Monopod robot locomotion using data-driven genetic algorithms

Guang Yang[1], Morteza Azad[2]

***Abstract*— This paper presents a novel implementation of genetic algorithms for locomotion control of robots. The main difference is that, instead of assuming a feedback model for the controller and using genetic algorithm to find the optimal control gains, the proposed algorithm directly generates the optimal joint torques for the robot. This new form of genetic algorithm is called data-driven genetic algorithm (DDGA) in this paper. We implement our proposed control algorithm on a planar hopping robot in simulation. The results show that the robot is able to perform hopping motions on a flat surface as well as climbing the stairs. We also show that, using our proposed controller, the robot can execute a range of highly dynamic manoeuvres such as somersault and backflip.**

I. INTRODUCTION

Legged robots have been one of the significant topics in robotics due to their extra accessibility compared to wheeled robots. In different types of legged robots, the single-legged (monopod) robots are easiest to balance due to its constant hopping motion, which provides extra time to adjust foot placement in the flight phase (when the robot foot leaves the ground). Study of monopod robot control is a promising direction in tackling robot locomotion challenges.

According to Azad [1], monopod robots are categorized into two forms: prismatic-leg hoppers and knee-leg hoppers. The former ones are comprised of prismatic and revolute joints while the later ones only have revolute joints. Although knee-leg hoppers have a simpler form and resemble human leg structures, they are also much more difficult to control. Azad et al. [1] developed an angular momentum based controller for knee-leg hoppers, which can balance a single-joint knee-leg monopod at any unstable balanced configuration while following trajectories. For prismatic-leg hoppers, state-of-art methods are derived from Raiberts feedback controller [4].

While current state-of-the-art controllers have achieved stable hopping and balancing [1] [4], a human-designed controller is often required; specifically, they are model-based controllers. Designing model-based controllers can be tedious since it involves kinematic modeling of the robot. Compared to model-based methods, DDGA does not require a designed controller. This feature allows DDGA to generate actuation signals based on terrain conditions and the presence of obstacles.

*This work was not supported by any organization

[1]Guang Yang is Robotics Msc student in University Of Birmingham, Edgbaston, B15 2TT, United Kingdom `yang@guang.email`

[2]Dr. Morteza Azad is research fellow in University Of Birmingham, Edgbaston, B15 2TT, United Kingdom http://www.mortezaazad.com/

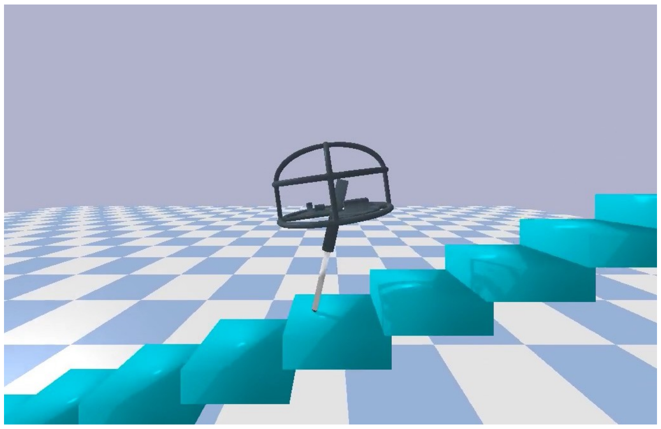

Fig. 1: A prismatic-leg hopper climbing stairs using the proposed data-driven genetic algorithm in the Bullet Physics simulation environment [2].

II. DATA-DRIVEN GENETIC ALGORITHM

DDGA borrows inspiration from natural evolution. We define the whole process in six steps, as shown in Fig. 2.

A. Step 1: Chromosome Initialization

The first step of evolution using DDGA is initialization of chromosomes. Chromosomes are optimization objects in genetic algorithms. We define a full chromosome c ((b) in Fig. 2) as a head to tail connected sub-chromosomes $[c_1, c_2, ..., c_i, ..., c_n]$ ((a) in Fig. 2), where i represents actuator index, n represents index for the last actuator. A sub-chromosome c_i is a series of genes g ((c) in Fig. 2). A gene g carries information τ_i^t, which represents encoded torque (for revolute joint) or force (for prismatic joint) for actuator i at time step t. In the first step, 500 chromosomes are randomly initialized as the first generation. The initialization process is described as following: for each gene g in sub-chromosome, randomly select a continuous value between $[-G_i, G_i]$, where G_i represents the joint torque limit.

B. Step 2: Constrained Crossover

The constrained crossover resembles mating behavior in natural evolution. In DDGA, the crossover operation performs on the sub-chromosome scale instead of the full-chromosome scale. The imposed constraint prevents cross-actuator signal mixing, ensuring that torque values in offspring chromosome do not exceed joint limit (maximum torque or force for the joint).

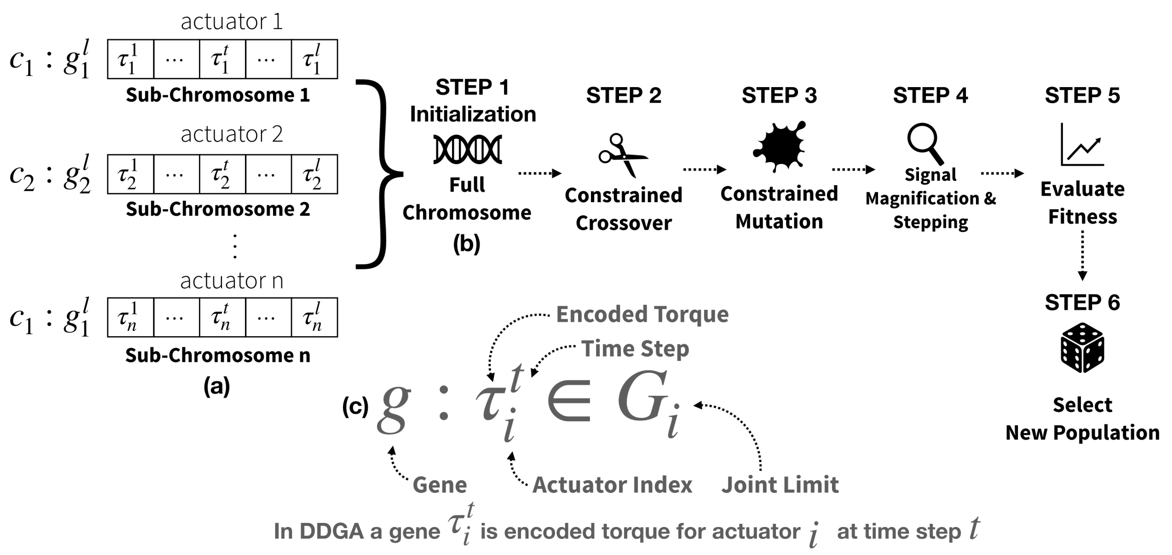

Fig. 2: Six evolution steps in data-driven genetic algorithm.

C. Step 3: Constrained Mutation

The mutation operation alters gene values in chromosomes following a probabilistic distribution. In DDGA, the probability that gene g_i carries value τ_i after mutation is :

$$\mathbf{Pr_m}[\tau_i] = \begin{cases} \dfrac{1}{T_{max}^i - T_{min}^i}, & T_{min}^i \leqslant \tau_i \leqslant T_{max}^i \\ 0, & Otherwise \end{cases} \quad (1)$$

Where τ_i represents the value after mutation, following a uniform distribution whose upper and lower boundary are joint torque level limit T_{min}^i and T_{max}^i.

D. Step 4: Signal Magnification & Stepping

To convert the chromosome into the actuation signal, we introduce a sigmoid magnification function:

$$T_i^t = \frac{T_{Max}}{1 + e^{-\tau_i^t + 5}} - 2 \quad (2)$$

Where τ_i^t represents encoded torque for joint i at time t, T_i^t represents actual torque value, $T_{Max} = 300N$ represents maximum torque in all actuators.

After magnification, the signal is extended by replicating every gene N times, where N represents step size. The signal stepping operation converts continuous actuation signals into discrete pulses and introduces the hopping behavior.

E. Step 5: Evaluation

Evaluation serves as a benchmark for a chromosome's performance. In DDGA, a chromosome is evaluated in Bullet's physics simulation [2] by sending its converted actuation signal to the robot. The locomotion performance is calculated using the following equation:

$$F = \frac{1}{N} \sum_{i=1}^{N} [x + h - k] \quad (3)$$

Where N represents time steps, x and h represents x coordinate and body height in current time step. k is contact punishment and $k = 10$ when the robot body touches the ground, otherwise $k = 0$.

F. Step 6: Selection

The selection step serves as a filter to choose offspring chromosomes. A three-round tournament selection [3] is used in DDGA. Selected offspring enter the next generation and start from step 2 (constrained crossover) in every subsequent generation.

III. Results

We tested DDGA in several environments. On the flat rigid ground, DDGA achieved four stable hops for 5.00 seconds of operation. On the stair climbing test, the robot ascended 1.5m. When a 7-degree slope is added to the environment, DDGA automatically generates somersault movement; backflip is performed when a 20-degree slope is introduced. These results show DDGA is able to generate actuation signals based on terrain conditions and the presence of obstacles.

References

[1] Azad, Morteza. Balancing and hopping motion control algorithms for an under-actuated robot. Diss. Australian National University, 2014.
[2] Coumans, E. (2010). Bullet physics engine. Open Source Software: http://bulletphysics. org, 1, 3.
[3] Fortin, Flix-Antoine, et al. "DEAP: Evolutionary algorithms made easy." Journal of Machine Learning Research 13.Jul (2012): 2171-2175.
[4] Raibert, Marc H. Legged robots that balance. MIT press, 1986.