GAIT OPTIMISATION FOR DISTINCT HORSE MODELS USING GRAMMATICAL EVOLUTION

James E. Murphy, Michael O'Neill, and Hamish Carr

University College Dublin
School of Computer Science and Informatics
Belfield, Dublin 4
Ireland
james.murphy@ucd.ie

Abstract: *Motion data is required for realistic animation of physics-based animal models. This data is expensive to acquire for a single animal and in herd situations, the large variation in animal shape and consequent motion increases this expense. We propose a method in which data measured from a single horse can be used to animate horses of different age, breed and conformation. The construction and animation of a physics-based horse is described. Details of an application, which automatically generates horse models of a user-specified age, are also presented. We compare two approaches in which Grammatical Evolution is used to optimise a generated model's motion parameters, to produce realistic motion. In one approach, the constant coefficients of a model's spring-damper based muscle system are optimised prior to the gait optimisation. We contrast this method with a parallel optimisation of both spring-damper constants and gait. The sequential approach was found to be the most successful for gait optimisation.*

Keywords: *Grammatical Evolution, physics-based animation, horse, gait optimisation, spring-damper system*

1 Introduction

Animations of large herds of animals are often required for the entertainment industry. The diversity of shape and size of the animated herd's population must be as found in nature, if the scene is to be considered realistic by a viewer. As such, many different-shaped animal models must be constructed and animated.

Physics-based animations of animal models are highly realistic, given a well constructed model and appropriate motion data. Accurately constructing a physics-based model can be time consuming and requires information pertaining to the model's real-life animal equivalent. For realistic movement, motion data measured from a real-life animal, or specifically optimised or generated for the model, is also required. Assembling this data for a single model is expensive. When multiple models are required, as in a herd situation, the data collection cost can become prohibitive.

We describe a technique for constructing and animating physics-based horse models. To produce the variety of models required for a herd scene, we have developed an application that can generate novel horse models of a user-specified age. The breed and body proportions of the generated models are determined by user supplied animal data files. When a model is constructed, motion data is optimised for that model using a Grammatical Evolution based gait optimisation technique. However, the large range of body shapes and sizes that can be generated is problematic, as each novel model requires a bespoke set of spring-damper constant coefficients for the simulation application's muscle force calculations.

To produce motion, torques are applied about the joints of the model. These torques are calculated as spring-damper forces, according to a pattern specified by input motion data. The use of spring-damper equations for muscle force emulation requires that spring-damper constant coefficients be assigned appropriate values so that the bones of the model move in a smooth manner. This is traditionally a non-trivial problem for this type of physics-based model. A common solution involves a time-consuming manual tuning of the constants, in a trial-and-error fashion. While this may be acceptable when animating a single animal, the diversity of body segment proportion and mass in the automatically generated models demands an automated approach. In this paper, we explore how the spring-damper constant coefficients can be optimised both prior to and during the gait cycle optimisation process.

Details of the physics-based horse model construction and animation, and the technique for generating novel models, can be found in Section 3 and 4 respectively. Section 5 provides an overview of how Grammatical Evolution is used for both gait and spring-damper constant optimisation. The results of the spring-damper constant optimisation experiments are then presented and discussed in Section 6 and 7. The following section describes related work.

2 Related Work

The animation of animals in motion is a well-addressed topic in the computer graphics field. Advances in animal animation and simulation also often come as a by-product of robotics research. As horses are so ubiquitous in our everyday lives, there is a large demand for animations of them in the entertainment industry. The use of measured motion data for horse animations is described in [1][2] while [3][4] details the use of optimisation for stable gait generation.

To produce a realistic physics-based animal, data measured or estimated from the model's real-life animal equivalent must be available. Data pertaining to an animal's anatomy is often available in biological and veterinary publications however, detailed data on equine body segment mass and inertial properties is hard to find with one notable exception in [5]. When data is unavailable, estimation of body attributes is made possible via data presented in studies on growth-rate and intra-breed conformational variations [6][7].

Animal motion data is essential for realism. Joint-angle data plots taken from horses in motion are found in [8]. The high-speed photographs of Muybridge have been a valuable, albeit crude, source of motion data for many animators throughout the last century [9]. Studies of animal motion have, for years, explored why animals transition between gaits at particular velocities. Dynamic similarity theory describes how the characteristics of an animal's gait can be predicted based on its velocity and hip-height [10].

Regardless of the large pool of knowledge on animal locomotion, manual production of stable gaits for physics-based animal models and quadruped robots remains a non-trivial task. Gaits for quadruped robots have been generated using Genetic Algorithms in [11][12]. In [16] Grammatical Evolution is used for the optimisation of horse gait data.

A major factor in any gait optimisation is the gait data representation. The form this takes depends largely on the physics-based model's construction and the technique used to animate it. The following section describes the construction of a horse model, its locomotion system and motion data representation.

3 Physics-Based Horse Model

In physics-based animation, the laws of physics are used to determine the behaviour of a computer constructed scene or model. A physics engine controls the physical simulation of a model through integration of its equations of motion. The resulting animations are realistic and remove the need for an artist to hand-animate the frames of motion. In the following sections, we discuss how a physics-based animal model is constructed and animated.

3.1 Model Construction

We have manually constructed a horse model using data for individual body segments obtained from [5]. This data includes values for mass, centre of mass, density and inertial tensor for 26 body segments of Dutch Warmblood horses. The data is physically measured from six dissected animals and an average of the results is presented. Body segment dimensional data, such as bone length, was also obtained in part from [5]. Other dimensional and positioning information is not provided however. Joint inclination angle values of a horse at stance are manually measured from detailed illustrations in [14]. Neck positioning information was estimated from the high-speed photographs in [9].

The data is used to construct the horse model using the Open Dynamics Engine (ODE) [15]. The model is constructed from multiple interconnected ODE supplied rigid bodies (bones). Each bone corresponds to a segment of the horse and is modelled as a cylinder capped at each end with hemispheres. Each of these bones has attributes such as length, mass, centre of mass and moment of inertia. The physics engine ensures that the bones react in a physically realistic manner to gravity, friction, collisions, applied forces and torques. These bones are connected together by joints, which have constraints imposed upon them ensuring that they do not rotate past some specified threshold or about an impermissible axis.

Forces and torques applied to the bones produce linear and angular accelerations. Forces applied to a single rigid body may propagate throughout the rigid body system allowing for full body locomotion, as described in the following section.

3.2 Model Motion

For intra-limb movement, torques of specic magnitude and direction are applied by motion controllers about each of the joints in a limb to produce bone rotation. The resultant limb movement must be aesthetically realistic and propel the model at the expected velocity. Inter-limb motion timing is also crucial.

The torques required to rotate each of the bones to some target joint-angle, are calculated using a spring-damper equation based on Hooke's law, as shown in Equation 1.

$$F = -kx - bv \tag{1}$$

Where F is the calculated force, k is the spring constant, x is the current displacement from the bone's target joint-angle, b is the damping constant and v is the current angular velocity of the bone. The greater the distance between the current joint-angle and the target angle, the greater the magnitude of torque required to move the bone to its target position. Providing the spring and damper constants are set correctly, the motion controllers can keep the joints moving tightly to the angles suggested by the target data.

Spring and damper constant tuning is a non-trivial task. We have applied Genetic Programming to the task of automatically calculating spring-damper constants based on model mass, model hip-height, bone length, bone mass and position in the hierarchy. Attempts to produce an equation for the entire model, and each joint respectively, were unsuccessful, most probably due to the large amount of variables.

A larger goal of our research involves the retargeting of gaits between different species, as proposed in [13]. Efforts to date have been hindered by the constant need to retune the spring-damper constants for a continuously changing model morphology. A more intuitive and automatic method of defining the constants based on a particular model's attributes would be ideal, however, we currently use a simple numerical optimisation approach, which although time-consuming, produces good results.

Even with perfectly set spring-damper constants, to produce realistic movement, the motion controllers must be given some gait cycle data as input. Without this data, the motion controllers have no way of calculating the magnitude of torque required, and when to apply it. A description of the motion data representation follows.

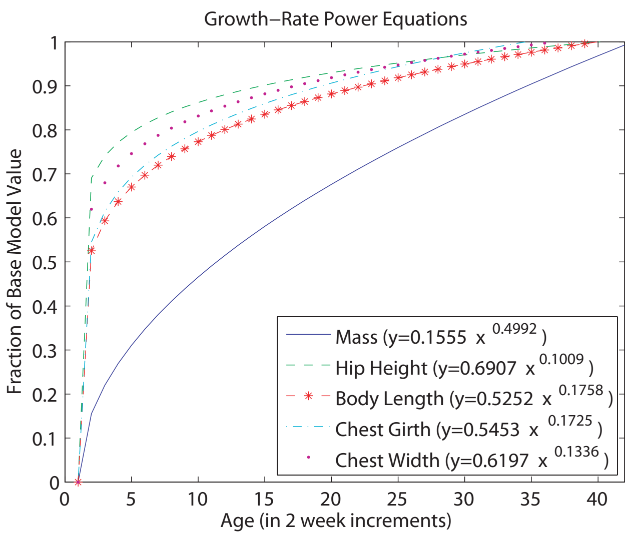
3.3 Motion Data and Representation

The magnitude and direction of the torques mentioned above are calculated from motion data. Motion data measured from a walking horse is published as joint-angle plots in [8]. These plots are discretised and a Fourier analysis is performed on the data, decomposing the joint-angle curves into their consituent sinusoidal functions. Functions which fall below an amplitude threshold are discarded and the remaining are recombined. The motion data for an animal's gait cycle is therefore represented as a set of minimal summations of sinusoidal functions.

Each summation of functions represents the motion of a single joint for a single gait cycle. To produce motion in the model, motion data for a single gait cycle, i.e. a set of sinusoidal functions, must be supplied. If this single gait cycle is optimised for use with a particular model, its periodic nature allows it to be used repeatedly, producing stable locomotion.

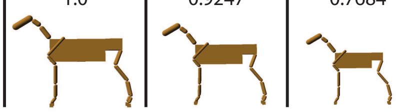
4 Multiple Horse Model Generator

We have developed an application which automatically generates a horse model of user-specified age. The breed of horse generated is determined by user supplied growth-rate data, which defines how different aspects of an animal's body grow in relation to each other, as the animal ages. Accurate anatomical data for a single horse (base model) must also be provided.



Example Model Attributes

	Base Model (84 weeks old)	Model I (40 weeks old)	Model II (10 weeks old)
Mass	1.0	0.6938	0.3473
Hip Height	1.0	0.9345	0.8125
Body Length	1.0	0.8893	0.6970
Chest Girth	1.0	0.9142	0.7198
Chest Width	1.0	0.9247	0.7684



Values indicate fraction of Base Model's actual measurements

Figure 1: Growth-rate equations *(left)* and example model values *(right)*.

The growth-rate data is supplied in the form of power equations pertaining to the rate of increase in body mass, hip-height, body length, chest girth and chest width for a particular breed of horse. The growth equation plots presented in Fig. 1 *(left)* are taken from a study of thoroughbred horses, whose attributes were measured from birth to 588 days old [6]. The data is published as a set of values measured at a consistent range of ages from a large sample of horses. The values are extracted and approximated by a power equation.

The actual model values that are calculated using these growth-rate equations, are fractions of the base horse model data, as can be seen in Fig. 1 *(right)*. The base model data is measured from Warmblood horses [5]. There is an age and breed mismatch between the growth-rate equation horses and our base horse. This is because published studies regarding growth-rates and anatomical data for specific breeds are often unavailable. The small variance between the two sets of data is acceptable for proof of concept.

The variety of models that can be generated is only limited by the data available to a particular user. As well as age, factors such as breed, conformation, sex and deformity could be included in the model generation process. Once a model is generated, gait data must be optimised for use with that model. The following section describes the gait generation process and introduces the problem of spring-damper constant coefficient tuning.

5 Grammatical Evolution Optimisations

To produce stable motion in the generated physics-based models, some motion data must be optimised for that model. Manual optimisation and tuning of motion data is difficult and time consuming. The problem domain is complex and the multivariable nature of the problem motivates our use of Genetic Programming (GP).

Although the low-level joint motions of an animal's gait may be too complex for a human to easily fabricate, there is much domain knowledge available. Muscles tend to relax and contract in a sinusoidal manner leading to smooth joint motion, while skeletal structure and musculature impose limits on a joint's motion. These constraints are easily expressible using grammar-based GP. Grammatical Evolution (GE) is a popular form of grammar-based GP. Easy grammar specification make it an ideal choice for a gait optimisation problem where the level of constraint required is undefined. The following section introduces the gait optimisation technique.

5.1 Gait Optimisation Process

The gait optimisation process involves the concatenation of sinusoidal functions onto seed data, which is in the summation of sinusoidal functions format, as described in Section 3.3. A full explanation of this gait optimisation technique can be found in [16]. It was concluded experimentally that the most efficient and flexible gait optimisation technique involves numerical optimisation of seed data onto which sinusoidal functions are concatenated. This approach ensures that the optimisation remains constrained to the realism of the supplied data, whilst allowing scope to deviate, thus accomodating different morphologies.

The seed data is included in the grammar as a set of triples having amplitude, frequency and phase values. Each of these values is then optimised within a 25% range. Sine and Cosine functions are added or subtracted to the seed data. The functions that may be appended have a set frequency range of 1 to 8Hz and an amplitude range determined by the Fourier analysis mentioned in Section 3.3.

Once a candidate phenotype is generated from the grammar it is passed to the simulation application for evaluation. The model, for which the gait is being optimised, is constructed in the physics engine environment. Using the generated phenotypic gait cycle data, the model moves for a few cycles. The outcome of this simulation run is then scored by a fitness function and that scored is passed back to the GE system.

The fitness function for a gait optimisation is based on energy efficiency, distance travelled, relative stride length and the percentage of each cycle that a hoof was in contact with the ground (duty factor). We can predict the characteristics a gait cycle should exhibit based on dynamic similarity theory, mentioned in Section 2. The score each fitness component achieves for a gait cycle is a function of its measured error from the predicted value and its weight in the function. A lower score implies a better gait cycle however, a score of zero is impossible due to the always positive energy component. An extra fitness component is also included in situations involving the optimisation of spring-damper constant coefficients, as discussed in the following section.

5.2 Spring-Damper (S-D) Constant Optimisation Process

As mentioned in Section 3.2, the tuning of the spring-damper (s-d) constant coefficients is of vital importance. As previously stated, attempts to automate the s-d constant calculations have been unsatisfactory. As manual tuning of the s-d constants is unsuitable on this occasion, a numerical optimisation approach is used.

The general range of s-d values, which will allow a model to move, are linked to the mass of that model however, the model's skeletal structure is also influential. Setting the spring constant too low will not allow the model to apply the necessary torques to move according to the motion data. Setting the spring constant too high can cause the simulation to explode and crash. The risk of an invalid simulation run due to a single extreme value prevents us from running a numerical optimisation on an unconstrained range of values.

Our solution is to determine an appropriate range by running a rapid series of tests to ascertain a range of values that will provide the most stable gait cycle given some sample motion data, albeit it unoptimised for that particular model. A numerical optimisation is then performed within that range.

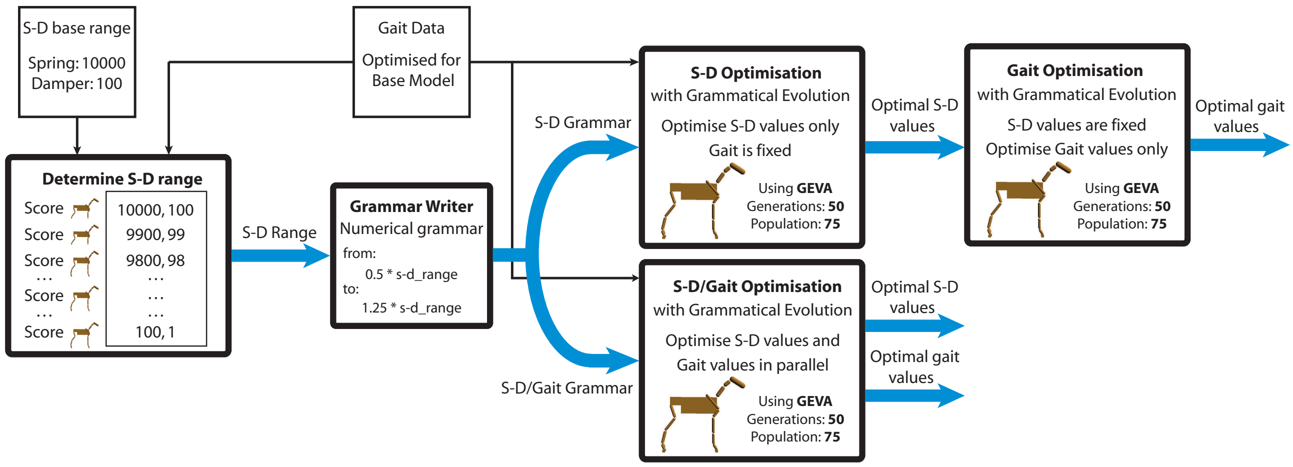


Figure 2: Flow chart of spring-damper (s-d) constant coefficient and gait optimisation process. **Note**: The two distinct approaches investigated in this paper are distinguished by the fork in the flow chart.

The entire optimisation process is illustrated in Fig. 2 . Prior to commencement, an s-d base range, sufficient for base model motion, is experimentally established. As all generated models will be smaller in mass and dimensions, it is assumed that each model's ideal s-d constant range will fall beneath this maximum value.

The process begins with a rapid test to determine what s-d value range is appropriate for the generated model. The gait used to test each value is data pre-optimised for the base model. This gait cycle should not produce perfectly stable motion as the generated models differ in proportion. However, the gait cycle is sufficiently stable enough to provide an indication of performance, given a certain set of constants. At this point the fitness function in the simulation application is weighted to score only based on how well the set of s-d values being tested allow the model's joints to move according to the gait cycle data. The score is a weighted function of the error between the gait cycle data and the motion actually produced.

The generated model is automatically tested with a set resolution of increments of the s-d base range values. Each joint in the model is set with the same double of s-d constants. For a more stable gait, each joint will require its own double of values but at this stage, the uniform values tested here simply indicate an acceptable range. The values which give the most stable gait throughout the testing dictate the range of values used for the s-d constant numerical optimisation, and are passed into the GrammarWriter application. A grammar is automatically written to allow for a numerical optimisation within a set range. In these experiments, the range allowed by the grammar spans from 0.5 up to 1.25 times the passed-in values. This range is divided into 50 even increments and this set of numbers are written into the grammar. Once the numerical optimisation range is established, there are two separate approaches to the optimisation, as detailed in the following section.

Table 1: GE parameters.

Parameter	**Value**
Generations	50
Population	75
Max. wrapping	3
Replacement	generational
Elite size	7
Selection	Tournament (3)
Initialisation	RampedFullGrow
Max. depth	10
Grow prob.	0.5
Crossover prob.	0.9
Crossover point	fixed
Mutation prob.	0.02

6 Spring Constant and Gait Optimisation Experiments

We investigate two approaches to the s-d optimisation problem. In the first, indicated by the upper path after the fork in Fig. 2, we perform a numerical optimisation to establish the s-d constants. We then optimise the gait cycle data. The second experiment on the lower path, optimises the s-d constants and gait in parallel.

Each of the experiments presented in this section are performed using a java-based Grammatical Evolution implementation called GEVA [17]. Each optimisation run uses the parameters presented in Table. 1. For each approach, we present results for two generated models, corresponding to models I and II as shown in Fig. 1.

6.1 Sequential Spring-Damper Constant and Gait Optimisation

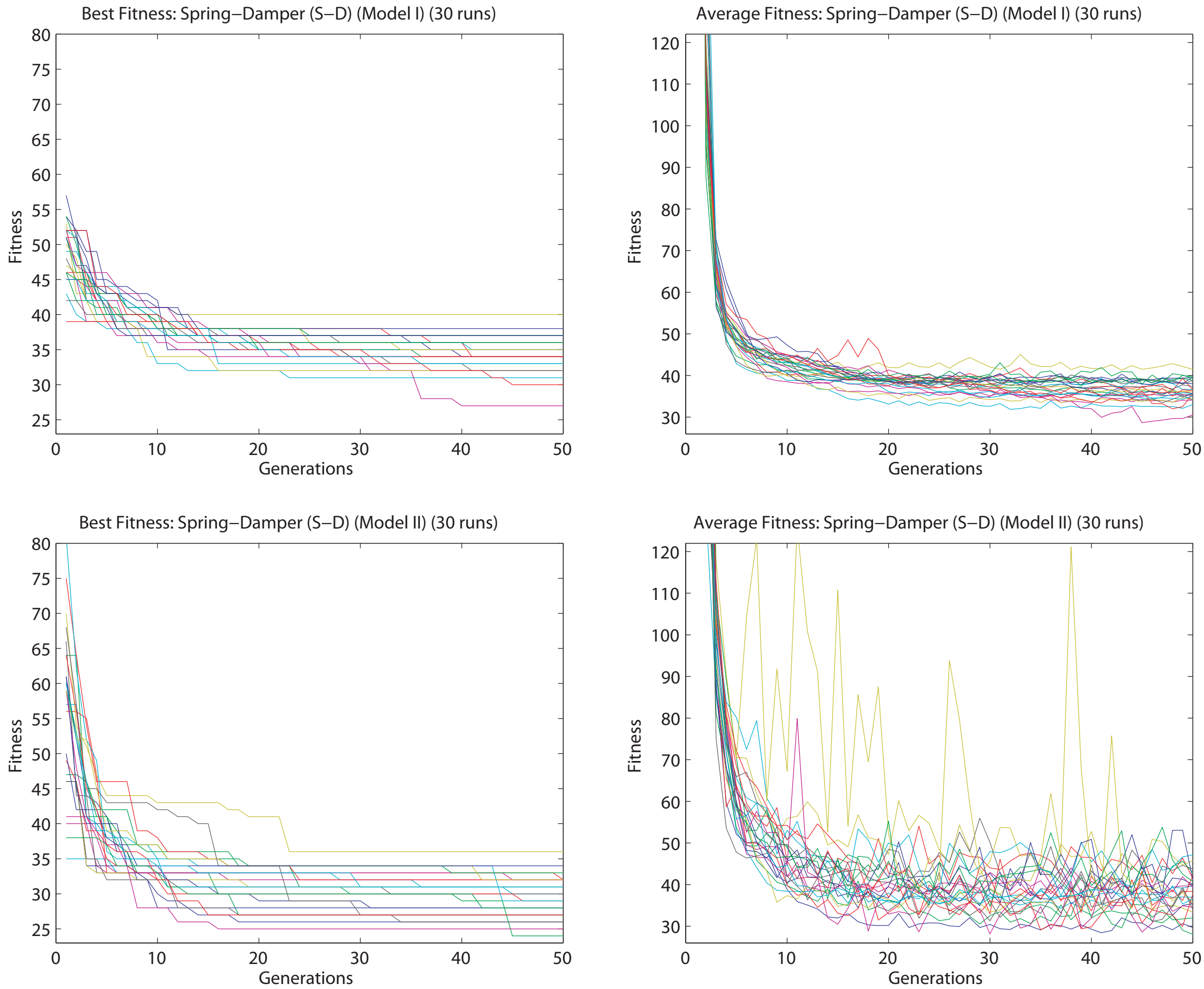


Figure 3: Best fitness and average fitness for spring-damper (s-d) constant optimisation (models I and II, 30 runs).

The s-d constant optimisation process uses the base model-optimised gait cycle data and the fitness function described in Section 5.1. The results are presented in Fig. 3. The plots show variance in fitness across the runs. The variance is most obvious for model II. In the earliest generations, both best and average fitness scores are higher for model II than for model I. This may be because model II is a younger horse. Being younger, it is therefore more different in body proportion to the base model, than the older model I, as implied by the growth-rate curves in Fig. 1. As the gait cycle used for these s-d constant optimisations is optimised for the base model, it is assumed that the more a model differs from the base model, the less stable the gait. Less stability implies a worse fitness score, and this may explain these results.

The average of the best and average fitness scores is shown in Fig. 4. What is somewhat surprising is the best fitness score of model II is significantly better than model I despite it starting off worse. Model II may be scoring highly because the s-d constant optimisation is compensating for the unstable gait through its evolution of s-d constants. Although interesting, this is undesirable as we wish to evolve the best constants for future

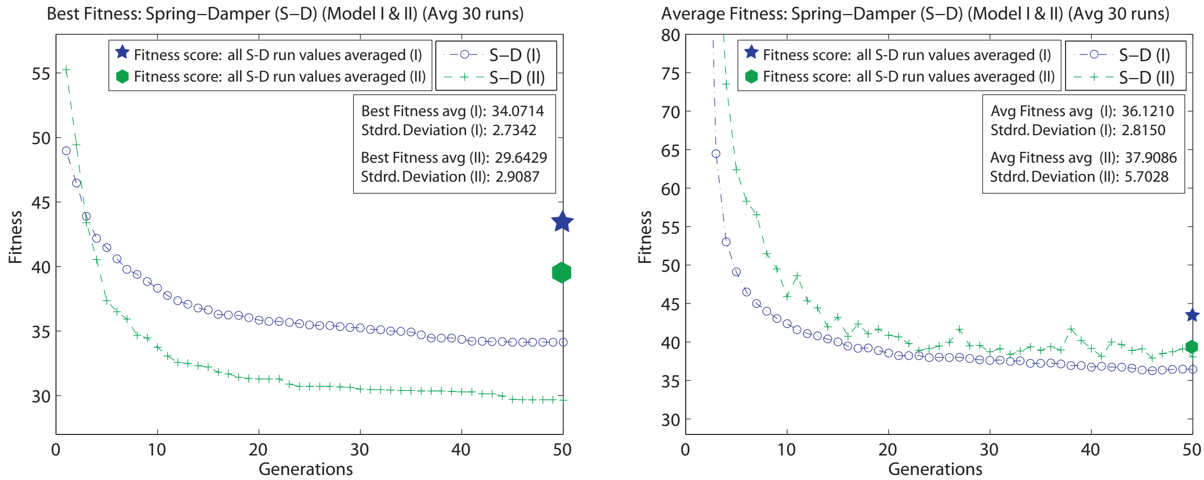


Figure 4: Best fitness and average fitness for spring-damper (s-d) constant optimisation (models I and II, averaged 30 runs). Standard deviation values are included for the averaged best and average fitness scores. The generated s-d values for all 30 runs were averaged and scored for comparison.

gait optimisation, rather than values that provide stable motion from an unsuitable gait cycle. It can be seen from the average fitness plots that model II's average fitness score was worst overall. Model II's average fitness standard deviation is significantly higher than model I's. This high disparity in average score between runs again indicates the high level of instability caused by the base model's inappropriate gait.

Once the s-d constant optimisation is complete, the best performing set of s-d values is returned and the gait optimisation process begins. The best and average fitness results of the gait optimisation are presented in Fig. 5. The results are discussed in Section 7 in relation to the parallel s-d constant and gait optimisation experiment, a brief overview of which follows.

6.2 Parallel Spring-Damper Constant and Gait Optimisation

In this second experiment, the s-d constant values are optimised in parallel with the gait cycle data. Once the best performing s-d constant range is established, the GrammarWriter writes the grammar in exactly the same way as in Section 6.1. This time however, the grammar includes the gait optimisation terms as well. The same value ranges as stated before are used, but instead of having two sequential runs of 50 generations, the optimisation is performed in a single run of 50 generations. The results are presented in Fig. 5.

7 Comparison and Conclusions

A comparison between the sequential and parallel gait optimisation approaches is shown in Fig. 5. It can be seen that the sequential optimisation approach to s-d constant and gait optimisation has won in terms of best fitness scores. The average fitness scores show that both experiments with model II scored worse than their model I equivalent, regardless of optimisation approach. The sequential approach overall had double the computational time than the parallel approach however, further experiments in which the parallel approach was given equivalent time did not show much of an improvement.

The sequential approach performs well however, the fact that the s-d constant is performed with gait cycle data, which may not be stable for a diverse range of models, is a potential problem. This problem would be resolved by using the parallel optimisation approach however, this has performed poorly, perhaps due to the large number of variables in the optimisation. In future, the parallel approach could be improved by imposing certain constraints during the optimisation in an attempt to reduce the number of free variables. The best solution may be to try a combination of both sequential and parallel approaches using both approaches inter-changibly during the optimisation process.

In conclusion, we have successfully developed an application that can automatically produce distinct horse models according to a user's specifications. Using our s-d constant and gait optimisation technique, it is then possible to optimise a single piece of motion data for use with each of these models. Although neither of the approaches to the s-d optimisation problem presented here are completely appropriate to the problem, a future hybrid of the two approaches may be successful.

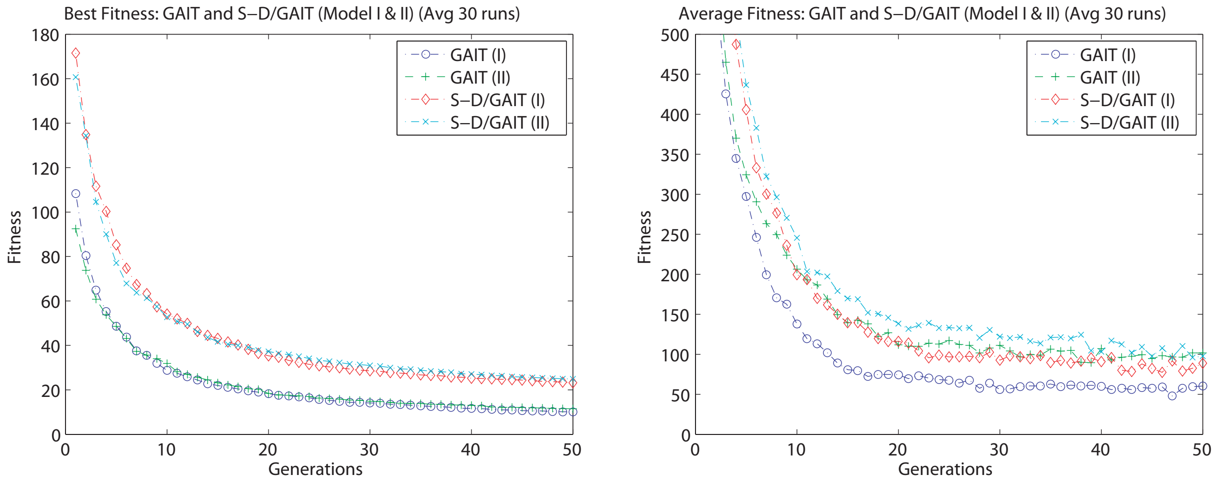


Figure 5: Best fitness and average fitness for sequential gait and parallel spring-damper (s-d) constant and gait optimisation (models I and II, averaged 30 runs).

References

[1] Herr, H. M., McMahon, T. A., A Galloping Horse Model, Journal of Robotics Research, Vol.20, No.1, 2001, pp. 26-37

[2] Marsland, S. A., Lapeer, R. J., Physics-Based Animation of a Trotting Horse in a Virtual Environment, Proc. Information Visualisation, 2005, pp. 398-403

[3] van de Panne, M., Lamouret, A., Guided Optimization for Balanced Locomotion, Computer Animation and Simulation, 1995, pp. 165-177

[4] Hardt, M., von Stryk, O., Increasing Stability in Dynamic Gaits Using Numerical Optimization, Proc. IFAC, 2002, pp. 1636-1641

[5] Buchner, H. H. F., Savelberg, H. H. C. M., Schamhardt, H. C., Barneveld, A., Inertial Properties of Dutch Warmblood Horses, Journal of Biomechanics, Vol.30, No.6, 1997, pp. 653-658

[6] Thompson, K. N., Skeletal Growth Rates of Weanling and Yearling Thoroughbred Horses, Journal of Animal Science, Vol.73, 1995, pp. 2513-2517

[7] Holmstrøm, M., Magnusson, L.-E., Philipsson, J., Variation in Conformation of Swedish Warmblood Horses and Conformational Characteristics of Élite Sport Horses, Equine Veterinary Journal, Vol.22, No.3, 1990, pp. 186-193

[8] Back, W., Clayton, H. M., Equine Locomotion, Harcourt Publishers, 2001

[9] Muybridge, E., Horses and Other Animals in Motion, Dover Publications, 1985

[10] Alexander, R. M., Jayes, A. S., A Dynamic Similarity Hypothesis for the Gaits of Quadrupedal Mammals, Journal of Zoology (London), Vol.201, 1983, pp. 135-152

[11] Kiguchi, K., Kusumoto, Y., Watanabe, K., Izumi, K., Fukuda, T., Energy-Optimal Gait Analysis of Quadruped Robots, Artificial Life and Robotics, Vol.6, 2002, pp. 120-125

[12] Xu, K.,Chen, X., Liu, W., Williams, M., Legged Robot Gait Locus Generation Based on Genetic Algorithms, Proc. International Symposium on Practical Cognitive Agents and Robots, Vol.213, 2006, pp. 51-62

[13] Murphy, J. E., Carr, H., O'Neill, M., Grammatical Evolution for Gait Retargeting, Proc. Theory and Practice of Computer Graphics, 2008, pp. 159-162

[14] Grønberg, P., ABC of the Horse: A Handbook of Equine Anatomy, Biomechanics and Conditioning, PG-Team, 2002

[15] Smith, R., Open Dynamics Engine v0.5 User Guide, 2006

[16] Murphy, J. E., O'Neill, M., Carr, H., Exploring Grammatical Evolution for Horse Gait Optimisation, Proc. EuroGP, 2009, pp. 183-194

[17] GEVA: Grammatical Evolution in Java, *http://ncra.ucd.ie/geva*