EvoGen: Evolutionary Motion Generator

Amir Hossein Rabbani*
La Forge Research Lab
Ubisoft Montreal

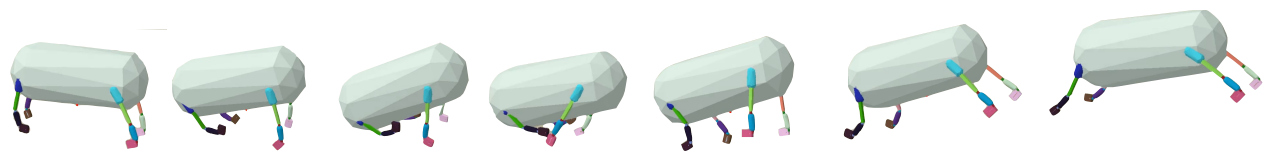

Figure 1: *Quadruped running with 12 DoF. Learning individual joint motions for each leg. A synchronized gait in the front-back leg pairs emerges naturally.*

1 Introduction

We present an automatic system to generate locomotion gaits with different speeds and styles for imaginary terrestrial creatures. Despite the great achievements in this domain, like the nominal evolutionary work by Karl Sims [4], optimal gait for animal locomotion [5], or the recent develeopments in the field of reinforcement learning [3] [6], there are still only a few options in designing an automatic tool to make a creature run given only information on its shape. We propose a gradient free optimization approach that requires no intuition on the locomotion task, which is particularly useful when animating creatures which are extinct or fanciful. Inspired by the periodic nature of muscle dynamics in real animals, at the core of our approach lies a periodic signal generator that spans the space of continuous cyclic curves that are used as desired reference motion trajectories for joint PD controllers to move the character. We then use CMA-ES, a gradient free optimization technique, to search for those signal parameters that minimize a given cost function of traveled distance and speed. We show that finding natural gaits is possible even without relying on a good initial or pre-authored motion, and the resulting locomotions are both visually plausible and complex. To demonstrate the effectiveness of our method, we synthesize dynamic locomotions of a quadruped, a hexaped and an imaginary biped with a rolling ball attached to its end. The current framework only constructs open-loop controllers. Despite producing fairly stable motions we intend to improve the stability of creature motions by also learning balance feedback policies in future work.

2 Methodology

2.1 Motion Generator

The motion of each degree of freedom (DoF) of the character is set to follow a periodic signal with varying magnitude, frequency, phase and impulse shape, covering the whole manifold of all cyclic motion trajectories. It is a time-dependent function of the form

$$f(t) = aN(\beta(\lambda))\left[e^{\beta(\lambda)cos(\omega t+\theta)} + C(\beta(\lambda))\right] \tag{1}$$

where a, ω and θ are the magnitude, angular frequency and phase of the signal cycle, and

$$\beta(\lambda) = tanh^{-1}(\lambda), -1 < \lambda < 1 \tag{2}$$

is the impulse shape control function. When $|\lambda| = 1$ the signal transforms into a periodic unit impulse. In case of $\lim_{\lambda\to 0} f(t)$ it can be shown the signal generator yields a cosine using Taylor expansion (Figure 2 and 3). To impose the mean of the signal to be at zero

$$C(\beta(\lambda)) = -0.5(e^{\beta(\lambda)} + e^{-\beta(\lambda)}) \tag{3}$$

is included as the centering term, and

$$N(\beta(\lambda)) = \frac{2}{e^{\beta(\lambda)} - e^{-\beta(\lambda)}} \tag{4}$$

normalizes the signal magnitude to always remain in $[-1,+1]$. The proposed function is somewhat similar to a duty cycle function except that 1) it is C^k continuous, which makes it suitable for tracking both joint position and velocity, and 2) it can shift its shape from a sinusoid to an impulse using the special parameter λ. In our experiments we found λ to be a key state variable that is highly favoured by the optimization to enable the creature successfully deal with large perturbations when making contacts with the ground.

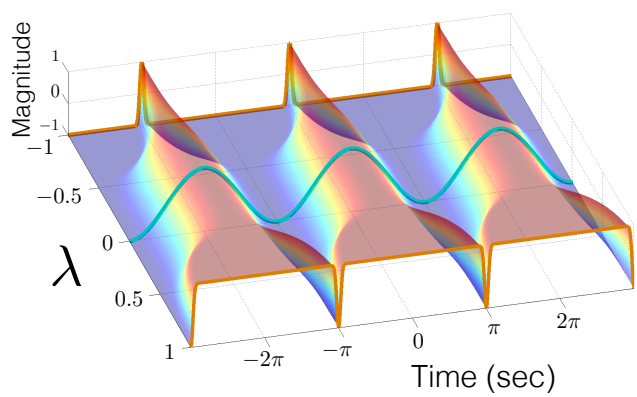

Figure 2: Varying λ values. Note the sinusoid at $\lambda = 0$ and impulses at the λ boundaries.

2.2 Tracking

We use a Proportional Derivative (PD) controller to track the reference time trajectories from Equation 1. Conventional PD controllers often suffer from stability issues, particularly for long multi-body chains, partially due to the absence of the inertia term in their formulation. While in a somewhat tedious process one can tune the gains for individual joints to improve the stability of tracking, we use an alternative method of tuning soft constraint parameters in our physics engine (Bullet). In this approach both the proportional and

*e-mail:amir.hossein-rabbani@ubisoft.com

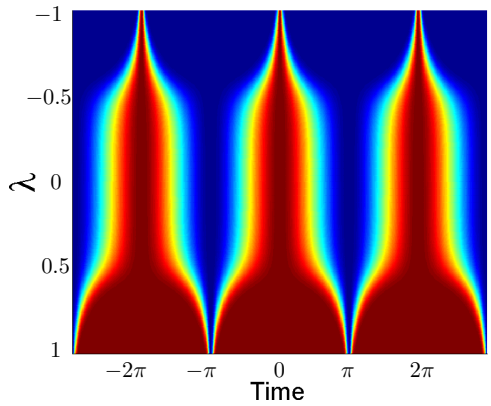

Figure 3: Varying λ vs time. Red regions are positive and blue regions are negative magnitudes, demonstrating symmetry on the opposite sides of λ.

derivative gains are mapped to the error reduction parameter (ERP) and constraint force mixing (CFM) of a motorized joints. This will result in higher stability of the controller with larger simulation time steps as well as using the same stiffness and damping values for all the joints. For further details on tracking with soft constraints the reader is referred to [1].

2.3 Optimization

We use a popular derivative free optimization of Covariant Matrix Adaptation Evolutionary Strategy (CMA-ES) [2]. The goal of the optimization is to find the best set of control parameters in the signal generator function that minimizes a cost function of the form

$$C = w_d \frac{1}{d} + w_j j + w_b b, \tag{5}$$

where d is the distance traveled by the character in the duration of the experiment, j and b are the penalty terms for preventing the character from jumping too high or falling over, and w_d, w_j and w_b are the corresponding weighting terms. In a nutshell, this simple cost function encourages the simulated character to move as fast and as farthest as it can. We are planning to add an energy term to the cost function to achieve more energy efficient locomotion gaits. In a typical optimization scenario we experiment with different sets of available parameters in Equation 1. For example, we can manually set the frequency ω and the phase shift θ but let the optimization find the impulse shape λ and the cycle magnitude a. We can opt for different optimization state variables for each joint DoF, or likewise learn the motion of one leg and mirror the signal onto the rest of the legs with an additional phase shift variable found by the optimization. Note that the choice of $\beta(\lambda)$ as the impulse shape regulator is important because it might pose sensitivity issues during CMA sampling. We already experimented with a power function as well as $sinh^{-1}$ instead of $tanh^{-1}$ in Equation 2. While a power function and $sinh^{-1}$ make better uniform samplings possible, $tanh^{-1}$ makes the equation mathematically more consistent to produce perfect impulse on the boundaries of λ. Other λ regulators that satisfy the signal cycle function will be explored in the progress of this project.

3 Preliminary Results

We run our experiments on a 12-core 2.7 GHz CPU. Typically we learn a motion cycle over a few seconds and test it for about one minute. As demonstrated in the supplementary video, we experimented with a hexaped, a quadruped and an imaginary biped with a rolling body part. While we found using hinge joints sufficient to generate inetersting motions in a low dimensional space, there are no restrictions to use other joint types like a spherical joint. Figure 4 shows the reference time trajectories of one leg of the biped with 3 joints. The same signals are applied to the opposite leg with $\frac{\pi}{2}$ phase shift to impose symmetry. We could have likewise optimized for both legs separately to give a different gait, like what we did with the quadruped. Figure 5 demonstrates the convergence of CMA for the same problem in about 7 wall clock minutes.

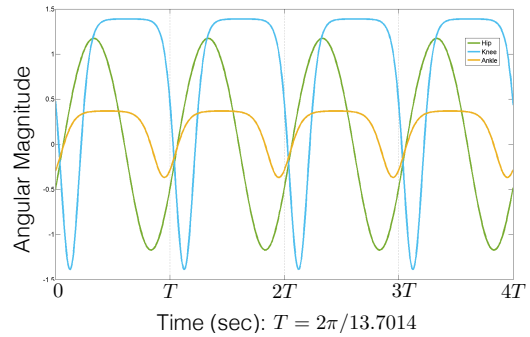

Figure 4: Optimized reference trajectories for each 1-DoF joint of the biped. Green: hip, cyan: knee, orange: ankle.

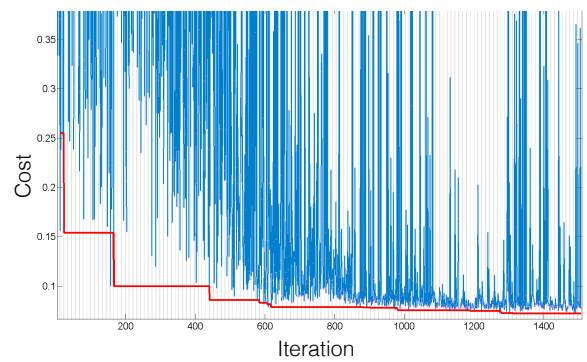

Figure 5: CMA convergence (red) and cost evaluations (blue).

4 Conclusion

We show that with a straightforward setup we can achieve rich and plausible locomotions for imaginary creatures. We continue to explore different signal generators, an energy term in the cost function and the possibility to learn feedback policies to improve the stability of the synthesized motion.

References

[1] E. Catto. Soft constraints, reinventing the spring. Game Developers Conference (GDC), 2011.

[2] N. Hansen. The CMA evolution strategy: A comparing review. In *Towards a New Evolutionary Computation*, volume 192, pages 75–102. Springer Berlin / Heidelberg, 2006.

[3] N. Heess, D. TB, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S. M. A. Eslami, M. A. Riedmiller, and D. Silver. Emergence of locomotion behaviours in rich environments. *CoRR*, abs/1707.02286, 2017.

[4] K. Sims. Evolving virtual creatures. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '94, pages 15–22, New York, NY, USA, 1994. ACM.

[5] K. Wampler and Z. Popović. Optimal gait and form for animal locomotion. *ACM Trans. Graph.*, 28(3):60:1–60:8, July 2009.

[6] W. Yu, G. Turk, and C. K. Liu. Learning symmetry and low-energy locomotion. *CoRR*, abs/1801.08093, 2018.