Spatio-Temporal Motion Retargeting for Quadruped Robots

Taerim Yoon[1], Seungmin Kim[1], Dongho Kang[2], Minsung Ahn[3], Stelian Coros[2] and Sungjoon Choi[1]

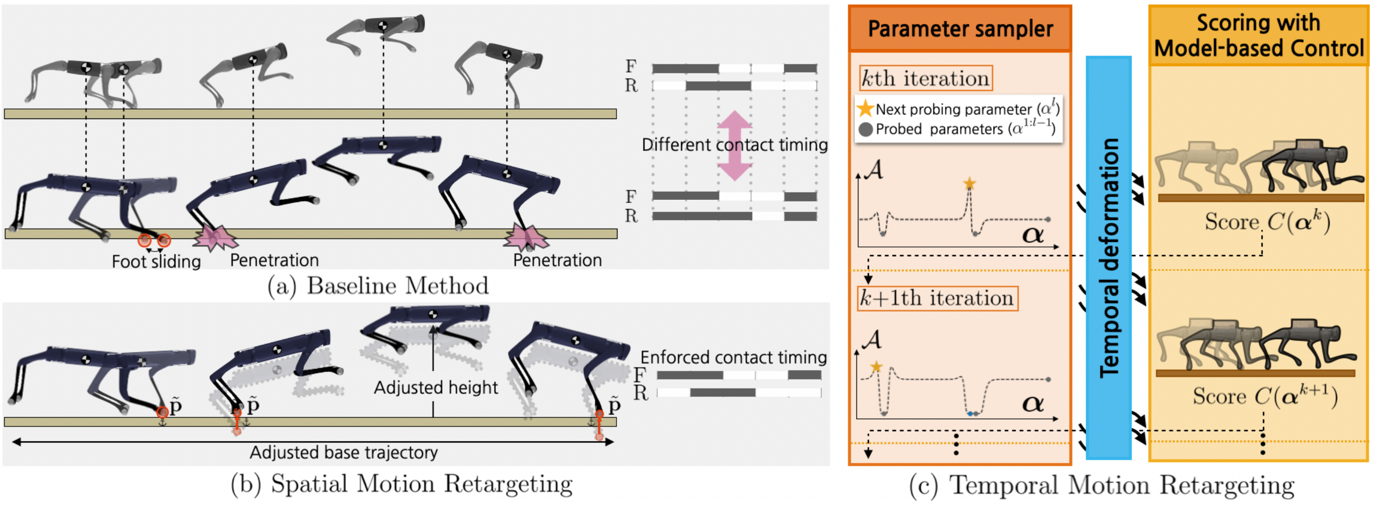

Fig. 1: Illustration of (a) baseline method, (b) spatial motion retargeting (SMR), and (c) temporal motion retargeting (TMR). In our spatio-temporal motion retargeting (STMR) approach, we sequentially conduct SMR and TMR to refine a target motion to address both the kinematic and dynamic properties of the target robotic system.

I. Spatio-Temporal Motion Retargeting

In this paper, we introduce a motion retargeting approach for quadruped robots, which aims to create motion controllers that imitate the agile movements of animals. Our motion retargeting method, namely *spatio-temporal motion retargeting* (STMR), effectively addresses the morphological differences between the source and target systems while guaranteeing the retargeted motion is dynamically feasible on the target system.

This method sequentially deforms motions in both spatial and temporal dimensions as described in the Figure 1. The resulting retargeted motion facilitates the training of a control policy through reinforcement learning (RL), achieving high accuracy in motion tracking by accounting for the kinematic and dynamic constraints of the target system.

A. Notation

We represent the generalized coordinate of the robot with $\mathbf{q}$ and its time derivative with $\dot{\mathbf{q}}$. The state of the robot is defined as $\mathbf{x} = [\mathbf{q}, \dot{\mathbf{q}}]^T$. We specify N keypoints of the robot to track its motion, whose position is given by $\mathbf{p} \in \mathbb{R}^{N\times 3}$. These keypoints, more specifically, include four each of hips, thighs, knees, and feet, totaling $N = 16$. Furthermore, keypoints are defined in each frame, with the position of the jth keypoint in the ith frame expressed as $\mathbf{p}^i_j$ for $j \in \{1, 2, \ldots N\}$, $i \in \{0, 1, \ldots T\}$ where T denotes the total number of frames.

B. Problem Formulation

As keypoint trajectory $\mathbf{p}^{0:T}_{1:N}$ is acquired from an arbitrary quadruped system, it can be physically infeasible for the target robot to track. Additionally, the keypoint trajectory $\mathbf{p}^{0:T}_{1:N}$ may not include the global base movement when the motion is recorded with a hand-held camera. Therefore, STMR seeks to recreate physically feasible whole-body motion by optimizing across both spatial and temporal dimensions.

The objective of the STMR (Spatio-Temporal Motion Retargeting) problem is to establish a mapping $\mathbf{ST}_\alpha : \mathbf{p}^{0:T}_{1:N} \to \mathbf{X}^*$, aiming to produce robot states $\mathbf{X}^*$ that are both kinematically and dynamically feasible. We approach this as a numerical optimization problem, where, given the discrete dynamics f, we aim to find the optimal temporal parameters $\boldsymbol{\alpha}^*$ and control sequence $\mathbf{U}^*$, as specified in Equation (1). The search incorporates additional constraints, notably foot constraints g, to ensure the motion remains kinematically feasible.

$$\min_{\mathbf{U},\alpha\in\mathcal{I}} \mathcal{J} \quad \text{s.t.} \quad g(\mathbf{X}) = 0, \;\; \mathbf{x}^{i+1} = f(\mathbf{x}^i, \mathbf{u}^i), \tag{1}$$

where

$$\mathcal{J} = \sum_{i=0}^{T-1} l(\mathbf{x}^i, \mathbf{u}^i, \mathbf{ST}^i_\alpha(\mathbf{p}^{0:T}_{1:N})) + l_f(\mathbf{x}^T, \mathbf{ST}^T_\alpha(\mathbf{p}^{0:T}_{1:N})).$$

[1]These authors are with the Robot Intelligence Lab in the Department of Artificial Intelligence at Korea University, South Korea.

[2]These authors are with the Computational Robotics Lab in the Department of Computer Science at ETH Zurich, Switzerland.

[3]Minsung Ahn is with the Robotics and Mechanisms Laboratory in the Department of Mechanical and Aerospace Engineering at UCLA, USA.

The authors thank Jin Cheng[2] for his assistance with the experiments.

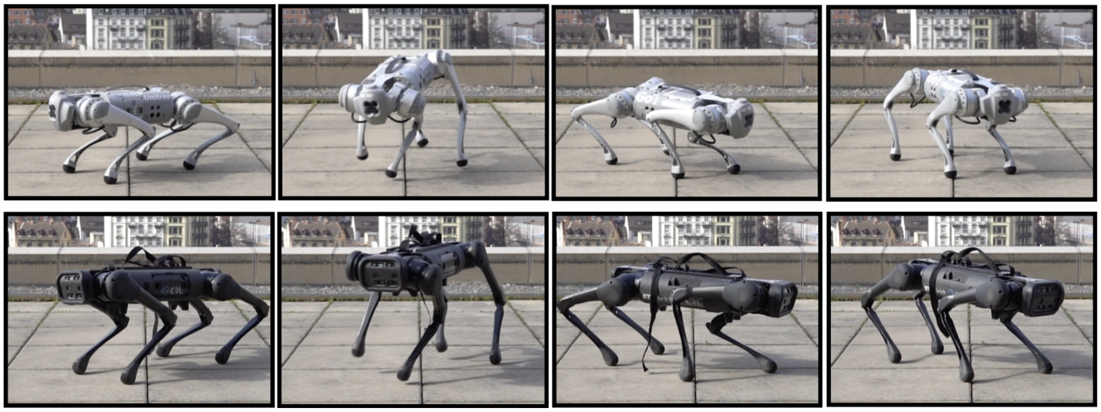

Fig. 2: Deployment of control policy in real-world

It is worth noting that the STMR problem involves actively deforming the target motion rather than mere tracking. Therefore, the motion retargeting function $\mathbf{ST}_{\alpha}(\cdot)$ should be constructed in a way that its resulting motion does not lose the semantic meaning of the original motion. Additionally, due to its nonconvex nature, the standard convex optimization methodologies can not be applied. To address this, we divide this problem into two subproblems: spatial motion retargeting (SMR) and temporal motion retargeting (TMR).

C. Two-stage optimization

Due to the challenges mentioned earlier, we decompose the STMR problem as $\mathbf{ST}_{\alpha}(\cdot) = \mathbf{T}_{\alpha} \circ \mathbf{S}(\cdot)$ where $\mathbf{S}(\cdot)$ represents SMR, and $\mathbf{T}_{\alpha}(\cdot)$ represents TMR process. Adopting this approach, we find these two mappings sequentially through two-stage optimization.

The SMR maps keypoint trajectory $\mathbf{p}_{1:N}^{0:T}$ to kinematically feasible robot states, denoted as $\bar{\mathbf{X}}$ such that $\mathbf{S} : \mathbf{p}_{1:N}^{0:T} \rightarrow \bar{\mathbf{X}}$. In contrast to the baseline method, namely, unit vector method [1] illustrated in Figure 1a, the generated motion is free of foot sliding, adjusts base movement and enforces original contact timing as shown in Figure 1b. Since we focus on kinematic motion, the dynamics f is dropped, and the state-only objective function denoted as $\mathcal{J}_{\mathbf{x}}$ is minimized under foot constraints g, as shown in Equation (2).

$$\bar{\mathbf{X}} = \arg\min_{\mathbf{X}} \mathcal{J}_{\mathbf{x}} \quad \text{s.t.} \quad g(\mathbf{X}) = 0 \tag{2}$$

Following this, a temporal retargeting function, denoted as $\mathbf{T}_{\alpha}(\cdot)$, obtains dynamically feasible states $\mathbf{X}^*$ such that $\mathbf{T}_{\alpha} : \bar{\mathbf{X}} \rightarrow \mathbf{X}^*$. In particular, $\mathbf{T}_{\alpha}$ performs temporal deformation by dividing the motion into S segments with an equal time step size and scaling each with temporal parameters $\boldsymbol{\alpha}$ that lies in interval $\mathcal{I} = [\boldsymbol{\alpha}_{\min}, \boldsymbol{\alpha}_{\max}]$ for $\boldsymbol{\alpha}_{\min}, \boldsymbol{\alpha}_{\max} \in \mathbb{R}^S_{>0}$. In detail, TMR solves the optimization problem from Equation (1) where the objective function is written as Equation (3).

$$\mathcal{J} = \sum_{i=0}^{T-1} l(\mathbf{x}^i, \mathbf{u}^i, \mathbf{T}_{\alpha}^i(\bar{\mathbf{X}})) + l_f(\mathbf{x}^T, \mathbf{T}_{\alpha}^T(\bar{\mathbf{X}})) \tag{3}$$

The TMR problem encompasses a finite-horizon optimal control problem (OCP), which aims to find the optimal control sequence to track the given reference states under dynamics.

Motion	Robot	DeepMimic	STMR (ours)
Hopturn	Go1	6.5(1.2)	**2.1**(0.2)
	A1	8.2(1.5)	**1.6**(0.2)
	Aliengo	5.3(1.9)	**1.7**(0.2)

TABLE I: L1 distance with Dynamic time warping (DTW) [4] normalized by total keypoint trajectory length (%). The values in parentheses indicate the standard deviation.

For example, by ignoring the temporal parameter $\boldsymbol{\alpha}$, the TMR problem reduces to OCP with the goal of tracking the reference states $\bar{\mathbf{X}}$. This property allows us to utilize model-based optimal control (MBOC) [2] as a subprocess of TMR to solve for control sequence $\mathbf{U}^*$, whereas temporal parameter $\boldsymbol{\alpha}$ is searched using Bayesian optimization with Expected Improvement [3] as shown in Figure 1c.

II. Preliminary Results

We validated the effectiveness of our proposed STMR method through a series of both simulation and real-world experiments. For the experiments, we trained control policies using residual learning, which utilizes the motion refined through the STMR process to serve as the base control signal while the control policy provides a feedback control mechanism.

A. Simulational Experiment

In simulation experiments, we evaluated the motion-tracking precision with the baseline method by Peng et al. [5]. We used three distinctive robot models, *Unitree Go1*, *A1*, and *Aliengo*, and measured the normalized motion tracking precision using Dynamic time warping (DTW) [4] divided by total keypoint trajectory length. As summarized in Table I, our method shows enhanced tracking precision by a considerable margin.

B. Real-world Experiment

We showed that the learned policy can effectively produce highly dynamic motion on robot hardware. To bridge the sim-to-real gaps, we randomize controller gains, mass, inertia, friction, and floor restitution and randomly push the robot to change torso velocity during the training. Utilizing our policies, two robots, *Go1* and *Aliengo*, successfully executed a highly dynamic hop-turn motion as depicted in Figure 2.

References

[1] S. Choi and J. Kim, “Towards a natural motion generator: a pipeline to control a humanoid based on motion data,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 4373–4380.

[2] Y. Tassa, T. Erez, and E. Todorov, “Synthesis and stabilization of complex behaviors through online trajectory optimization,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2012, pp. 4906–4913.

[3] J. Snoek, H. Larochelle, and R. P. Adams, “Practical bayesian optimization of machine learning algorithms,” in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012.

[4] P. Senin, “Dynamic time warping algorithm review,” *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA*, vol. 855, no. 1-23, p. 40, 2008.

[5] X. B. Peng, P. Abbeel, S. Levine, and M. Van De Panne, “DeepMimic: example-guided deep reinforcement learning of physics-based character skills,” *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 1–14, Aug. 2018.