evolutionary methods [16], learning and solution databases [17], etc. Here we focused on one of the most popular in the robotics field, due to its simplicity, low computational time and any-time nature: multi-starts [1], [9]. One interesting direction of research is to compare our fast heuristic with other global algorithms.

Other possible directions include the use of explicit regularizers to keep exploration close to nominal posture(s), and actually integrating these methods into robot interfaces.

REFERENCES

[1] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.

[2] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa, "Chomp: Covariant hamiltonian optimization for motion planning," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1164–1193, 2013.

[3] K. Hashimoto, S. Kimura, N. Sakai, S. Hamamoto, A. Koizumi, X. Sun, T. Matsuzawa, T. Teramachi, Y. Yoshida, A. Imai, K. Kumagai, T. Matsubara, K. Yamaguchi, G. Ma, , and A. Takanishi, "Warec-1 - a four-limbed robot having high locomotion ability with versatility in locomotion styles," in *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, Oct 2017, pp. 172–178.

[4] K. Yamane, J. J. Kuffner, and J. K. Hodgins, "Synthesizing animations of human manipulation tasks," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 532–539, Aug. 2004.

[5] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popović, "Style-based inverse kinematics," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 522–531, Aug. 2004.

[6] D. Holden, J. Saito, and T. Komura, "A deep learning framework for character motion synthesis and editing," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 138:1–138:11, July 2016. [Online]. Available: http://doi.acm.org/10.1145/2897824.2925975

[7] M. Brandao, K. Hashimoto, J. Santos-Victor, and A. Takanishi, "Footstep planning for slippery and slanted terrain using human-inspired models," *IEEE Transactions on Robotics*, vol. 32, no. 4, pp. 868–879, Aug 2016. [Online]. Available: http://www.martimbrandao.com/papers/Brandao2016-tro.pdf

[8] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "Stomp: Stochastic trajectory optimization for motion planning," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 4569–4574.

[9] H. Oleynikova, M. Burri, Z. Taylor, J. Nieto, R. Siegwart, and E. Galceran, "Continuous-time trajectory optimization for online uav replanning," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 5332–5339.

[10] S. M. Lavalle, J. J. Kuffner, and Jr., "Rapidly-exploring random trees: Progress and prospects," in *Algorithmic and Computational Robotics: New Directions*, 2000, pp. 293–308.

[11] K. Hauser and J.-C. Latombe, *Multi-modal Motion Planning in Non-expansive Spaces*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 615–630. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-00312-7˙38

[12] J. Stewart, *Calculus*. Cengage Learning, 2011.

[13] Bullet physics library. [Online]. Available: http://bulletphysics.org/

[14] J. J. Gibson, *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979.

[15] R. Mart, M. G. Resende, and C. C. Ribeiro, "Multi-start methods for combinatorial optimization," *European Journal of Operational Research*, vol. 226, no. 1, pp. 1 – 8, 2013.

[16] T. Salimans, J. Ho, X. Chen, and I. Sutskever, "Evolution strategies as a scalable alternative to reinforcement learning," *arXiv preprint arXiv:1703.03864*, 2017.

[17] K. Hauser, "Learning the problem-optimum map: Analysis and application to global optimization in robotics," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 141–152, Feb 2017.