Evolving free-form stick ski jumpers and their neural control systems

Maciej Komosinski and Jan Polak

Institute of Computing Science, Poznan University of Technology,
Piotrowo 2, 60-965 Poznan, Poland, email: maciej.komosinski@cs.put.poznan.pl

Abstract. This paper concerns evolution of three-dimensional stick agents in a simplified ski-jumping task. Both body morphologies and control systems are optimized. Evolutionary processes are performed in a range of conditions: the air drag and the friction of the ramp varies. Qualitative and quantitative analyses are presented that show how jump distance, jump height, and flight trajectory depend on environmental conditions. Jumping and landing strategies are investigated, and the most interesting evolved behaviors are reported.

1 Introduction

Ski jumping is a sport in which difficult optimization problems need to be solved in order to obtain good results [8]. While this is a truly demanding task for a human and requires perfect coordination, no research has been performed so far on evolution of free-form jumping agents. The task so difficult for a human can be considered a benchmark problem for the optimization algorithms.

In this paper we describe a series of experiments concerning evolution of stick agents (both their three-dimensional morphology and control) in a simplified ski jumping task. The goal (i.e., fitness) is jump length. The length of the jump depends on a number of factors: in-run velocity, the velocity perpendicular to the ramp at the takeoff, aerodynamic forces, and weight of the athlete [10]. We consider a simulation model of this task and evolve 3D structures and coupled control systems for best results.

Related experiments with evolutionary design, active constructs and structures, evolution of control and coordination, evolution of morphologies, and evolutionary and bio-inspired robotics [1], have already proved to be successful in a number of areas [2, 4, 13]. However, no analogous optimization experiments have been conducted so far on the ski-jumping activity. This activity is attractive because it concerns practical problems of design, real-time control, and their simultaneous coevolution. While ski-jumping is difficult for optimization even in a simplified form, the realism and complexity of the model can be gradually increased when optimization algorithms are improved and more computational power is available.

2 Simulation and Optimization Framework

2.1 Mechanical Simulation

Each simulated agent consists of *parts* (points in the 3D space) and *joints* (sticks – rods or cylinders that connect the parts). Since we use the ODE engine [12] to simulate body dynamics within Framsticks [6], the agent's body consists of rigid components that are connected by hinges (each with their own muscle). To simulate air, additional resistance force is applied in each simulation step to each stick:

$$F_{\text{drag}} = \mu \cdot v \cdot |sin\alpha|,$$

where μ is the air drag coefficient, v is stick velocity, and α is the angle between the stick and its velocity vector. The F_{drag} force acts perpendicularly to the direction of the stick.

Note that even though the simulation is highly realistic, it is not our main goal to simulate all forces involved in real-world interactions. The simulation model used in these experiments, albeit complex, misses a number of features – e.g., aerodynamic forces which are very important in real-world ski jumping [9, 10, 11] only take the form of the equation above.

2.2 Control system

Agents are controlled by an evolvable neural network. In general, the network consists of any number of neurons, effectors and sensors; there are no restrictions regarding the topology [6]. The neural network is simulated synchronously and includes sensors such as a gyroscope or touch, effectors (muscles), and processing neurons such as non-linear functions, differentials, or thresholding units. Neural connections have evolvable weights. For the ski-jumping experiment, a special sensor has been introduced: the `PositionX`, whose output value depends on the position of the agent in the world. This sensor allows the agent to be aware of its own location with respect to the ramp. The characteristics of `PositionX` is determined by its two parameters, x_b and x_e, and the output value depends on x which is the coordinate at which this sensor is located in the environment:

$$output_{\text{PositionX}}(x) = \begin{cases} 0 & \text{if} \quad x \leq x_b \\ \dfrac{x - x_b}{x_e - x_b} & \text{if} \quad x_b < x < x_e \\ 1 & \text{if} \quad x \geq x_e \end{cases}$$

$$x_b, x_e, x \in [0, 1].$$

This sensor allows agents to be aware of their location on the ramp. The agent's network may contain many `PositionX` sensors. Note that x_b and x_e can be used to detect any x coordinate, because the optimization process may adjust these parameters. In this way agents could possibly obtain information about the end of the ramp and utilize this information to achieve high fitness values (e.g., by jumping at the takeoff).

2.3 Genetic Representation and Reconfiguration Operators

The preliminary experiments were performed using two genetic representations: a direct, phenetic encoding and a higher-level recursive encoding. They are defined in the Framsticks environment [6, 5] and are called *f0* and *f1*, respectively. Since the higher-level encoding imposes a number of restrictions regarding evolved structures (in particular, no cycles are allowed in the body graph), we ultimately decided to use the low-level phenetic encoding. This ensures that all imaginable 3D stick constructs can be represented.

The crossing over operator was not used in this experiment; it is not particularly efficient when morphologies and control systems that are strongly coupled are evolved. Despite the fact that the *f0* encoding is a textual description of agent's body and brain, the mutation actually operates on phenotypes, not genotypes. The following types of mutation were considered (on each mutation event, one mutation type was randomly picked and performed):

- Modify a part
  - Add a part at the random location (5)
  - Delete a random part (5)
  - Swap two random parts (10)
  - Move a random part randomly in 3D (10)
- Modify a joint
  - Add a joint that connects two random parts (5)
  - Delete a random joint (5)
- Modify a neuron
  - Add a neuron of random type (5)
  - Delete a random neuron (5)
  - Change a random property of a random neuron (10)
- Modify a neural connection
  - Add a connection between two randomly selected neurons (5)
  - Delete a random connection (5)
  - Change a random weight (10)

The numbers in parentheses are relative frequencies of the respective mutation events. For mutations that concern changing a value, the "creep" mutation was employed [3] that adds a random number generated with the Gaussian distribution to the existing value.

2.4 Evaluation Environment

To estimate fitness of each individual, it was put in the simulation environment, initially at the beginning of the ramp at $x = 0$ and $z = 20$ (Figure 1). The ramp had a sinus-shaped profile that ended at $x = 28$ and $z = 10$. The landing area (for $x > 28$) was slightly slanted so agents slid down after landing. The world size was 80×80.

After an agent has been placed in its initial location, it would fall and slide down the ramp. It would pass the end of the ramp, fly over the landing area, and, ultimately, touch the ground. From this moment, 100 additional simulation steps were performed. The fitness value (jump distance) was the lowest x for all collision locations of agent limbs and the ground.

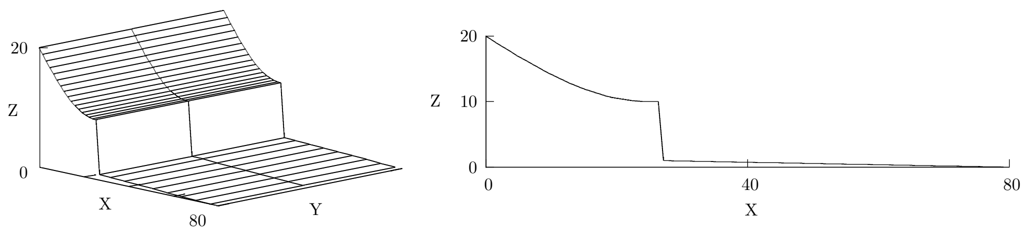

Figure 1. Left: 3D view of the simulated environment. Right: the profile of the take-off ramp and the landing zone.

3 Experiments and Results

In all evolutionary experiments, a population of 200 genotypes was used. The steady-state approach was employed [3]: the best genotype was chosen via tournament selection from three random genotypes, and then it was mutated and evaluated. After evaluation, it was added to the gene pool replacing a randomly selected genotype.

Each evolutionary process consisted of 5000 evaluations. During preliminary experiments, we used ten times longer optimization processes and confirmed that generating 5000 agents is sufficient to observe convergence (at that point, genotypes in the gene pool were usually highly similar and no significant improvements took place).

The following neuron types were enabled: the standard sigmoid neuron that used the weighted sum of inputs, the differentiating neuron, the constant value of 1, the bending and rotating muscles, the pitch (orientation) sensor and the `PositionX` sensor. The initial values of x_b and x_e for the `PositionX` sensor were set to 0.29 and 0.30, respectively, which is ahead of the takeoff point (0.35).

The initial population was seeded with the double-Y-shaped phenotype. Other seeds (like a single stick or a H-shaped body) were also considered, but the double-Y design resulted in a number of diverse body structures and jumping strategies in a short optimization process. The humanoid body used as the initial seed would be quickly simplified and diminished because air drag of the human-shaped body was extremely high compared to its reduced forms (cf. underweight ski jumpers [9]).

3.1 Influence of Air Drag and Ground Friction Coefficients on Jump Length and Height

To study influence of air drag and ground friction on the results of evolution, a number of experiments have been performed for each combination of drag and friction coefficients. There were 6 air drag values (from 0.001 to 0.215) and 10 friction values (from 0.0001 to 0.1) which yields 60 combinations. For each of these, four evolutionary runs have been performed.

The lowest friction value means that the ground is almost entirely slick. The highest value almost stops agents from sliding down the ramp. Both extremes of the friction coefficient make jumping difficult: near-zero friction makes it hard to push back, while high friction reduces speed and promotes rolling (each body part that touches the ramp acts like a ski pole causing rotation of the body).

Figure 2 presents four plots that summarize evolved jump length and jump height (measured as the highest position of the center of mass after takeoff). High values of air drag have helped in long jumps, but resulted in evolution of single-stick passive

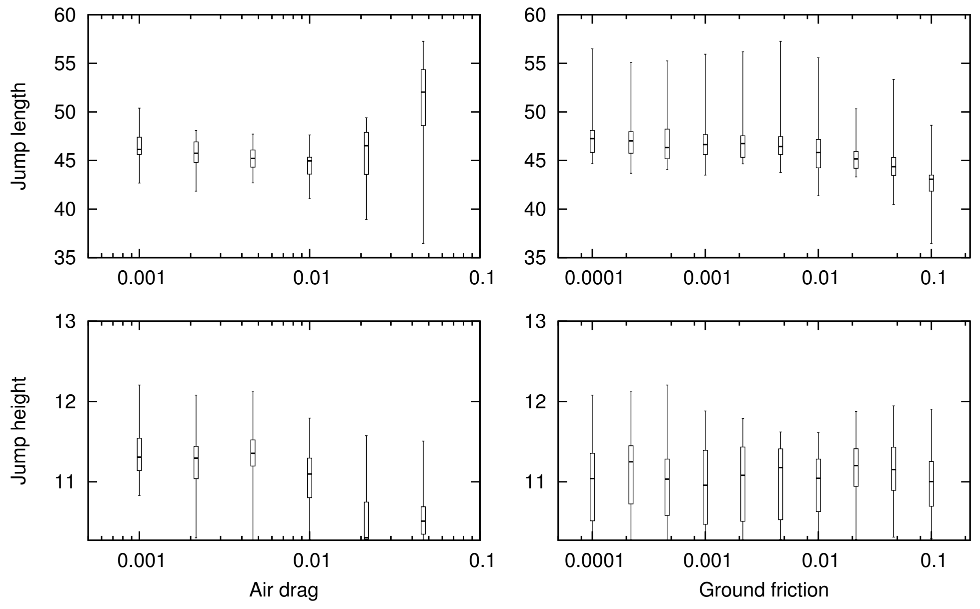

Figure 2. Four box-plots showing jump height and length in function of the air drag and ground friction coefficients.

agents that behaved like arrows in water (high drag acted against gravity). There is no significant influence of ground friction on jump height or jump length except from highest friction values which prevent long jumps by limiting in-run velocity (cf. [8]).

Comparison of flight trajectories was also performed. The distances between all pairs of trajectories were computed as a mean square of their differences in height, in each simulation step. The matrix of distances was then processed using multidimensional scaling [7], and results grouped by drag values are shown on Fig. 3. The distance between points reflects the difference between trajectories. It can be seen that for each air drag value, the evolved trajectories are similar or exhibit specific characteristics. On the other hand, the flight trajectories grouped by ground friction were scattered uniformly (not shown), so friction did not determine shapes of the trajectories.

3.2 Analysis of Evolved Behaviors

Careful analysis of evolved behaviors revealed a great variety of body structures, jumping techniques and ways to achieve high fitness. Most agents would jump at the takeoff, however, non-jumping agents also evolved. Some of the interesting strategies and their characteristics are described on Fig. 4. High friction caused agents to rotate during inrun. Therefore, bodies of most agents in the high friction setting evolved to rotate precisely as much during inrun, to become aligned in parallel to the direction of movement at the takeoff.

Landing techniques were not evaluated in the fitness function. However, the landing

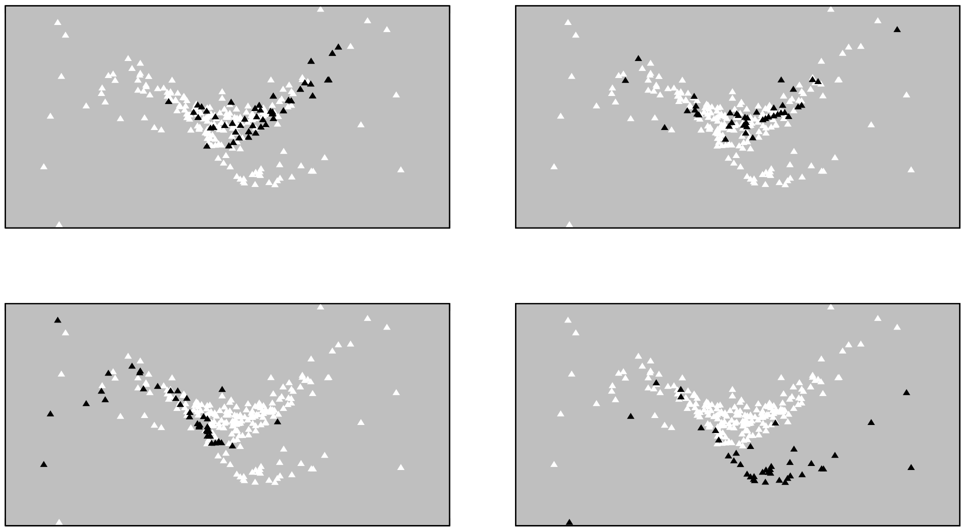

Figure 3. Visualization of differences between trajectories of evolved jumpers. 2D view of multidimensional data; 56% of total variance is preserved. White points represent all flight trajectories; black points are a group of trajectories evolved in a specific air drag conditions (from left to right, top to bottom, air drag coefficient is 0.001, 0.002, 0.021, and 0.046).

was also important as it allowed to gain additional distance. Figure 5 shows the agent that is built of two joints and a hinge. It does not perform a jump at the takeoff. Instead, it folds just before landing in a specific way so that it touches ground a bit later and farther.

Studies of evolved neural networks have shown that agents that used the `PositionX` sensor and connected its output to the muscle, used a very high weight for this neural connection. Therefore, a small increase of the sensory output resulted in a rapid bending of the muscle: agents follow the "don't bend muscle" rule on the inrun, and "apply maximal force to the muscle" near the takeoff.

4 Conclusions

The experiments with evolution of ski jumpers yielded a great number of jumping body structures, coupled neural control systems, and jumping and landing strategies. The most interesting techniques emerged in the low and moderate air drag settings, while high air drag promotes simple arrow-like constructs. On the other hand, ground friction does not affect jumping techniques significantly – they can evolve both in low and high friction environments. In the non-extreme conditions, each evolutionary process can produce a different jumping strategy, which suggests that there are many locally optimal, diverse body shapes and corresponding control systems.

The simulation model employed in these experiments does not fully reflect reality; a more accurate simulation engine might be used if needed. This would substantially increase simulation time, but the experimental framework could be used as is. Employing

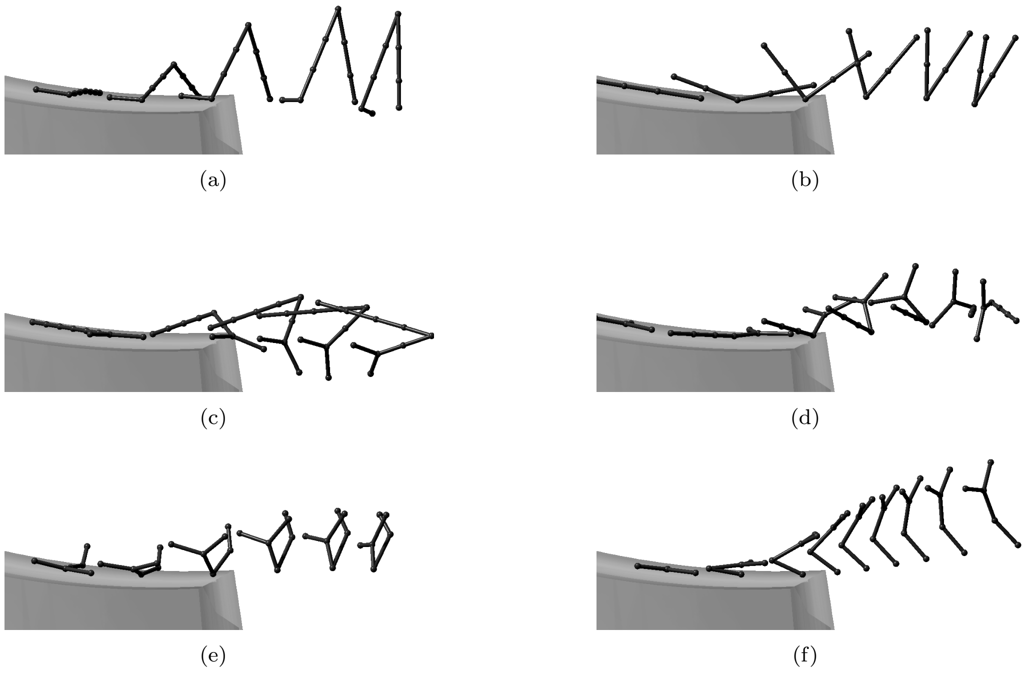

Figure 4. A variety of evolved jumping techniques. (a) An agent leaping in a low-friction environment by bending its parts in a plane parallel to the direction of movement. Additional part at the end of one limb increases stability during the jump. (b) A folding agent that increases jump length by moving up its center of mass. (c) An agent pushing off from the edge of the ramp to increase its velocity. (d) A typical Y-shaped agent evolved in a high-friction environment. High ground friction caused rotation of this agent by 90 degrees during inrun. (e) The Y-shaped agent that jumps by folding two parts together. (f) The agent that performs a spectacular jump by stretching its limb.

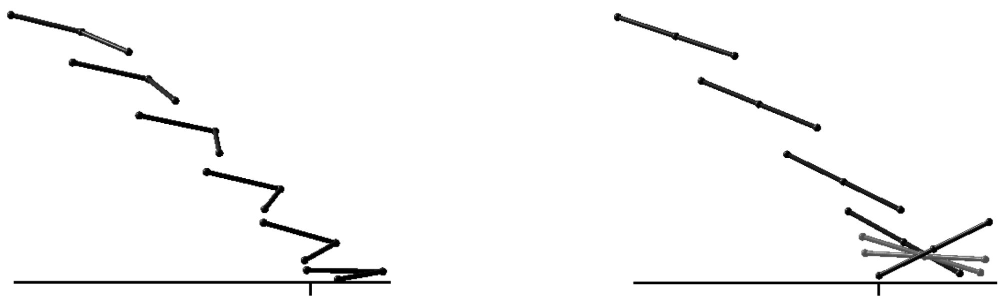

Figure 5. Landing technique: this agent folds just before landing to increase its jump distance. Left: neural network is active. Right: neural network is inactive to show how the passive body would behave without evolved control: it would bounce backward and lose some distance.

a highly realistic simulation engine could yield realistic behaviors known from real-world human competitions, and it might inspire a new view of ski jumping techniques because of the minimal bias towards existing knowledge.

Experiments with manual design of jumping agents failed. It was extremely hard for a human to design an agent (a body and coupled control) that would be competitive to evolved solutions.

While we focused on a single criterion: maximizing jump distance, other factors could also be taken into account. These include, for example, the shape of the trajectory, take-off velocity and vertical velocity, robustness against changing wind, maximizing stability and minimizing axial or angular limb stress during landing. The body structure could be pre-designed and fixed to discover the best jumping strategy (both in terms of jump distance and safety, cf. [8]) for the given design.

Acknowledgement

This work has been supported by the Ministry of Science and Higher Education, grant No. N N519 3505 33.

Bibliography

[1] Andrew Adamatzky and Maciej Komosinski, editors. *Artificial Life Models in Hardware*. Springer, 2009.

[2] Peter Bentley. *Evolutionary Design by Computers*. Morgan Kaufmann, 1999.

[3] A.E. Eiben and J.E. Smith. *Introduction to evolutionary computing*. Springer, 2003.

[4] Pablo Funes and Jordan B. Pollack. Evolutionary body building: adaptive physical designs for robots. *Artificial Life*, 4(4):337–357, Autumn 1998.

[5] Maciej Komosinski and Adam Rotaru-Varga. Comparison of different genotype encodings for simulated 3D agents. *Artificial Life Journal*, 7(4):395–418, Fall 2001.

[6] Maciej Komosinski and Szymon Ulatowski. Framsticks: Creating and understanding complexity of life. In Maciej Komosinski and Andrew Adamatzky, editors, *Artificial Life Models in Software*, chapter 5. Springer, New York, second edition, 2009.

[7] Carl D. Meyer. *Matrix Analysis and Applied Linear Algebra*. Society for Industrial and Applied Mathematics (SIAM), 2000.

[8] W. Müller. Determinants of ski-jump performance and implications for health, safety and fairness. *Sports Med*, 39:85–106, 2009.

[9] B. Schmölzer and W. Müller. The importance of being light: aerodynamic forces and weight in ski jumping. *J Biomech*, 35:1059–1069, Aug 2002.

[10] B. Schmölzer and W. Müller. Individual flight styles in ski jumping: results obtained during Olympic Games competitions. *J Biomech*, 38:1055–1065, May 2005.

[11] H. Schwameder. Biomechanics research in ski jumping, 1991–2006. *Sports Biomech*, 7:114–136, Jan 2008.

[12] Russell Smith. Open Dynamics Engine. http://www.ode.org/, 2007.

[13] Tim Taylor and Colm Massey. Recent developments in the evolution of morphologies and controllers for physically simulated creatures. *Artificial Life*, 7(1):77–88, Winter 2001.