Evolution of Online Update Rules for Robust Locomotion in the SLIP Model

Kathryn Walker[1*] and Helmut Hauser[1]

I. Introduction

Truly autonomous locomotion in machines is still a challenge; especially, mobility over a wide range of different terrains without the loss of stability. On the other hand, biological systems are able to move effectively under a large range of difficult conditions, seemingly without any problem. It is speculated that some of this adaptive motion could be due to the exploitation of morphological features like softness to store and release energy [2]. One model that considers robust locomotion for a wide range of animals is the SLIP model [1]. It combines morphological parameters (stiffness k, resting length l_0, and mass m) with a control parameter (α, attack angle), see Figure 1. For a given fixed set of parameters, it can tolerate small perturbations without losing its periodic locomotion pattern. However, when perturbations are too high the system fails. [3].

In this paper we use evolutionary algorithms to optimize a set of adaption rules to allow the SLIP model to change its angle of attack and spring stiffness and therefore increase the range of stable parameter configurations. Note that we don't genetically encode the stiffness or attack angle directly, instead we evolve the adaptation rules that adjust these parameters. To our knowledge, despite the vast amount of research into increasing the size of the self stabilizing region of the SLIP model no attempt has been made so far to investigate how evolutionary algorithms can assist in exploiting the morphology to improve robustness. In a previous work we systematically tested a small set of adaption rules to obtain the best rule set for offline learning of the SLIP model [5]. Previously our offline learning happens between two episodes where the time from start to when the SLIP model falls over (or the maximum time expires) is defined as an episode. In this paper we extend our approach to more complex rule sets and in addition we also include online learning, where the model is not allowed this failure.

II. Method

This SLIP model is commonly used as a way to model legged locomotion. It was first introduced in [1]. In this work we follow the simulation method from [4]. The leg was simulated in C++ using the Forward Euler method for integration with a time step of $\Delta t = 0.1ms$. The SLIP model was simulated until either the leg fell over (the vertical height of the mass was less than ground level, i.e. $y < 0$) or the maximum time of $t = 100$ seconds ($= 100'000$ simulation time steps) was achieved (the system was considered stable). The resting length of the leg in the simulation was 1 m, the point mass was 80 kg, which is consistent with the values used in literature, see [4].

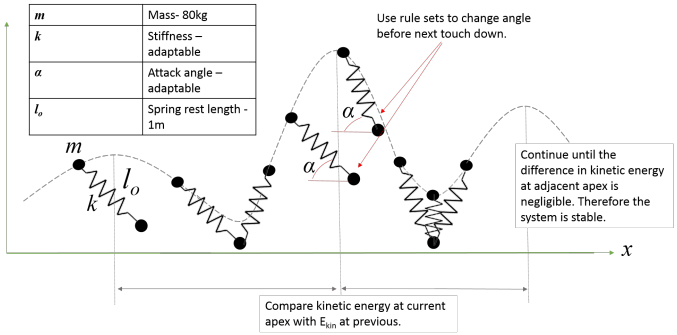

Fig. 1. Diagram showing how on-line learning is incorporated in the traditional SLIP model.The on-line learning is used to change the attack angle α and spring stiffness k on touch down based on kinetic energy change ΔE_{kin} between apexes.

In our previous work, to gain stability the model learned from failed episodes. Whilst acceptable in simulation this would be costly if implemented on a real robot. Therefore instead of learning from the success or failure of a previous episode, our current model uses update rules to learn to adapt based on changes in kinetic energy between consecutive flight apexes. Figure 1 shows how we incorporate on-line learning into the traditional SLIP model. At the apex of each flight phase the kinetic energy is calculated and compared with the value obtained at the previous apex. This is used as our key measurement. The percentage change in kinetic energy, at the ith time step, is given by:

$$\Delta E_{kin} = \frac{E_{kin}(i-1) - E_{kin}(i)}{E_{kin}(i)}. \quad (1)$$

The percentage change in kinetic energy, combined with a particular rule set (described in the section below) are used to dictate how the attack angle (α), or the stiffness (k), or both should change. This change takes place immediately after the apex, so that the parameters are updated before the next touch down.

As previously discussed the way the parameters of the model are to changed from step to step not only depends on the specific rule set that is used. Each rule set describes a

*This work was supported by the EPSRC Centre for Doctoral Training in Future Robotics and Autonomous Systems (FARSCOPE) EP/L015293/1. The research was also partly supported by the Leverhulme Trust Research Grant RPG-2016-345.

[1]Both authors are with the Department of Engineering Mathematics, University of Bristol, BS81UB, UK, and the Bristol Robotics Laboratory, Frenchay Campus BS16 1QY, UK *kw16876@bristol.ac.uk

Generic Rule Set			
Rule #	**Description**	**Value Options**	**Notes**
Rule 1&7	If $\Delta E_{kin} > 0$ should the change in parameter be positive or negative?	Increase/ Decrease	If $\Delta E_{kin} < 0$ the parameter will change in the opposite way to that described.
Rule 2&8	Does the parameter increase by the same amount each time or does it depend on the value of ΔE_{kin} ?	Fixed/Unfixed	If rule 2/3 is fixed, the amount of change is dictated by rule 4/5. If 2/3 is unfixed use the equation: $parameter\ change = \mu\ (\%\Delta E_{kin})$ Where μ is given by rule 4 and $\%\Delta E_{kin}$ is the percentage difference in current and previous kinetic energies.
Rule 3&9	Does the parameter decrease by the same amount each time or does it depend on the value of ΔE_{kin} ?	Fixed/Unfixed	
Rule 4&10	Either the fixed amount by which the parameter should be increased or the variable μ used in the unfixed equation.	Value between 1-0	If the parameter to be updated is stiffness, rules 4 and 5 are multiplied by a factor of 10^4.
Rule 5&11	The fixed amount by which the parameter should be increased.	Value between 0-1	
Rule 6&12	The threshold value – the parameter will only be changed if $\%\Delta E_{kin}$ is above this value	Value between 0-0.2	

TABLE I

DESIGN OF GENERIC RULE SET. ALSO SHOWN IS THE POSSIBLE VALUES EACH INDIVIDUAL RULE COULD TAKE. RULES ARE ADAPTED FROM THOSE USED IN [5].

unique way of using the kinetic energy information to adapt either attack angle or stiffness of the model.

Table I shows the generic design of the rule sets, which adapted from the rule sets used in [5] by using the kinetic energy measurement. If both the stiffness and angle are to be changed a separate rule set for each is required.

If we use both options for rules 1-3 and 10 discrete values for rules 4, 5 and 6, this results in 8000 different rule sets for either changing just k or just α. However, if both α and k are to be changed using different rule sets (12 rules in total) the number of possible combinations rises to 64×10^6. Therefore, to find an optimal solution it is necessary to use a heuristic search algorithm i.e. a genetic algorithm.

The different rule sets are encoded in the genomes of the genetic algorithm. As previously stated, the simulation is run for 10 seconds, or until the SLIP model falls over following the rule set described by the genome of the current individual. For each genome, the simulation was run 1,600 times, each with a different starting configuration of attack angle and stiffness (refer to Figure 2 for the ranges of the starting parameters). Each starting combination is awarded a score depending on the time taken to reach stability (e.g 10 seconds = score 10). We chose the objective function (i.e. the fitness) to be the mean score across all starting combinations. If, at a particular starting combination, stability cannot be reached that combination is given the maximum score of 100.

The population size of the genetic algorithm was 200 and was run for 100 generations. A new generation was formed by combining elitism, multi-point crossover, and mutation for the top 35% of the population. It should be noted that whilst the objective function for the GA was the time for stability to be reached, another consideration was the percentage success rate. The percentage success rate is the number of starting configurations for which stability could be found, independent of the time it took.

III. RESULTS

The genetic algorithm was run three times with three different random starting populations. The best solution found was: [Decrease, Unfixed, Unfixed, 0.33, 0.21, 0.0256, Increase, Unfixed, Fixed, 0.14, 0.45, 0.0054]. This means that the angle always decreases with an increase in kinetic energy and any changes (increases/decreases) are based on the amount of kinetic energy, i.e. not a fixed value. In contrast the stiffness increases by a changing amount, but almost decreases by a fixed value/ This top rule set had an average speed of 69 strides and an overall percentage success rate of 21.9 %. This is compared with a non-adaptable system, which has an average speed of 97 and a percentage success rate of 3%. Figure 2 shows the time taken to recover from unstable parameter combinations when following the best evolved rule set.

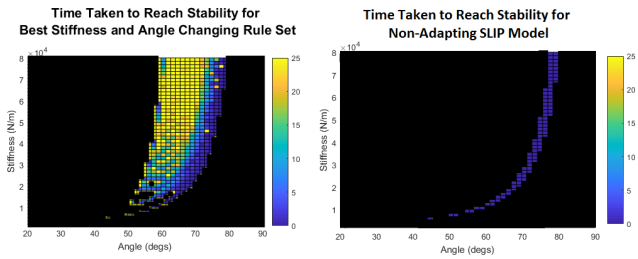

Fig. 2. Diagram shows starting points from which the best learning rule is able to find a stable solution. The different colours show how long it takes the time a stable solution. The black area indicates the starting points where no stability was reached.

These results show that, by following the best update rule, the SLIP model can regain stability for a significantly larger region of stiffness/attack angle combinations. The dark blue region of the graph also represents the area where the non-adaptable system is stable.

REFERENCES

[1] R. Blickhan. The spring-mass model for running and hopping. *Journal of Biomechanics*, 22(11-12):1217–1227, 1989.

[2] R. Pfeifer, M. Lungarella, and F. Iida. Self-organization, embodiment, and biologically inspired robotics. *Science*, 318(5853):1088–1093, 2007.

[3] A. Seyfarth, H. Geyer, M. Günther, and R. Blickhan. A movement criterion for running. *Journal of Biomechanics*, 35(5):649–655, 2002.

[4] A. Seyfarth, H. Geyer, and H. Herr. Swing-leg retraction: a simple control model for stable running. *Journal of Experimental Biology*, 206(15):2547–2555, 2003.

[5] K. Walker and H. Hauser. Adapting stiffness and attack angle through trial and error to increase self-stability in locomotion. *Journal of Biomechanics - accepted*, 2018.