Improving Quadrupedal Locomotion on Granular Material Using Genetic Algorithm

Jakub Hulas
School of Mechanical Engineering
University of Leeds
Leeds, United Kingdom
jakubhulas@gmail.com

Chengxu Zhou
School of Mechanical Engineering
University of Leeds
Leeds, United Kingdom
C.X.Zhou@leeds.ac.uk

***Abstract*—This paper describes process of generating a stable gait for a quadruped robot on granular material such as sand. To achieve this goal a simulated environment was created, to model the kinematics and dynamics of the robot. Genetic algorithm was used as a gait generation technique. The simulation model of Laikago was built in MATLAB and Simscape. The contact model between the foot and ground, was based on existing results from sand penetration testing. The genetic algorithms optimised the joint space trajectory, to evaluate of the gait performance on granular terrain. The proposed method required to first create a simulation constrained to planar motion to simplify the problem. The 2D gait was then used as the initial guess for the optimisation in 3D simulation and allowed for creation of an acceptable gait. The final gait has demonstrated the effectiveness of the proposed approach for quadruped robot walking on granular terrain.**

***Index Terms*—quadruped robot, genetic algorithm, legged robot, locomotion**

I. Introduction

The legged robot technology has experienced a large development in the last few decades mostly caused by the development of smaller high-torque motors. Legged robots have lower efficiency than the wheeled ones, when moving on flat ground. Their advantage is, however the flexibility when moving across complex environments [1]. Adaptivity of the robot to the environment is crucial for its practical utility. The hardware cannot be tested on all the possible grounds, that is why there is a need for a quick generation method which yields an appropriate gait. One of possible methods is central pattern generation (CPG), Which uses the data from animal walking patters and generates appropriate gaits using deep learning [2]. This approach requires gathering and processing a large normalized data set. The approach chosen in this paper was reinforcement learning with GA. It uses a simulated environment to generate gaits and evaluate and them using a cost function [3].

Laikago is a new affordable quadruped robot, which is lacking the ability of locomotion through complex environments. It can walk up the hills or on the grass but the ability of walking on granular material such as sand and mud has not been demonstrated, even though it is a feature included in the more expensive quadruped robots such as ANYmal. [1].

[1]Unitree.cc, [online], Available at: http://www.unitree.cc/, [Accessed 2 Apr. 2019]

II. Simulation

A. Quadruped Robot Model

The Laikago model is imported into Simscape. It represents relationships between the individual links and joints. For the initial gait generation, the chassis is connected to the world using a planar joint. This allows the robot to move upwards and forwards without the risk of falling to the side. This reduces the degrees of freedom (DoF) from 6 to 3, reducing the complexity of the problem. In the succeeding simulations the planar joint is replaced with a unconstrained joint block to freeing all 6 DoF. This procedure allowed convergence in the first generation. The results of that could be later used as an initial guess for the gait generation of the unconstrained robot, significantly improving the resulting gait.

B. Sand Contact Modeling

A non-linear contact model is chosen for the simulation. As shown in the eq. 2 the normal force is a function of penetration depth and velocity. It is dependent on the stiffness and damping constants which are based on empirical results [4].

$$F_n = kz^n + c\dot{z} \tag{1}$$

An additional feature of the model, is a transition depth, at which the slope of the force-penetration curve is changing its gradient. The transition depth is dependent on the sand properties such as the volume fraction. It occurs because of the way the normal and shear forces are distributed in sand.

To verify the model, before applying it to the whole robot, a simulation consisting of one leg fixed at the base, is created. The leg moves vertically downwards with a constant end-effector velocity and penetrates the sand. When the ground contact is detected and the appropriate reaction forces are applied to the foot. The results of the simulation are shown on the force penetration graph (fig. 2). The loading case clearly shows a non-linear nature of the contact. For this simulation the assumed volume fraction of the sand is 0.6. Based on this value the transition depth is 0.3 m and the stress/depth value is equal to $10^6 Nm^{-3}$ For one foot, the stiffness constant k, is equal to $3927N/m$ [4]. The damping coefficient is set to $250Ns/m$. The exponential constant is set to 1.25 [5].

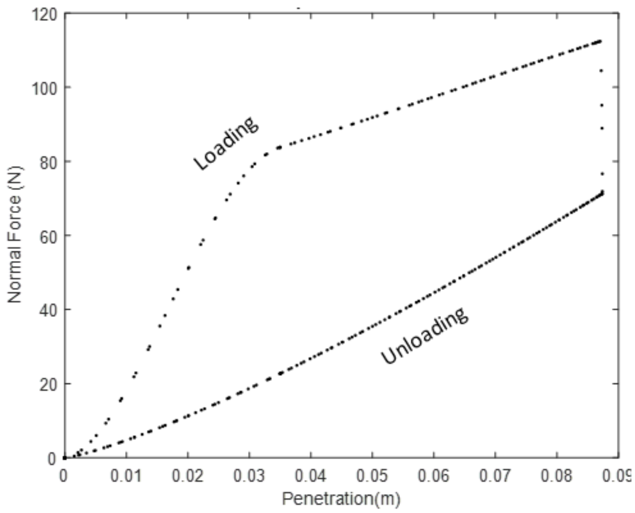

Fig. 1. Sand model force-penetration curve.

III. Gait Generation

GA is a an optimisation method based on natural selection. The algorithm creates a generation of solutions, which consists of the best solutions from the previous generation, a crossover between them, and random mutations. This allows to check a wider range of possibilities compared to other optimisation methods, such as linear regression. It reduces the problem of misjudging local minima for a global minimum.

The simulated robot is actuated by 6 joint space positions during one repeating gait period. The Laikago robot has 3 motors in each leg, which means there are 72 variables to optimise. To simplify the problem, the front and rear pair of legs are actuated using the same trajectory with half a period delay between the right-left pairs. This allowed to reduce the optimisation to 18 variables, significantly increasing the convergence.

A. Cost Function

Cost function(CF) is used to evaluate the performance of the generated gait. The value of the cost function is fed back to the GA. The algorithm uses the value as a variable to optimise. The main role of the CF is to constrain and push the robot gait towards a solution which a human would judge as successful. That's why the optimisation had to be initially repeated to find proper penalties and adjust their weights accordingly. The positive penalties represent what is the goal of the robot gait, in this case the distance and the time walked without falling. The negative once are all the constrains put on the robot. They include: excessive motion of the robot body in the undesired directions, robot walking off a straight path, aggressive joint moves and a dynamic walk (when less than 2 feet are on the ground).

$$cost = -\Pi(+ve)/\Pi(-ve)(2) \tag{2}$$

Eq. 2 indicates the CF is always negative, to ensure finding the minimum.

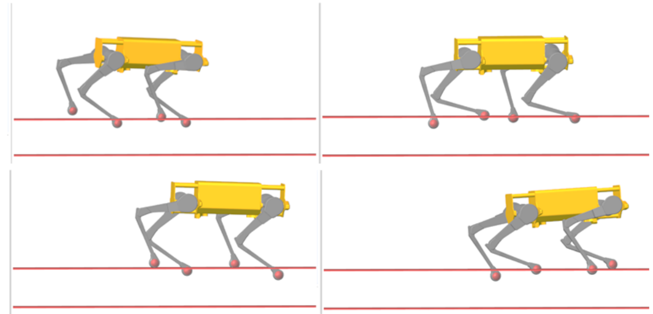

Fig. 2. Snapshots of the final gait movement on sand.

B. Final Gait

The gait was first generated in 2D because the additional constrains applied on the body simplified the overall problem. This allows the optimisation to converge faster to stable results. The 2D trajectory was used as an initial guess for the gait generation in 3D simulation. That yields much higher convergence rate than a pure 3D generation. The velocity of the final gait was $0.4m/s$ which is 30% of the max speed. During walking, the robot is slightly drifting to the side from the straight path. The final gait can be further improved with adjusting the weights of the CF.

IV. Conclusion

The presented GA is shown to generate a stable walking gait for the Laikago robot on granular material. Gait simulation can be used as a first step for generation of new gaits. It could be applied to different robots, or environmental scenarios. In the next steps a reinforced learning algorithm could optimise the gait in an experimental setup, using the simulation gait as a starting point. The contact model should be validated by performing experiments using the hardware.

The procedure of choosing the right reward system for the gait evaluation is time consuming, because it is based on trial and error approach. It might be hard to judge if the given penalty system is right or the success is caused by random mutations. It is difficult to avoid reward hacking of the model and predict the exact outcome. This could be solved with reward modelling. This approach uses a feedback from a human, who chooses a preferred solution. The model then adjust the cost function to satisfy the human choices [6].

References

[1] C. Zhou, X. Wang, Z. Li, D. Caldwell and N. Tsagarakis, "Exploiting the redundancy for humanoid robots to dynamically step over a large obstacle," IEEE/RSJ IROS, pp. 1500–1604, 2015.

[2] D. Bucher, G. Haspel, J. Golowasch and F. Nadim, "Central Pattern Generators," In eLS, John Wiley & Sons, Ltd (Ed.), 2015.

[3] J. A. Carr," An Introduction to Genetic Algorithms," 2014.

[4] A. Peter, "Establishment of Bekker's Model for Predicting the Pressure-Sinkage Behaviour of a Loamy Sand Soil," 2014.

[5] J. Aguilar and D. Goldman, "Robophysical study of jumping dynamics on granular media," Nature Physics, 12(3), 2015, pp. 278–283.

[6] P. Christiano, J. Leike, T. Brown, M. Martic, S. Legg and D. Amodei, "Deep reinforcement learning from human preferences," 2017.