Evolving ice skating behavior with a Nao robot in a simulated environment using HyperNEAT

Bas Bootsma (0719080) Roland Meertens (3009653)
Tom de Ruijter (3016269)

January 22, 2012

1 Introduction

...

Literature background...

In section 2 *Initial Experimental Setup* most of the details of the project will be explained and our .... In section ...

2 Initial Experimental Setup

2.1 NEAT

NeuroEvolution of Augmenting Topologies (NEAT) technique developed Ken O. Stanley in 2002, which makes use of genetic algorithms for evolving artificial neural networks [2]. The technique both alters the weights and the structure (topologies) of the network to find a balance between the fitness and the diversity of the solutions. NEAT has several properties which make it an interesting technique.

- **Complexifying**: Complexifying is the incremental increase of complexity over time. This means that NEAT starts out with simple topologies and gradually increases complexity over generations.
- **Speciation**: With speciation topologies which are similar are grouped together in niches. This protects for example new topologies, which generaly have worse fitness when compared to already existing topologies, thus allowing the new topologies evolve in a protected niche instead of immediatly discarding them.
- **Historical marking**: Each gene is assigned a unique historical marker which allows later generations to track origin of their genes.

2.2 HyperNEAT

HyperNEAT is the successor of NEAT and provides several extra properties which we believe are beneficial for generating ice skating behavior.

- **Geometric relationships**: HyperNEAT is aware of the geometric relationships of the input. In our case this means that it not only knows the different joints, but also the relationships between the joints. For example, it might be able to exploit the relationship between the ankle and knee joint by posing extra restrictions to prevent the Nao robot from falling over.
- **Symmetrical and recurrent patterns**: HyperNEAT is able to find symmetrical and recurrent patterns. In our case this is important, since ice skating behavior contains symmetrical and recurrent elements, which we hope HyperNEAT is able to find.

3 Hypothetical Experimental Setup

In the previous section we have explained the major details of our project and we also have explained how we did not get HyperNEAT to work successfully. In the case that we did got HyperNEAT working successfully we would have liked to perform several experiments. In the following sections the hypothetical setup, the experiments and the related HyperNEAT configurations are described (and the related expected results).

3.1 Setup

As we already explained we would have wanted to use a different approach, namely using a direct feedback loop from the simulator to the HyperNEAT algorithm. This means that for each time step t the current joint values of the robot in the simulator are read and feed as input into the HyperNEAT algorithm which produces the joint values for time step $t + 1$. Trying to evolve a complete behaviour (i.e. a motion file) has deemed to prove too difficult. However the described approach allows for a more direct mapping between the evolving behavior and the HyperNEAT algorithm in which the problem can be solved in smaller steps. A similar approach was used in evolving a walking gait for a quadruped using generative enconding [1].

* Able to use the scaling abilities of HyperNEAT; * Something about how long (settings) each run would take, etc.

3.2 Experiments

In order to investige our research question, whether it is possible to learn ice skating behavior to a Nao robot in a simulated environment, we would have to perform several experiments. As we have already explained ice skating behaviour consists of two phases, namely the startup phase and the recurrent phase. Since we are primarely interested in the recurrent phase all the experiments focus on trying to evolve an behaviour which has the property of a recurrent motion. This means that any evolved behavior not only has to work for the duration in which the algorithm runs, but also for durations larger than what is being evolved for. ...

3.2.1 Condition: With Initial Velocity

For the baseline condition we are interested in the recurrent phase of an ice skating behavior. Therefor we want to investigate whether it is possible to evolve an ice skating behavior from a standing position with an initial velocity. This means that the Nao is in the initial position (i.e. all the joints are initially at 0) and a specific force has been applied to the robot which results in a certain velocity.

@TODO: THis probably means that the robot only has too small movements...

3.2.2 Condition: From Resting Position

If the evolved behaviour from the condition with an initial velocity for the Nao robot, works out out well, we want to investigate whether it is possible to evolve a similar behaviour from a resting position. In this condition the initial position of the Nao robot is the same as the condition with the initial velocity except no velocity is given. This means that the algorithm has to evolve a behaviour for both phases. This condition will probably take much more computational time to

In the case that both previous conditions produced no suitable ice skating motions there was also a different approach we could have taken. That is adding an already evolved behaviour to the popuplation. In this case there were two behaviours possible, namely a standard walking behaviour and a human ice skating behaviour. Even if both previous conditions did produce suitable ice skating motions it would still have been interesting to investigate whether adding such behaviours by default to the population would help to evolve a suitable ice skating behaviour.

3.2.3 Condition: Walking Behaviour

By default a walking motion is already provided by Webots containing the exact joint values at each time step. We could have used the motion file to train the algorithm with a fitness function based on how much the output differs from the already known joint values. After a certain number of generations or until the algorithm has perfectly learning the motion we could then save the CPPN and ... load it in

3.2.4 Condition: Human Ice Skating Behaviour From a Kinect

3.3 Fitness Functions

Predictions about the experiments...

3.4 HyperNEAT Configurations

3.5 Expected Results

4 Proof of Concept

5 Conclusion

References

[1] J. Clune, B.E. Beckmann, C. Ofria, and R.T. Pennock. Evolving coordinated quadruped gaits with the hyperneat generative encoding. In *Evolutionary Computation, 2009. CEC '09. IEEE Congress on*, pages 2764 –2771, may 2009.

[2] Wikipedia. Neuroevolution of augmenting topologies. `http://en.wikipedia.org/wiki/Neuroevolution_of_augmenting_topologies`, 2012. [Online; last accessed 16-October-2012].