

Evolving Aggressive Biomechanical Models with Genetic Programming

Theodoros Theodoridis
Department of CSEE
University of Essex
Wivenhoe Park, Colchester,
CO4 3SQ, UK
ttheod@essex.ac.uk

Panos Theodorakopoulos
Department of Automation
DSP Lab, TEI of Piraeus
Petrou Rally & Thibon
122 44, Athens, Greece
panthead@gmail.com

Huosheng Hu
Department of CSEE
University of Essex
Wivenhoe Park, Colchester,
CO4 3SQ, UK
hhu@essex.ac.uk

Abstract— A repertory of nine biomechanical aggressive activities is investigated in this paper, in our effort to instigate a new paradigm at aggregating descriptive mathematical models with evolutionary, symbolic program representations. Such representations are based on shared biomechanical primitives inspired from *kinematics*, *dynamics*, and *energetics*. Our intension is twofold, initially to study the nature of aggressive biomechanical models and then to classify their physical activities by evolving expression-trees with biomechanical synthesis. The methodology targets on evolving expression programs using the *Gaussian Ground-plan Projection Area* model, to discriminate among three aggressive behaviours and recognise the individual actions involved. For the n -class problem, three programs have been evolved, each for an aggressive behaviour such as the *arm-Launch*, the *legLaunch*, and the *bodyLaunch* behaviour, so that to be able to examine separately the evolvable characteristics induced. The proposed approach has evidently shown strong classification and discrimination performances.

I. INTRODUCTION

Genetic Programming (GP) [1], has introduced a variety of contributions solving various pattern recognition problems, by proposing new paradigms using principles inspired from Darwinian evolution. For the multiclass problem, we introduce an innovative probabilistic fitness function, which stipulates geometric circles to represent Gaussian distributions of program evaluations, in a virtual ground-plan projection (see Fig. 1). The goal of this architecture is to evolve expression program-trees, which output single numeric values denoting the probability of a pattern being recognised, as well as the generation of features reflecting towards a certain pattern. The patterns we investigate regard aggressive physical activities described by the behaviours shown in Fig. 2: $\text{armLaunches} = \{\text{punching}^1 (P), \text{slapping}^2 (S), \text{hammering}^3 (H)\}$, $\text{legLaunches} = \{\text{frontkicking}^4 (FK), \text{sidekicking}^5 (SK), \text{kneeing}^6 (KK)\}$, and $\text{bodyLaunches} = \{\text{pushing}^7 (PS), \text{pulling}^8 (PL), \text{headering}^9 (HD)\}$. Through trial combinations of six biomechanical features, depicted in Table I, GP undertakes to construct a solution discriminating among these patterns, similar to [2]. The derived solution is represented by the generation of three evolutionary models, based on each behavioural type: $\text{armLaunches} \rightarrow \text{upper}_m$, $\text{legLaunches} \rightarrow \text{lower}_m$, and $\text{bodyLaunches} \rightarrow \text{inertial}_m$ model.

The aggressive pattern recognition task is designated for ubiquitous indoor surveillance, utilising intelligent environments and mobile robots.

Previously, in the multiclass problem of binary representations, proposed by [3], a discriminant function is evolved through training examples for each class, and only samples belonging to the same class are credited by strengths of association degrees. In their work, a separate expression-tree is evolved per class, assessing whether unseen test instances belong to the class being tested. Similarly, in [4] the binary problem is decomposed in two binary subproblems letting GP to evolve a solution based on the class being tested. Two dynamic boundary determination approaches based on centre and slotted boundaries, suggested by [5], were used to generate a single numeric output where a class label is given after some transformation. In a probability-based approach proposed by [6], Gaussian distributions have been to construct fitness evaluations assessing the classification performance with two different methods: the distribution distance and the overlapping area, similar to our work.

Outline: Section II presents the aggressive biomechanical analysis. In section III, the evolutionary architecture is discussed. Experimental results are demonstrated in section IV, whereas section V points out conclusions and future work.

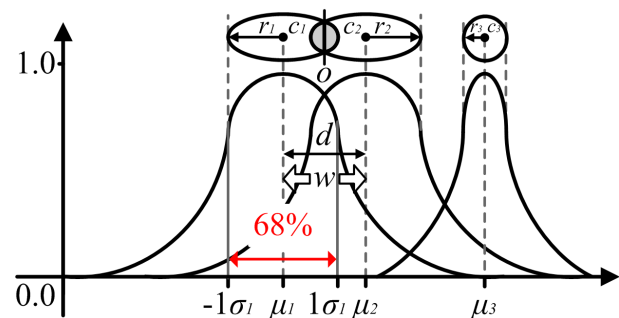


Fig. 1: The ground plan projection showing the Gaussian intersection and the Gaussian extremes with virtual circles.

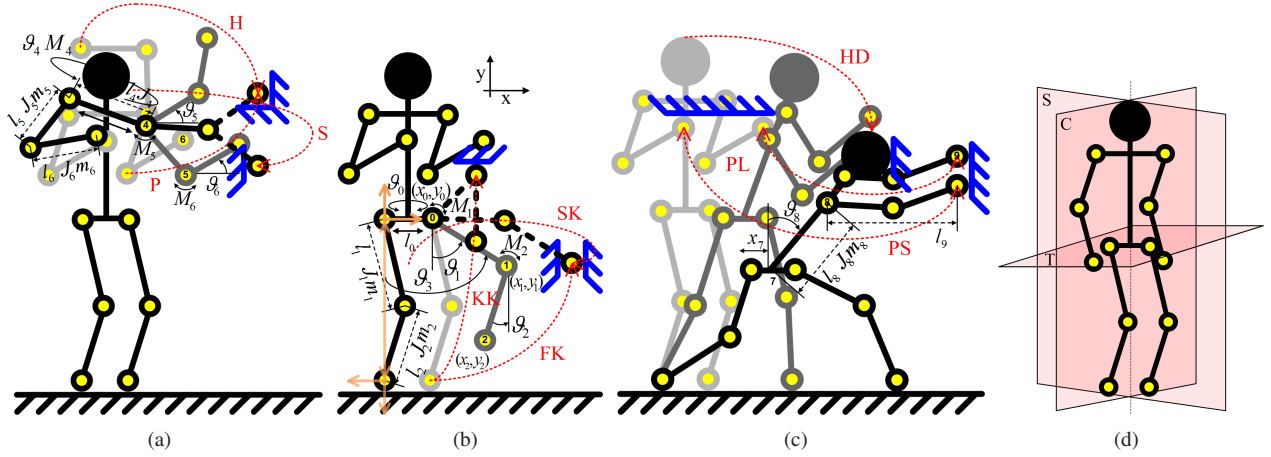


Fig. 2: Biomechanical models. (a) Upper_m, (c) Lower_m, (b) Inertial_m, (d) Action planes (S: sagittal, C: coronal, T: transversal).

II. BIOMECHANICAL AGGRESSIVE MODELLING

A detailed biomechanical analysis [7] of the nine aggressive, physical activities is discussed in this section. Each activity is represented by a set of descriptive mathematical models inspired from kinematics, dynamics, and energetics categories utilising six biomechanical features (see Table I). The aim of this analysis is to describe the expressive nature of aggressive actions, where the derived descriptive formulas are being used partially to construct expression-trees with biomechanical synthesis.

A. Kinematics Analysis

1) *Upper Strikes*: We study the kinematic equations of the position displacements with respect to the three upper-body strikes (see Fig. 2(a)): punching, hammering, and slapping. The first two actions take place on the sagittal plane as Fig. 2(d) depicts, while the third is expressed on the transversal plane. The equations that describe the position and acceleration of the two first actions (Eq. 1) are similar, with only some differences in the constraints.

$$\begin{aligned} x_4 &= \frac{l_4}{2} \sin \theta_4 & y_4 &= 0 & (a) \\ x_5 &= x_4 + l_5 \cos \theta_5 & y_5 &= y_4 + l_5 \sin \theta_5 & (b) \\ x_6 &= x_5 + l_6 \cos \theta_6 & y_6 &= y_5 + l_6 \sin \theta_6 & (c) \end{aligned} \quad (1)$$

where x_4, y_4 , x_5, y_5 and x_6, y_6 define the position xy -coordinates of the shoulder, elbow, and fist respectively. The angle θ_4 is the angle of the line of shoulders as the torso is turning to extend the arm, θ_5 is the vertical shoulder angle, and θ_6 is the elbow angle. Finally, l_4 , l_5 , l_6 are the lengths of shoulder to shoulder distance, shoulder to elbow, and elbow to hand lengths. For the punching case, the fist must cross a horizontal trajectory, satisfying the following restriction:

$$l_5 \sin \theta_5 + l_6 \sin \theta_6 \approx 0 \quad (2)$$

There are some additional constraints that help to distinguish one movement from another: the shoulder itself and the elbow must satisfy the following restrictions for (a) Punching:

$\theta_4, \theta_5, \theta_6 \in \{0, \pi/3\}$. (b) Hammering occurs under different conditions $\theta_4 \in \{0, \pi/3\}$, and $\theta_5, \theta_6 \in \{0, \pi/2\}$. Slapping occurs on the transversal plane and involves mostly a twist of the shoulder line and an extension of the arm (Eq. 3).

$$\begin{aligned} x_4 &= \frac{l_4}{2} \sin \theta_4 & y_4 &= \frac{l_4}{2} \cos \theta_4 & (a) \\ x_6 &= x_4 + (l_6 + l_5) \cos \theta_5 & y_6 &= x_4 + (l_6 + l_5) \sin \theta_5 & (b) \end{aligned} \quad (3)$$

2) *Lower Strikes*: We consider three types of leg induced launches (see Fig. 2(b)): front and knee-kicks (Eq. 4), and side-kicks.

$$\begin{aligned} x_1 &= l_1 \sin \theta_1 & y_1 &= -l_1 \cos \theta_1 & (a) \\ x_2 &= x_1 + l_2 \sin \theta_2 & y_2 &= y_1 - l_2 \cos \theta_2 & (b) \end{aligned} \quad (4)$$

where x_1, x_2, y_1, y_2 give the xy -coordinates of the knee and foot, l_1, l_2 are the lengths of the hip to knee and knee to foot, whereas θ_1 and θ_2 are the hip and knee angles. Front kicks and knee kicks share the same set of kinematic equations as they expressed on the same sagittal plane. However, there is one constraint that helped us to distinguish this movement from the front kick (see Eq. 5):

$$\pi - \theta_1 \approx \theta_2 \quad (5)$$

Sidekicks make use of slightly more parameters within the equations because they are more complex (see Eq. 6).

$$\begin{aligned} x_0 &= \frac{l_0}{2} \sin \theta_0 & y_0 &= 0 & (a) \\ x_1 &= x_0 + l_1 \sin \theta_3 \sin \theta_0 \sin \theta_1 + \theta_0 & y_1 &= y_0 - l_1 \cos \theta_3 \cos \theta_1 & (b) \\ x_3 &= x_1 + l_2 \sin \theta_3 \sin \theta_0 \sin \theta_2 + \theta_0 & y_2 &= y_1 - l_2 \cos \theta_3 \cos \theta_2 & (c) \end{aligned} \quad (6)$$

Here, we have added the hip to hip distance (l_0), and an extra angle for the hip (θ_3).

3) *Inertial Strikes*: Inertial strikes (see Fig. 2(c)), include movements such as pushing, pulling, and head-launches. The motions of pushing and pulling, shown in Eq. 7, appear similar differences only in the direction of the action. Such motions take place in the horizontal axis as there is no significant change in the other axes if we take into account

the simplifications mentioned earlier. Mechanically, the head-strike is an ordinary activity that uses two degrees of freedom originating from the back and head.

$$x_8 = x_7 + l_8 \sin \theta_8 + l_9 \quad y_8 = l_8 \cos \theta_1 + l_9 \quad (7)$$

One can easily obtain speeds and acceleration by differentiating the above equations.

B. Dynamics Analysis

1) *Upper Strikes*: All the upper-body activities are described by very similar equations, which are given by the angular forces of the joint momentums M and inertias J :

$$M_4(t) = J_4 \frac{d^2 \theta_4}{dt^2} + M_5(t) \quad (a)$$

$$M_5(t) = J_5 \frac{d^2 \theta_5}{dt^2} + m_5 g \frac{l_5}{2} \sin \theta_5 + M_4(t) + M_6(t) \quad (b) \quad (8)$$

$$M_6(t) = J_6 \frac{d^2 \theta_6}{dt^2} + m_6 g \frac{l_6}{2} \sin \theta_6 + M_5(t) \quad (c)$$

where friction and elasticity are ignored as they are not important in this context.

2) *Lower Strikes*: The front-kick is a clear movement to analyse as it typically uses one plane of direction. While the force originates from the leg being on the ground (see Eq. 9), in this movement the leg passes through the hip to the other leg cohesively.

$$M_1(t) = J_1 \frac{d^2 \theta_1}{dt^2} + m_1 g \frac{l_1}{2} \cos \theta_1 + M_2(t) \quad (a)$$

$$M_2(t) = J_2 \frac{d^2 \theta_2}{dt^2} + m_2 g \frac{l_2}{2} \cos \theta_2 + M_1(t) \quad (b) \quad (9)$$

3) *Inertial Strikes*: The dynamics of the inertial expressiveness of the actions pushing (+) and pulling (-) is identical, and can be described by Eq. 10, while the dynamics of the headinging action (+) is described by two body axes (lower back and head) as the models showed earlier.

$$F(t) = M \frac{d^2 x_7}{dt^2} + l_8 M_8(t) \quad (a)$$

$$M_8(t) = J_8 \frac{d^2 \theta_8}{dt^2} + \frac{F(t)}{l_8} \quad (b) \quad (10)$$

where mass M equals to the average human weight. The impulse is given by the integral of momentums: $\int_{t_1}^{t_2} M dt$.

C. Energetics Analysis

In the case of punching there are three body parts that move: the torso that rotates on the traversal plane, and the motion of the arm and forearm. The kinetic energy of this system can be then calculated by Eq. 11. All the other energies of the body segments can be derived similarly. In the case of mechanical power we engage the product of momentum and angular velocity: $M \times \dot{\theta}$, for all the body segments involved.

$$T = \frac{1}{2} J_4 \dot{\theta}_4^2 + \frac{1}{2} m_5 (\dot{x}_5^2 + \dot{y}_5^2) + \frac{1}{2} m_6 (\dot{x}_6^2 + \dot{y}_6^2) \quad (11)$$

III. EVOLUTIONARY ALGORITHM, VARIATION OPERATORS, RUN PARAMETERS, AND FITNESS

A. Program Representation Language

Table I presents the strongly-typed programming language used for the construction of probabilistic classifiers. The `List` argument type, of the six biomechanical features, receives the entire timeseries signal from a parameter input. For each primitive a slope sign change method, based on the directional changes of a vector, undertakes to break the signal in n fragments equal to the number of extracted slopes. Thereafter, a mechanical feature evaluation is estimated for every chunk, and the maximum value among all is returned in `double`.

B. Evolutionary Parameters

Our evolutionary algorithm (EA) is an elitist, generational model genetic algorithm, which promotes diversity with panmictic population. The evolutionary run proceeds for 100 generations, and the population size is set to 100 individuals. Evolution halts when all of 100 generations have elapsed for each of the 50 independent runs. Small population with large generation number, employed to allow GP to search for appropriate biomechanical primitives for each aggressive behaviour. Ramped-half-and-half with probability < 0.2 performs random sampling of *full* expression-program trees, otherwise *grow* trees are created. This preference towards grow trees is to avoid bushy ones created by the full, which engage large amounts of memory, and consequently delay the evolutionary runs. With initial depth of 4, the trees are allowed to grow up to medium depth of 12 so that to be interpretable. During fitness assignment, each program is being evaluated with 27 input parameter timeseries.

The EA employs a mutation-based variation scheme to explore the space with probability 0.9, whereas the rest 0.1 is set to perform Koza's sub-tree crossover [1]. A heuristic search scheme is defined by a probabilistically governed

TABLE I: Primitive element language for evolving probabilistic classifier programs with biomechanical synthesis.

Function	Argument(s) type	Returns
acceleration · $\alpha = u(t) - u_0(t) / \Delta t \rightarrow m/s^2$	List	double
displacement · $d_{x,y} = u(t)\Delta t + (1/2\alpha\Delta t^2) \rightarrow m$	List	double
force · $F = m\alpha \rightarrow N$	List	double
impulse · $J = F \cdot t \rightarrow Ns$	List	double
power · $P = F u(t) \rightarrow W$	List	double
kinetic-energy · $T = 1/2m[u^2(t) - u_0^2(t)] \rightarrow J$	List	double
+, -, *, /	double, double	double
Conditional	Argument(s) type	Returns
If-Then-Else	boolean, double, double	double
and, or	boolean, boolean	boolean
>, <	double, double	boolean
Terminal	Value	Type
Constants	50 reals in range {0, 1} step 0.02	double
Parameters	27 timeseries	double

application, based on a mixture of standard mutation variation operators initially presented by Chellapilla [8]. In our scheme, the λ parameter of the multi-mutation architecture of Eq. 12, has been set to 6. Unlike to [8], our scheme induces three groups of operators to (a) enhance diversity: `grow` and `fair`, (b) control bloating: `hoist`, `permut`, and `fair`, and (c) perform smooth variability: `point` and `term`. Eventually, a tiny probability of 0.0001 has been set for reproduction.

```
Offspring = hoist(permut(point(term(fair(grow(Parent)))))) (12)
```

For the selection mechanism, tournament selection has been used with dynamically adaptive selection pressure to promote exploration for the early generations, and exploitation for later ones. Consequently, Eq. 13 uses the factor 0.2 to adjust the range of the individuals being engaged from the first to the last generation. With the parameters given above, the adaptive selection pressure $s_p(g)$ ranges in the interval $\{2, 20\}$ individuals. In addition, negative tournament is also employed by replacing the worst individual with the elitist.

$$s_p(g) = \begin{cases} 2 & \text{if } [(0.2 \frac{g}{G}) \cdot P] < 2 \\ [(0.2 \frac{g}{G}) \cdot P] & \text{otherwise} \end{cases} \quad (13)$$

where g is the current generation, G is the total generation number, and P is the population size.

C. Fitness Function

1) *The Gaussian Ground-plan Projection Area (GGPA) Model:* The GGPA model accounts Gaussian distributions as circles on a ground plan imaginary view, where distribution are being evaluated from the top as depicted in Fig. 1. The representation makes use of geometric circles, where their dimensional properties reflect characteristics from the Gaussians such as the mean μ and the standard deviation σ , to stipulate the centre and the radius of a circle.

Based on this concept, during the evolutionary process each individual is evaluated through a number of training cases for each class; hence, every pattern forms a normal distribution of evaluations with equal samples to the number of training cases. From such Gaussian distribution, the first σ , which corresponds to $\mu \pm 1\sigma \equiv 68\%$, is exploited as it is the area with the most essential samples of the most significant evaluations. The model for each distribution surrogates a geometric circle to represent the 68% of the evaluation samples, in a ground plan projection as Fig. 1 shows. A congruence relation is established between the centre of the circle c and its radius r , equaling to the distribution's μ and σ respectively.

A repulsive weight $w = 1/d$ is employed to equalise the two Gaussians away from each other, thus the closer the distribution means are, the more repulsive the weight becomes. This is analogous to the distance d (Eq. 14), showing the distance of every two distributions (see Fig. 1).

$$d = |\mu_1 - \mu_2| \quad (14)$$

$$x_1 = \frac{d^2 + r_1^2 - r_2^2}{2d} \quad (15)$$

$$x_2 = \frac{d^2 + r_2^2 - r_1^2}{2d} \quad (16)$$

$$a(x, r) = \frac{1}{2} \pi r^2 - x \sqrt{|r^2 - x^2|} - r^2 \cdot \arcsin\left(\frac{x}{r}\right) \quad (17)$$

Eqs. 15 and 16 represent the circle's geometry of the distance between the centre c and the edge of the half plane, where the overlapping area intersects the circle [9]. The area of intersection given by Eq. 17, computes the overlap covered by the two circles. The overlap area is also defined by the shared chord o , with the two edge points coming from the pair of circles denoting the intersection instances (see Fig 1).

There are two extreme cases also shown by Fig. 1. In the first, the two Gaussians are under full overlap with circle c_1 being embedded in c_2 ; this contingency is covered by Eq. 18(b). On the other extreme, Gaussians do not overlap at all as μ_3 shows (according to Eq. 18(a)) with the intersection area being zero. In the case of normal overlap (18(c)), as with $\mu_{1,2}$, the sum of the two intersected areas is returned.

$$area(d) = \begin{cases} 0 & \text{if } d \geq r_1 + r_2 & (a) \\ \pi r_1^2 & \text{if } r_1 < r_2 & (b) \\ \pi r_2^2 & \text{otherwise} & (c) \\ a(x_1, r_1) + a(x_2, r_2) & \text{otherwise} & (c) \end{cases} \quad (18)$$

$$A(d) = \frac{1}{d + area(d)} \quad (19)$$

For the sake of convenience, the area derived by Eq. 18 is normalised as the fraction of Eq. 19 presents, which is a distance-based metric.

2) *Binomial Fitness Function of Gaussians:* Assuming a binary case used to determine the overlapping/intersection area as a distance metric between the classes i and j , similar to [6], [4], by correlating the μ with c and the σ with r . In multiclass pattern classification, the fitness function is determined by considering the distribution distance between every two classes. For the N -class problem, there are $C_N^2 = N!/2!(N-2)!$ class combinations.

$$fitness = \frac{1}{T} \sum_{i=1}^T \sum_{j=1}^{C_N^2} \frac{1}{1 + A(d)_{ij}} \quad (20)$$

where T is the number of training examples, N is the number of classes, $A(d)_{ij}$ is the distance of the intersection area for the index of training examples i and class combinations j ¹.

3) *Ensemble Pattern Classification:* An alternative ensemble weighting method called *statistical voting*, induces 2, 3, and 4 binomial combinations of ten best programs described by Eq. 21. The method utilises the generalised mean to estimate distributions of votes which have similar values. The exponent $p = 10$ is set relatively high so that distributions with more votes to be rewarded. Eventually, the class with the highest density probability (Eq. 22), derived from the ensemble assessment, is designated as the class of the pattern.

$$E_{sv}(x) = \underset{y}{argmax} f(y) = \left[\frac{1}{C} \sum_{i=1}^C \left(\sum_{m=1}^M g(P_m(Pdf = c_i|x), c_i) \right)^p \right]^{1/p} \quad (21)$$

¹The indexes i and j correspond to identifiers 1 and 2 respectively, as shown in Eqs. 14 through 18.

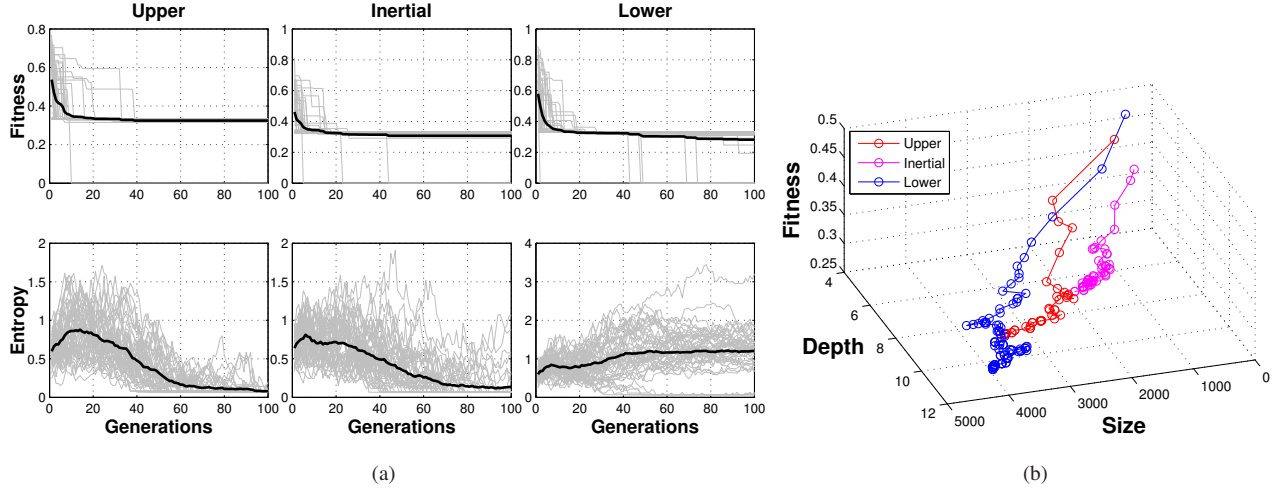


Fig. 3: (a) Fitness and fitness-entropy learning curves of the three models, (b) Bloating paths.

$$Pdf(\mu, \sigma, x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right) \quad (22)$$

where m is the classifier model, tested with M max number of models, P_m is the probability Pdf of class c ($C = \text{max number of classes}$), given a testing instance x derived by Eq. 20 [10]. Function $g(\cdot) = 1$ if $P_m(\cdot) = c$, 0 otherwise.

IV. EXPERIMENTAL RESULTS

The Essex robotic arena was the main experimental hall, where the data collection took place by five subjects who expressed physical activity on a full-body standing bag. Basic safety precautions have been kept, while the subjects' performance has been recorded by the Vicon 3D tracker using 27 channels: 9 markers $\times \{x, y, z\}$, attached on the head and limbs. The recorded data are in timeseries format containing 1,000 samples normalised in $\{0, 1\}$ (Training=60%, Testing=40%).

A. Diversity and Fitness

The fitness graphs for all the three models of Fig. 3(a), presented similar learning behaviour by minimising the training distance-error to 0.4. There have been some outstanding runs where the fitness, after the 40 first generations, reached zero, which is an evidence of overfitting. Moreover, the learning curves seem to stagnate for the next 60 generations due to the small population size. Nonetheless, as it has initially been stated, the purpose of this investigation is to examine the type of biomechanical features evolved by each model, and the deduced recognition accuracy.

A phenotypic diversity measure depicted also by Fig. 3(a), demonstrates that for the upper_m and inertial_m models, the fitness entropy (see Eq. 23) for the first 20 generations showed moderate diversity preserving stable levels. For the next generations though, the diversity of both models had a diminishing direction, opposite to the lower_m model which showed augmented trend; this is an indication of increasing

diversity. The outcome derived from the entropy graphs is that the two first models suffered from stagnation by recycling existing genetic material and trapping to local optima. However, all the models outperformed with similar classification performance, despite the diminishing diversity.

$$-\sum_k p_k \cdot \log p_k \quad (23)$$

where p_k is the proportion of the population P , and k is the occupied partition [11].

A different perspective to our analysis is given by Fig. 3(b), in which the GGPA fitness model shows to guide the evolution without bloating, as depth and fitness progress linearly up to a certain level. Moreover, it is also observed that almost linear direction of the fitness path was followed for all the models, versus the max size (= 12) and depth over 50 runs. From the graph, the lower_m model seems to evolved bushy trees with large expansion, engaging continuously conditional statements, while the average size was below the maximum. The rest models followed similar behaviour with smaller lengths and sizes (> 1000 primitives). Intuitively, we assess that the power of the energy spent, needed from kick-based activities, was the reason why the lower_m model dominated in size and depth. Hence, bushy programs loaded with primitives was the ultimate solution for the model to perform recognition. An alternative proof for this outcome is given by the upper_m model, which needed less expressive power. Finally, for the inertial_m model the power was minimal due to slow and low frequency activities.

B. Evolution of Primitives

The average number of primitive biomechanical features evolved by each model is depicted by Fig. 4, which also indicates the standard errors sdv/\sqrt{N} (where N is the generation number) of each mean value. The error bars verify our previously made assumption of the high power needed for

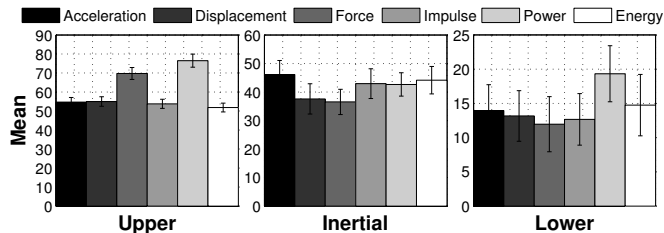


Fig. 4: Error bars of the six biomechanical features, evolved for each behaviour to construct a recognition model.

TABLE II: Performance Measures

Model	Progs	Precision	Recall	Accuracy	F-Measure	D-Power
Upper _m	1	0.60	0.60	0.73	0.60	3.2
	2	0.31	0.47	0.53	0.36	1.6
	3	0.30	0.34	0.53	0.31	1.8
	4	0.20	0.27	0.60	0.22	2.0
Inertial _m	1	0.40	0.40	0.47	0.40	2.4
	2	0.45	0.60	0.67	0.51	2.0
	3	0.53	0.70	0.73	0.60	2.4
	4	0.47	0.60	0.67	0.52	2.4
Lower _m	1	0.53	0.53	0.73	0.53	4.0
	2	0.27	0.33	0.60	0.29	1.8
	3	0.33	0.40	0.67	0.36	3.0
	4	0.35	0.47	0.67	0.39	1.8

the lower_m model, following the upper_m and inertial_m. The power bars for these models exceed the mean usage of all the other features. Furthermore, the force feature for the upper_m and inertial_m models have been used frequently, which seems rational for the inertial actions to require higher force to push or pull an object. Eventually, max usage of features was made by the upper_m model following the inertial_m and the lower_m.

C. Performance Measures and Statistics

Table II illustrates the classification and discrimination performance measures induced by multiple best-programs, designated for each behaviour. Previous research on multiple ensemble models done by [6] and [2], showed that the combination of more than four programs does not increase the classification accuracy. Therefore, with four ensemble models we observed classification accuracy from 60% to 73.3%, whereas for the models upper_m and lower_m, a single individual needed to achieve the highest accuracy as the ensemble method failed. Among all the models where their precisions and f-measures kept low values, the proportional completeness validation of the classification (recall) of the inertial_m was high, with the contribution of three programs. In a closer look at which model performed better, Fig. 5 depicts a comparison test with Anova1. The test for both figures derived with p values equal to 0.57 and 0.64 for the accuracy and discrimination respectively, meaning that all the models achieved similar performance ($df = 11$). However, apart from the fact that almost all the performances appear to be skewed, the interquartile range of the lower_m emphasises on the highest possible performances. This shows relatively symmetrical distributions, with the lower_m dominating.

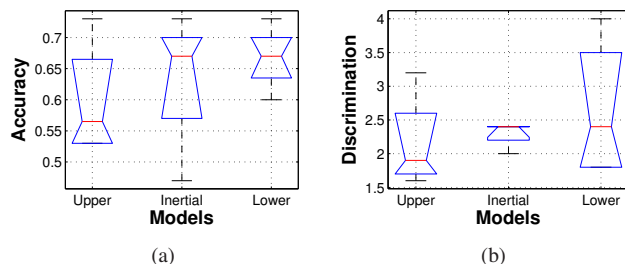


Fig. 5: One way Anova1, testing the models' performance. (a) Classification accuracy, (b) Discrimination power.

V. CONCLUSIONS AND FUTURE WORK

In this paper, an attempt has been made to provide a complete foundation on the study of mathematical biomechanical models, induced by six mechanical features. Such features have been employed to evolve solutions for the n -class pattern recognition of nine aggressive activities. The guidance of the evolutionary process was accomplished by an innovative probabilistic fitness measure, which represents Gaussians of program evaluations with geometric circles, manifested in a ground-plan projection. Satisfactory results have been obtained by the recognition performance, with maximum classification accuracy reaching the 73.3% and discrimination power up to 3.2 (= 80%).

Our proposal to future investigation is to exploit the induction of this new evolutionary paradigm, to evolve programs using biomechanical features related with myoelectrical intensities (EMGs).

REFERENCES

- [1] J.R. Koza. *Genetic Programming: on the programming of computers by means of natural selection*. MIT Press, Cambridge, MA, (1992).
- [2] T. Theodoridis, A. Agapitos, H. Hu, and S. M. Lucas. Mechanical feature attributes for modeling and pattern classification of physical activities. In *ICIA IEEE*, pages 528–533, 2009.
- [3] J. K. Kishore, Lalit M. Patnaik, V. Mani, and V. K. Agrawal. Application of genetic programming for multicategory pattern classification. *IEEE Trans. Evolutionary Computation*, 4(3):242–258, 2000.
- [4] W. D. Smart and M. Zhang. Using genetic programming for multiclass classification by simultaneously solving component binary classification problems. In *EuroGP*, pages 227–239, 2005.
- [5] M. Zhang and W. D. Smart. Multiclass object classification using genetic programming. In *EvoWorkshops*, pages 369–378, 2004.
- [6] W. Smart and Zhang M. Probability based genetic programming for multiclass object classification. Technical report, Computer Science, Victoria University of Wellington, 2004.
- [7] T. J. Walilko, D. C. Viano, and C. A. Bir. Biomechanics of the head for olympic boxer punches to the face. *British Journal of Sports Medicine, ProBiomechanics LLC*, 39:710–719, 2005.
- [8] K. Chellapilla. Evolving computer programs without subtree crossover. *IEEE Trans. on Evolutionary Computation*, 1:209–216, 1997.
- [9] Thompson and Kelvin. Area of intersection: two circles. *Graphics gems, Academic Press Professional, Inc.*, pages 43–46, 1990.
- [10] L. Rokach and O. Maimon. *Data Mining with Decision Trees*, volume 69. World Scientific Publishing Co. Pte. Ltd., 2008.
- [11] E. K. Burke, S. Gustafson, and G. Kendall. Diversity in genetic programming: An analysis of measures and correlation with fitness. *IEEE Transactions on Evolutionary Computation*, 8(1):47–62, 2004.