CS6271

EVOLUTIONARY HUMANOID ROBOTICS

I. EAs & Control of Systems

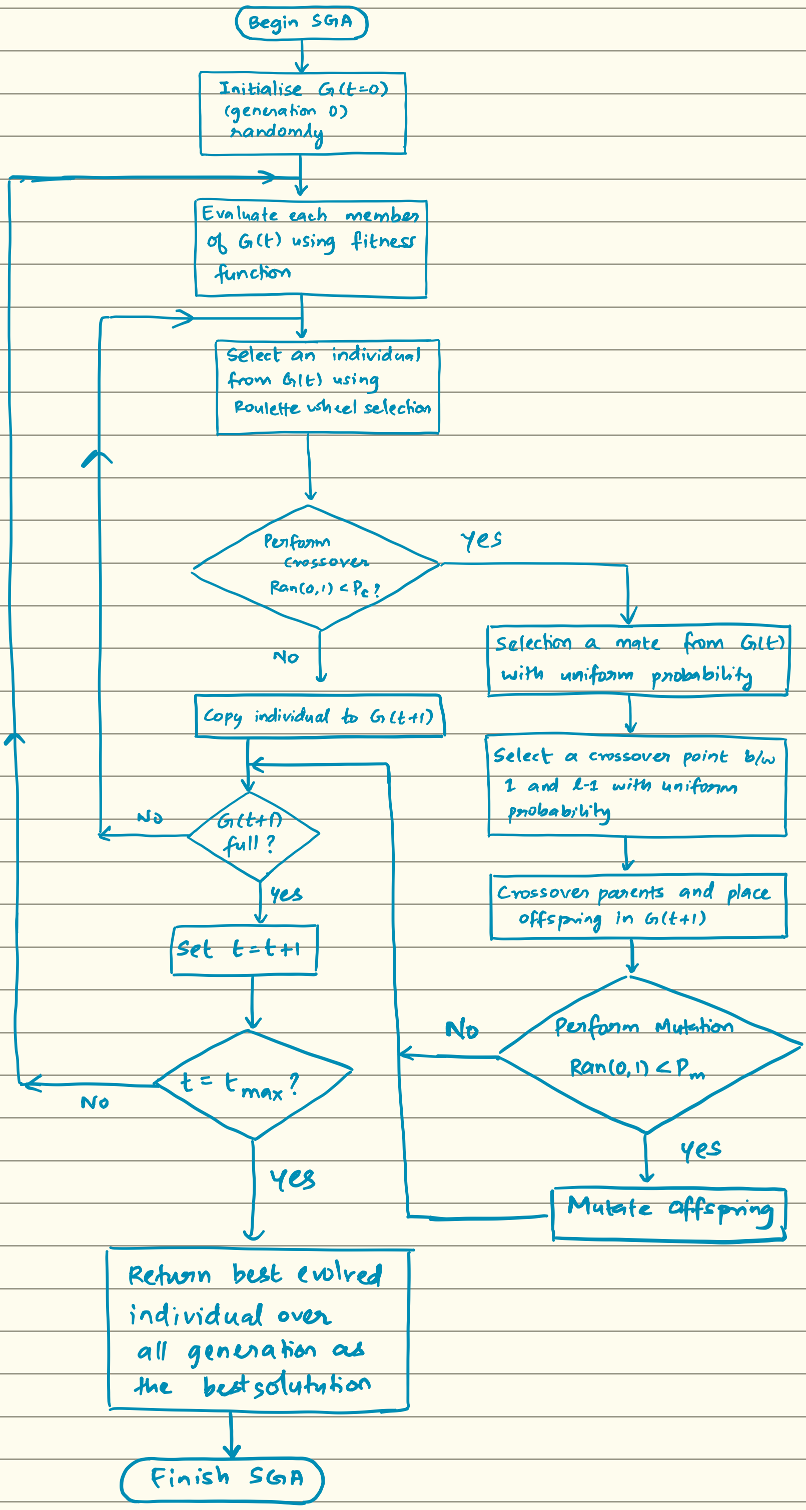
- Motobot

- Festo Emotion Butterflies

- Genetic Algorithms

  - Search algorithms based on mechanics of natural selection & natural genetics.

  - Survival of the fittest
  - structured but randomized exchange of info b/w competing solutions.
  - Exploit historical info to improve performance over time.
  - SGA Flowchart:



II. Evolutionary Robotics

- Evolutionary Robotics involves the application of evolutionary techniques to the generation of either the "brain" (control systems) or the "body" (morphology) of autonomous robots, or perhaps both.

- Early research in ER (early, mid 1990s) → use of wheeled robots.

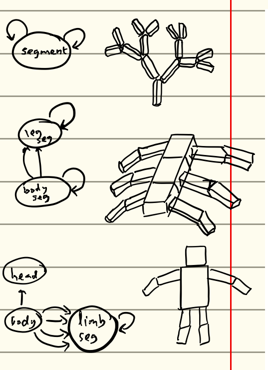
III. Research in ER

1. Level of bio-inspiration
   (drawing inspiration / mimic aspects of natural behaviours)

2. Level of Physical Realisation
   - → simple control algorithms
     creatures with certain defined structures
     do not utilise realistic physics engines to generate motions.
   - → Robots with correct physical model
     utilising simulators with accurate physics
   - → Robots with an actual physical realisation in a real embodied robot.

- Evolving Virtual Creatures
  — Karl Sims, 1994

  - Simulated Evolution of both "body" and "brain"
  - Individual Representation: directed graph of nodes and their connections.
    - composed of articulated 3D rigid components (simple rectangular solids) which are allowed to overlap at the joints.
    - Each node contains info about:
      - dimensions of rigid component
      - Type of joint
      - degrees of freedom
      - Type of joint movements



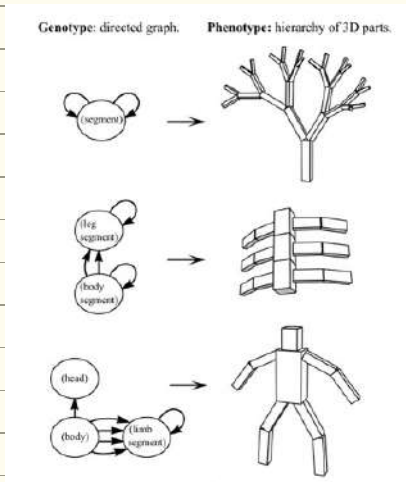


Figure 1: Designed examples of genotype graphs and corresponding creature morphologies.

- Three types of sensors:

    1. Joint Angle Sensors: detected current angle at each joint for each of its degrees of freedom

    2. Contact Sensors: For collision detection

    3. Photo sensors: For light detection in simulated env.

- Robot movement: Force generated on joints by motor effectors

- Evolvind different Behaviours:

    1. Walking

        fitness func $\propto$ distance travelled by centre of mass over a period of time.

    2. Jumping
        2 fitness func.

        $f_1 \propto$ max height above ground plane reached by lowest part of the creature

        $f_2 \propto$ avg height of lowest part of creature above ground over the simulation period.

    3. Swimming

        viscosity effect implemented

- Conclusion

    It might be easier to evolve virtual entities exhibiting intelligent behaviour than it would be for humans to design & build them.

- Acknowledgement:
    As computers become more powerful, the creation of virtual actors, whether animals, human or completely unearthly, may be limited mainly by our ability to design them, rather than our ability to satisfy their computational requirements.

    A control system that someday actually generates "intelligent" behaviour might tend to be a complex mess beyond our understanding.

- Evolving Behaviours on a Real Robot

1994, at EPFL, Switzerland

By Dario & Francesco

- Kephera Robot - small miniature wheeled robot

- Goal: Robot to navigate its way through a looped maze without colliding with the walls.

- SGA used to evolve weights of a neural network
- Neural network - eight sensory neurons.
- Fitness Function:

diff b/w speed values of the two wheels

$$\varphi = V(1-\sqrt{\Delta V})(1-i)$$

avg rotation speed two wheels

normalized activation value of IR sensor with highest activity level

- The fitness function promotes individuals travelling at:

  - high speed
  - in relatively straight trajectories
  - while avoiding collisions with the walls of the maze.

Evolving Virtual Creatures

Karl Sims

Thinking Machines Corporation
245 First Street, Cambridge, MA 02142

Abstract

This paper describes a novel system for creating virtual creatures that move and behave in simulated three-dimensional physical worlds. The morphologies of creatures and the neural systems for controlling their muscle forces are both generated automatically using genetic algorithms. Different fitness evaluation functions are used to direct simulated evolutions towards specific behaviors such as swimming, walking, jumping, and following.

A genetic language is presented that uses nodes and connections as its primitive elements to represent directed graphs, which are used to describe both the morphology and the neural circuitry of these creatures. This genetic language defines a hyperspace containing an indefinite number of possible creatures with behaviors, and when it is searched using optimization techniques, a variety of successful and interesting locomotion strategies emerge, some of which would be difficult to invent or build by design.

1 Introduction

A classic trade-off in the field of computer graphics and animation is that of complexity vs. control. It is often difficult to build interesting or realistic virtual entities and still maintain control over them. Sometimes it is difficult to build a complex virtual world at all, if it is necessary to conceive, design, and assemble each component. An example of this trade-off is that of kinematic control vs. dynamic simulation. If we directly provide the positions and angles for moving objects, we can control each detail of their behavior, but it might be difficult to achieve physically plausible motions. If we instead provide forces and torques and simulate the resulting dynamics, the result will probably look correct, but then it can be very difficult to achieve the desired behavior, especially as the objects we want to control become more complex. Methods have been developed for dynamically controlling specific objects to successfully crawl, walk, or even run [11,12,16], but a new control algorithm must be carefully designed each time a new behavior or morphology is desired.

Optimization techniques offer possibilities for the automatic generation of complexity. The genetic algorithm is a form of artificial evolution, and is a commonly used method for optimization. A Darwinian “survival of the fittest” approach is employed to search for optima in large multidimensional spaces [5,7]. Genetic algorithms permit virtual entities to be created without requiring an understanding of the procedures or parameters used to generate them. The measure of success, or *fitness*, of each individual can be calculated automatically, or it can instead be provided interactively by a user. Interactive evolution allows procedurally generated results to be explored by simply choosing those that are the most aesthetically desirable for each generation [2,18,19,21].

The user sacrifices some control when using these methods, especially when the fitness is procedurally defined. However, the potential gain in automating the creation of complexity can often compensate for this loss of control, and a higher level of user influence is still maintained by the fitness criteria specification.

In several cases, optimization has been used to automatically generate dynamic control systems for given articulated structures: de Garis has evolved weight values for neural networks [4], Ngo and Marks have performed genetic algorithms on stimulus-response pairs [14], and van de Panne and Fiume have optimized sensor-actuator networks [15]. Each of these methods has resulted in successful locomotion of two-dimensional stick figures.

The work presented here is related to these projects, but differs in several respects. In previous work, control systems were generated for fixed structures that were user-designed, but here entire creatures are evolved: the optimization determines the creature morphologies as well as their control systems. Also, here the creatures' bodies are three-dimensional and fully physically based. The three-dimensional physical structure of a creature can adapt to its control system, and vice versa, as they evolve together. The “nervous systems” of creatures are also completely determined by the optimization: the number of internal nodes, the connectivity, and the type of function each neural node performs are included in the genetic description of each creature, and can grow in complexity as an evolution proceeds. Together, these remove the necessity for a user to provide any specific creature information such as shape, size, joint constraints, sensors, actuators, or internal neural parameters. Finally, here a developmental process is used to generate the creatures and their control systems, and allows similar components including their local neural circuitry to be defined once and then replicated, instead of requiring each to be separately specified. This approach is related to L-systems, graftal grammars, and object instancing techniques [6,8,10,13,20].

It is convenient to use the biological terms *genotype* and *phenotype* when discussing artificial evolution. A *genotype* is a coded representation of a possible individual or problem solution. In biological systems, a genotype is usually composed of DNA and contains the instructions for the development of an organism. Genetic algorithms typically use populations of genotypes consisting of strings of binary digits or parameters. These are read to produce *phenotypes* which are then evaluated according to some fitness criteria and selectively reproduced. New genotypes are generated by copying, mutating, and/or combining the genotypes of the most fit individuals, and as the cycle repeats the population should ascend to higher and higher levels of fitness.

Variable length genotypes such as hierarchical Lisp expressions

SIGGRAPH '94, July 24-29, Orlando, Florida
© ACM 1994 ISBN: 0-89791-667-0 ...$5.00

or other computer programs can be useful in expanding the set of possible results beyond a predefined genetic space of fixed dimensions. Genetic languages such as these allow new parameters and new dimensions to be added to the genetic space as an evolution proceeds, and therefore define rather a *hyperspace* of possible results. This approach has been used to genetically program solutions to a variety of problems [1,9], as well as to explore procedurally generated images and dynamical systems [18,19].

In the spirit of unbounded genetic languages, directed graphs are presented here as an appropriate basis for a grammar that can be used to describe both the morphology and nervous systems of virtual creatures. New features and functions can be added to creatures, or existing ones removed, so the levels of complexity can also evolve.

The next two sections explain how virtual creatures can be represented by directed graphs. The system used for physical simulation is summarized in section 4, and section 5 describes how specific behaviors can be selected. Section 6 explains how evolutions are performed with directed graph genotypes, and finally a range of resulting creatures is shown.

2 Creature Morphology

In this work, the phenotype embodiment of a virtual creature is a hierarchy of articulated three-dimensional rigid parts. The genetic representation of this morphology is a directed graph of nodes and connections. Each graph contains the developmental instructions for growing a creature, and provides a way of reusing instructions to make similar or recursive components within the creature. A phenotype hierarchy of parts is made from a graph by starting at a defined *root-node* and synthesizing parts from the node information while tracing through the connections of the graph. The graph can be recurrent. Nodes can connect to themselves or in cycles to form recursive or fractal like structures. They can also connect to the same child multiple times to make duplicate instances of the same appendage.

Each node in the graph contains information describing a rigid part. The *dimensions* determine the physical shape of the part. A *joint-type* determines the constraints on the relative motion between this part and its parent by defining the number of degrees of freedom of the joint and the movement allowed for each degree of freedom. The different joint-types allowed are: *rigid, revolute, twist, universal, bend-twist, twist-bend*, or *spherical*. *Joint-limits* determine the point beyond which restoring spring forces will be exerted for each degree of freedom. A *recursive-limit* parameter determines how many times this node should generate a phenotype part when in a recursive cycle. A set of local *neurons* is also included in each node, and will be explained further in the next section. Finally, a node contains a set of *connections* to other nodes.

Each connection also contains information. The placement of a child part relative to its parent is decomposed into *position, orientation, scale*, and *reflection*, so each can be mutated independently. The position of attachment is constrained to be on the surface of the parent part. Reflections cause negative scaling, and allow similar but symmetrical sub-trees to be described. A *terminal-only* flag can cause a connection to be applied only when the recursive limit is reached, and permits tail or hand-like components to occur at the end of chains or repeating units.

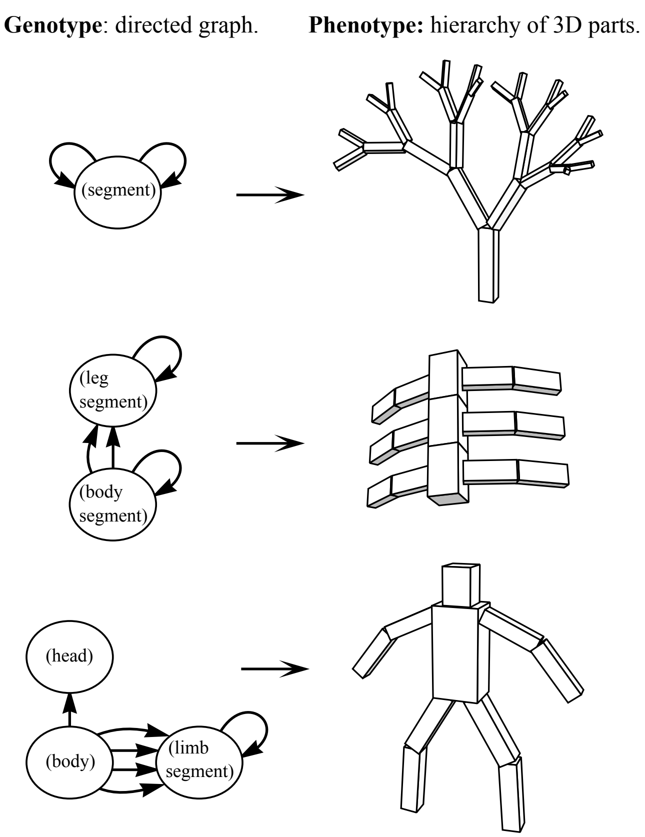


Figure 1: Designed examples of genotype graphs and corresponding creature morphologies.

Figure 1 shows some simple hand-designed graph topologies and resulting phenotype morphologies. Note that the parameters in the nodes and connections such as *recursive-limit* are not shown for the genotype even though they affect the morphology of the phenotype. The nodes are anthropomorphically labeled as "body," "leg," etc. but the genetic descriptions actually have no concept of specific categories of functional components.

3 Creature Control

A virtual "brain" determines the behavior of a creature. The brain is a dynamical system that accepts input sensor values and provides output effector values. The output values are applied as forces or torques at the degrees of freedom of the body's joints. This cycle of effects is shown in Figure 2.

Sensor, effector, and internal neuron signals are represented here by continuously variable scalars that may be positive or negative. Allowing negative values permits the implementation of single effectors that can both push and pull. Although this may not be biologically realistic, it simplifies the more natural development of muscle pairs.

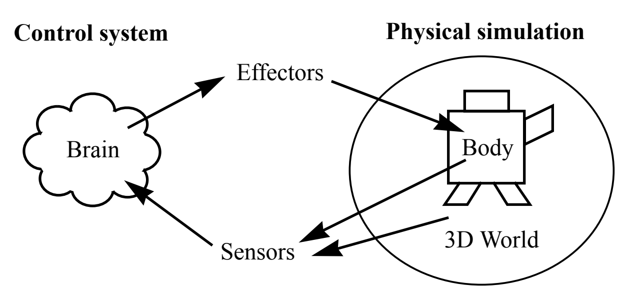


Figure 2: The cycle of effects between brain, body and world.

3.1 Sensors

Each sensor is contained within a specific part of the body, and measures either aspects of that part or aspects of the world relative to that part. Three different types of sensors were used for these experiments:

1. *Joint angle sensors* give the current value for each degree of freedom of each joint.

2. *Contact sensors* activate (1.0) if a contact is made, and negatively activate (-1.0) if not. Each contact sensor has a sensitive region within a part's shape and activates when any contacts occur in that area. In this work, contact sensors are made available for each face of each part. No distinction is made between self-contact and environmental contact.

3. *Photosensors* react to a global light source position. Three photosensor signals provide the coordinates of the normalized light source direction relative to the orientation of the part. This is the same as having pairs of opposing photosensitive surfaces in which the left side negates its response and adds it to the right side for the total response.

Other types of sensors, such as accelerometers, additional proprioceptors, or even sound or smell detectors could also be implemented, but these basic three are enough to allow interesting and adaptive behaviors to occur. The inclusion of the different types of sensors in an evolving virtual brain can be enabled or disabled as appropriate depending on the physical environment and behavior goals. For example, contact sensors are enabled for land environments, and photosensors are enabled for following behaviors.

3.2 Neurons

Internal neural nodes are used to give virtual creatures the possibility of arbitrary behavior. Ideally a creature should be able to have an internal state beyond its sensor values, or be affected by its history.

In this work, different neural nodes can perform diverse functions on their inputs to generate their output signals. Because of this, a creature's brain might resemble a dataflow computer program more than a typical neural network. This approach is probably less biologically realistic than just using sum and threshold functions, but it is hoped that it makes the evolution of interesting behaviors more likely. The set of functions that neural nodes can have is: *sum, product, divide, sum-threshold, greater-than, sign-of, min, max, abs, if, interpolate, sin, cos, atan, log, expt, sigmoid, integrate, differentiate, smooth, memory, oscillate-wave, and oscillate-saw.*

Some functions compute an output directly from their inputs, while others such as the oscillators retain some state and can give time varying outputs even when their inputs are constant. The number of inputs to a neuron depends on its function, and here is at most three. Each input contains a connection to another neuron or a sensor from which to receive a value. Alternatively, an input can simply receive a constant value. The input values are first scaled by weights before being operated on.

For each simulated time interval, every neuron computes its output value from its inputs. In this work, two brain time steps are performed for each dynamic simulation time step so signals can propagate through multiple neurons with less delay.

3.3 Effectors

Each effector simply contains a connection to a neuron or a sensor from which to receive a value. This input value is scaled by a constant weight, and then exerted as a joint force which affects the dynamic simulation and the resulting behavior of the creature. Different types of effectors, such as sound or scent emitters, might also be interesting, but only effectors that exert simulated muscle forces are used here.

Each effector controls a degree of freedom of a joint. The effectors for a given joint connecting two parts, are contained in the part further out in the hierarchy, so that each non-root part operates only a single joint connecting it to its parent. The angle sensors for that joint are also contained in this part.

Each effector is given a *maximum-strength* proportional to the maximum cross sectional area of the two parts it joins. Effector forces are scaled by these strengths and not permitted to exceed them. Since strength scales with area, but mass scales with volume, as in nature, behavior does not always scale uniformly.

3.4 Combining Morphology and Control

The genotype descriptions of virtual brains and the actual phenotype brains are both directed graphs of nodes and connections. The nodes contain the sensors, neurons, and effectors, and the connections define the flow of signals between these nodes. These graphs can also be recurrent, and as a result the final control system can have feedback loops and cycles.

However, most of these neural elements exist within a specific part of the creature. Thus the genotype for the nervous system is a nested graph: the morphological nodes each contain graphs of the neural nodes and connections. Figure 3 shows an example of an evolved nested graph.

When a creature is synthesized from its genetic description, the neural components described within each part are generated along with the morphological structure. This causes blocks of neural control circuitry to be replicated along with each instanced part, so each duplicated segment or appendage of a creature can have a similar but independent local control system.

These local control systems can be connected to enable the possibility of coordinated control. Connections are allowed between adjacent parts in the hierarchy: the neurons and effectors within a part can receive signals from sensors or neurons in their parent part or in their child parts.

Creatures are also given a set of neurons that are not associated

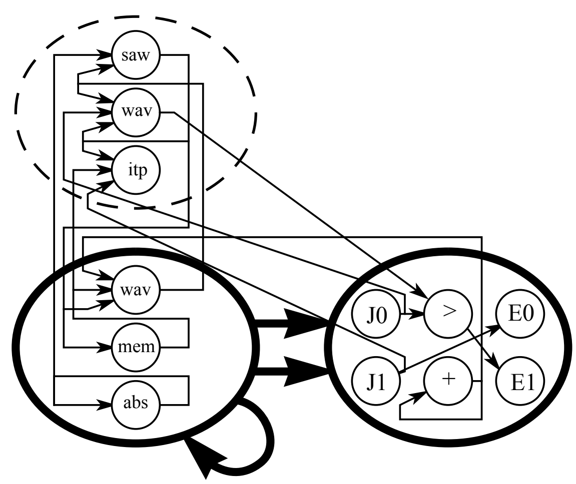


Figure 3: Example evolved nested graph genotype. The outer graph in bold describes a creature's morphology. The inner graph describes its neural circuitry. J0 and J1 are joint angle sensors, and E0 and E1 are effector outputs. The dashed node contains centralized neurons that are not associated with any part.

with a specific part, and are copied only once into the phenotype. This gives the opportunity for the development of global synchronization or centralized control. These neurons can receive signals from each other or from sensors or neurons in specific instances of any of the creature's parts, and the neurons and effectors within the parts can optionally receive signals from these unassociated-neuron outputs.

In this way the genetic language for morphology and control is merged. A local control system is described for each type of part, and these are copied and connected into the hierarchy of the creature's body to make a complete distributed nervous system. Figure 4a shows the creature morphology resulting from the genotype in figure 3. Again, parameters describing shapes, recursive-limits, and weights are not shown for the genotype even though they affect the phenotype. Figure 4b shows the corresponding brain of this creature. The brackets on the left side of figure 4b group the neural components of each part. Some groups have similar neural systems because they are copies from the same genetic description. This creature can swim by making cyclic paddling motions with four similar flippers. Note that it can be difficult to analyze exactly how a control system such as this works, and some components may not actually be used at all. Fortunately, a primary benefit of using artificial evolution is that understanding these representations is not necessary.

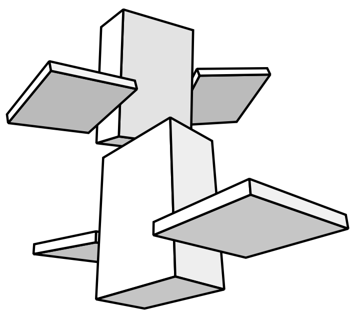


Figure 4a: The phenotype morphology generated from the evolved genotype shown in figure 3.

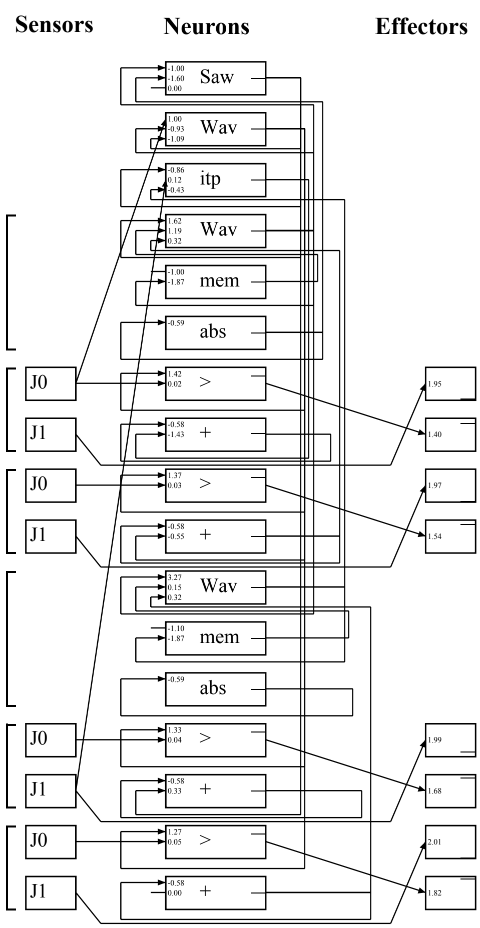


Figure 4b: The phenotype "brain" generated from the evolved genotype shown in figure 3. The effector outputs of this control system cause paddling motions in the four flippers of the morphology above.

4 Physical Simulation

Dynamics simulation is used to calculate the movement of creatures resulting from their interaction with a virtual three-dimensional world. There are several components of the physical simulation used in this work: articulated body dynamics, numerical integration, collision detection, collision response, friction, and an optional viscous fluid effect. These are only briefly summarized here, since physical simulation is not the emphasis of this paper.

Featherstone's recursive O(N) articulated body method is used to calculate the accelerations from the velocities and external forces of each hierarchy of connected rigid parts [3]. Integration determines the resulting motions from these accelerations and is performed by a Runge-Kutta-Fehlberg method which is a fourth order Runge-Kutta with an additional evaluation to estimate the error and adapt the step size. Typically between 1 and 5 integration time steps are performed for each frame of 1/30 second.

The shapes of parts are represented here by simple rectangular solids. Bounding box hierarchies are used to reduce the number of collision tests between parts from $O(N^2)$. Pairs whose world-space bounding boxes intersect are tested for penetrations, and collisions with a ground plane are also tested if one exists. If necessary, the previous time-step is reduced to keep any new penetrations below a certain tolerance. Connected parts are permitted to interpenetrate but not rotate completely through each other. This is achieved by using adjusted shapes when testing for collisions between connected parts. The shape of the smaller part is clipped halfway back from its point of attachment so it can swing freely until its remote end makes contact.

Collision response is accomplished by a hybrid model using both impulses and penalty spring forces. At high velocities, instantaneous impulse forces are used, and at low velocities springs are used, to simulate collisions and contacts with arbitrary elasticity and friction parameters.

A viscosity effect is used for the simulations in underwater environments. For each exposed moving surface, a viscous force resists the normal component of its velocity, proportional to its surface area and normal velocity magnitude. This is a simple approximation that does not include the motion of the fluid itself, but is still sufficient for simulating realistic looking swimming and paddling dynamics.

It is important that the physical simulation be reasonably accurate when optimizing for creatures that can move within it. Any bugs that allow energy leaks from non-conservation, or even round-off errors, will inevitably be discovered and exploited by the evolving creatures. Although this can be a lazy and often amusing approach for debugging a physical modeling system, it is not necessarily the most practical.

5 Behavior Selection

In this work, virtual creatures are evolved by optimizing for a specific task or behavior. A creature is grown from its genetic description as previously explained, and then it is placed in a dynamically simulated virtual world. The brain provides effector forces which move parts of the creature, the sensors report aspects of the world and the creature's body back to the brain, and the resulting physical behavior of the creature is evaluated. After a certain duration of virtual time (perhaps 10 seconds), a *fitness* value is assigned that corresponds to the success level of that behavior. If a creature has a high fitness relative to the rest of the population, it will be selected for survival and reproduction as described in the next section.

Before creatures are simulated for fitness evaluation, some simple viability checks are performed, and inappropriate creatures are removed from the population by giving them zero fitness values. Those that have more than a specified number of parts are removed. A subset of genotypes will generate creatures whose parts initially interpenetrate. A short simulation with collision detection and response attempts to repel any intersecting parts, but those creatures with persistent interpenetrations are also discarded.

Computation can be conserved for most fitness methods by discontinuing the simulations of individuals that are predicted to be unlikely to survive the next generation. The fitness is periodically estimated for each simulation as it proceeds. Those are stopped that are either not moving at all or are doing somewhat worse than the minimum fitness of the previously surviving individuals.

Many different types of fitness measures can be used to perform evolutions of virtual creatures. Four examples of fitness methods are described here.

5.1 Swimming

Physical simulation of a water environment is achieved by turning off gravity and adding the viscous water resistance effect as described. Swimming speed is used as the fitness value and is measured by the distance traveled by the creature's center of mass per unit time. Straight swimming is rewarded over circling by using the maximum distance from the initial center of mass. Continuing movement is rewarded over that from a single initial push, by giving the velocities during the final phase of the test period a stronger relative weight in the total fitness value.

5.2 Walking

The term *walking* is used loosely here to indicate any form of land locomotion. A land environment is simulated by including gravity, turning off the viscous water effect, and adding a static ground plane with friction. Additional inanimate objects can be placed in the world for more complex environments. Again, speed is used as the selection criteria, but the vertical component of velocity is ignored.

For land environments, it can be necessary to prevent creatures from generating high velocities by simply falling over. This is accomplished by first running the simulation with no friction and no effector forces until the height of the center of mass reaches a stable minimum.

5.3 Jumping

Jumping behavior can be selected for by measuring the maximum height above the ground of the lowest part of the creature. An alternative method is to use the average height of the lowest part of the creature during the duration of simulation.

5.4 Following

Another evaluation method is used to select for creatures that can adaptively follow a light source. Photosensors are enabled, so the effector output forces and resulting behavior can depend on the relative direction of a light source to the creature. Several trials are run with the light source in different locations, and the speeds at which a creature moves toward it are averaged for the fitness value. Following behaviors can be evolved for both water and land environments.

Fleeing creatures can also be generated in a similar manner, or following behavior can be transformed into fleeing behavior by simply negating a creature's photo sensor signals.

Once creatures are found that exhibit successful following behaviors, they can be led around in arbitrary paths by movement of the light sources.

6 Creature Evolution

An evolution of virtual creatures is begun by first creating an initial population of genotypes. These initial genotypes can come from several possible sources: new genotypes can be synthesized "from scratch" by random generation of sets of nodes and connections, an existing genotype from a previous evolution can be used to seed the initial population of a new evolution, or a seed genotype can be designed by hand. However, no hand-designed seed genotypes were used in the examples shown here.

A *survival-ratio* determines the percentage of the population that will survive each generation. In this work, population sizes were typically 300, and the survival ratio was 1/5. If the initially generated population has fewer individuals with positive fitness than the number that should survive, another round of seed genotypes is generated to replace those with zero fitness.

For each generation, creatures are grown from their genetic descriptions, and their fitness values are measured by a method such as those described in the previous section. The individuals whose fitnesses fall within the survival percentile are then reproduced, and their offspring fill the slots of those individuals that did not survive. The survivors are kept in the population for the next generation, and the total size of the population is maintained. The number of offspring that each surviving individual generates is proportional to its fitness – the most successful creatures make the most children.

Offspring are generated from the surviving creatures by copying and combining their directed graph genotypes. When these graphs are reproduced they are subjected to probabilistic variation or mutation, so the corresponding phenotypes are similar to their parents but have been altered or adjusted in random ways.

6.1 Mutating Directed Graphs

A directed graph is mutated by the following sequence of steps:

1. The internal parameters of each node are subjected to possible alterations. A mutation frequency for each parameter type determines the probability that a mutation will be applied to it at all. Boolean values are mutated by simply flipping their state. Scalar values are mutated by adding several random numbers to them for a Gaussian-like distribution so small adjustments are more likely than drastic ones. The scale of an adjustment is relative to the original value, so large quantities can be varied more easily and small ones can be carefully tuned. A scalar can also be negated. After a mutation occurs, values are clamped to their legal bounds. Some parameters that only have a limited number of legal values are mutated by simply picking a new value at random from the set

of possibilities.

2. A new random node is added to the graph. A new node normally has no effect on the phenotype unless a connection also mutates a pointer to it. Therefore a new node is always initially added, but then garbage collected later (in step 5) if it does not become connected. This type of mutation allows the complexity of the graph to grow as an evolution proceeds.

3. The parameters of each connection are subjected to possible mutations, in the same way the node parameters were in step 1. With some frequency the connection pointer is moved to point to a different node which is chosen at random.

4. New random connections are added and existing ones are removed. In the case of the neural graphs these operations are not performed because the number of inputs for each element is fixed, but the morphological graphs can have a variable number of connections per node. Each existing node is subject to having a new random connection added to it, and each existing connection is subject to possible removal.

5. Unconnected elements are garbage collected. Connectedness is propagated outwards through the connections of the graph, starting from the root node of the morphology, or from the effector nodes of neural graphs. Although leaving the disconnected nodes for possible reconnection might be advantageous, and is probably biologically analogous, at least the unconnected newly added ones are removed to prevent unnecessary growth in graph size.

Since mutations are performed on a per element basis, genotypes with only a few elements might not receive any mutations, where genotypes with many elements would receive enough mutations that they rarely resemble their parents. This is compensated for by temporarily scaling the mutation frequencies by an amount inversely proportional to the size of the current graph being mutated, such that on the average, at least one mutation occurs in the entire graph.

Mutation of nested directed graphs, as are used here to represent creatures, is performed by first mutating the outer graph and then mutating the inner layer of graphs. The inner graphs are mutated last because legal values for some of their parameters (inter-node neural input sources) can depend on the topology of the outer graph.

6.2 Mating Directed Graphs

Sexual reproduction allows components from more than one parent to be combined into new offspring. This permits features to evolve independently and later be merged into a single individual. Two different methods for mating directed graphs are presented.

The first is a *crossover* operation (see figure 5a). The nodes of two parents are each aligned in a row as they are stored, and the nodes of the first parent are copied to make the child, but one or more crossover points determine when the copying source should switch to the other parent. The connections of a node are copied with it and simply point to the same relative node locations as before. If the copied connections now point out of bounds because of varying node numbers they are randomly reassigned.

A second mating method *grafts* two genotypes together by connecting a node of one parent to a node of another (see figure 5b). The first parent is copied, and one of its connections is chosen at random and adjusted to point to a random node in the second parent. Newly unconnected nodes of the first parent are removed and the newly connected node of the second parent and any of its descendants are appended to the new graph.

A new directed graph can be produced by either of these two mating methods, or asexually by using only mutations. Offspring

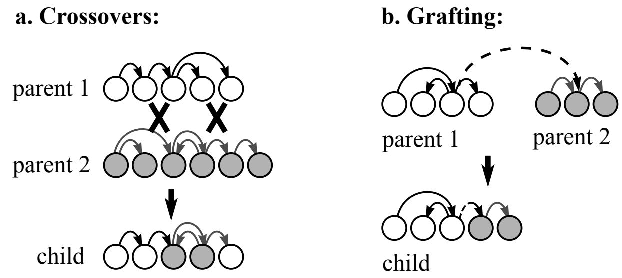


Figure 5: Two methods for mating directed graphs.

from matings are sometimes subjected to mutations afterwards, but with reduced mutation frequencies. In this work a reproduction method is chosen at random for each child to be produced by the surviving individuals using the ratios: 40% asexual, 30% crossovers, and 30% grafting. A second parent is chosen from the survivors if necessary, and a new genotype is produced from the parent or parents.

After a new generation of genotypes is created, a phenotype creature is generated from each, and again their fitness levels are evaluated. As this cycle of variation and selection continues, the population is directed towards creatures with higher and higher fitness.

6.3 Parallel Implementation

This genetic algorithm has been implemented to run in parallel on a Connection Machine® CM-5 in a master/slave message passing model. A single processing node performs the genetic algorithm. It farms out genotypes to the other nodes to be fitness tested, and gathers back the fitness values after they have been determined. The fitness tests each include a dynamics simulation and although most can execute in nearly real-time, they are still the dominant computational requirement of the system. Performing a fitness test per processor is a simple but effective way to parallelize this genetic algorithm, and the overall performance scales quite linearly with the number of processors, as long as the population size is somewhat larger than the number of processors.

Each fitness test takes a different amount of time to compute depending on the complexity of the creature and how it attempts to move. To prevent idle processors from just waiting for others to finish, new generations are started before the fitness tests have been completed for all individuals. Those slower simulations are simply skipped during that reproductive cycle, so all processors remain active. With this approach, an evolution with population size 300, run for 100 generations, might take around three hours to complete on a 32 processor CM-5.

7 Results

Evolutions were performed for each of the behavior selection methods described in section 5. A population of interbreeding creatures often converges toward homogeneity, but each separately run evolution can produce completely different locomotion strategies that satisfy the requested behavior. For this reason, many separate evolutions were performed, each for 50 to 100 generations, and the most successful creatures of each evolution were inspected. A selection of these is shown in figures 6-9. In a few cases, genotypes resulting from one evolution were used as seed genotypes for a second evolution.

The swimming fitness measure produced a large number of simple paddling and tail wagging creatures. A variety of more complex strategies also emerged from some evolutions. A few creatures pulled themselves through the water with specialized sculling appendages. Some used two symmetrical flippers or even large numbers of similar flippers to propel themselves, and several multi-segmented watersnake creatures evolved that wind through the water with sinusoidal motions.

The walking fitness measure also produced a surprising number of simple creatures that could shuffle or hobble along at fairly high speeds. Some walk with lizard-like gaits using the corners of their parts. Some simply wag an appendage in the air to rock back and forth in just the right manner to move forward. A number of more complex creatures emerged that push or pull themselves along, inchworm style. Others use one or more leg-like appendages to successfully crawl or walk. Some hopping creatures even emerged that raise and lower arm-like structures to bound along at fairly high speeds.

The jumping fitness measure did not seem to produce as many different strategies as the swimming and walking optimizations, but a number of simple jumping creatures did emerge.

The light-following fitness measure was used in both water and land environments, and produced a wide variety of creatures that can swim or walk towards a light source. Some consistently and successfully follow the light source at different locations. Others can follow it some of the time, but then at certain relative locations fail to turn towards it. In the water environment, some developed steering fins that turn them towards the light using photosensor inputs. Others adjust the angle of their paddles appropriately as they oscillate along.

Sometimes a user may want to exert more control on the results of this process instead of simply letting creatures evolve entirely automatically. Aesthetic selection is a possible way to achieve this, but observation of the trial simulations of every creature and providing every fitness value interactively would require too much patience on the part of the user. A convenient way of mixing automatic selection with aesthetic selection, is to observe the final successful results of a number of evolutions, and then start new evolutions with those that are aesthetically preferred. Although the control may be limited, this gives the user some influence on the creatures that are developed.

Another method of evolving creatures is to interactively evolve a morphology based on looks only, or alternatively hand design the morphology, and then automatically evolve a brain for that morphology that results in a desirable behavior.

Creatures that evolved in one physical world can be placed in another and evolved further. An evolved watersnake, for example, was placed on land and then evolved to crawl instead of swim.

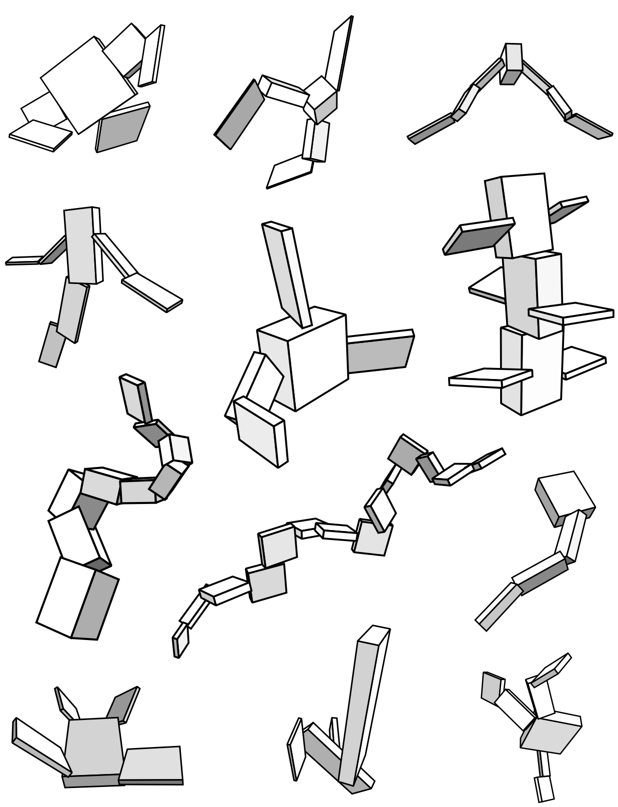


Figure 6: Creatures evolved for swimming.

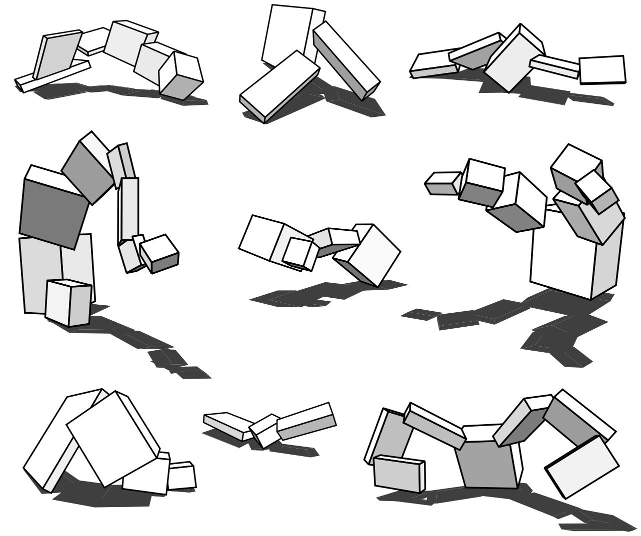


Figure 7: Creatures evolved for walking.

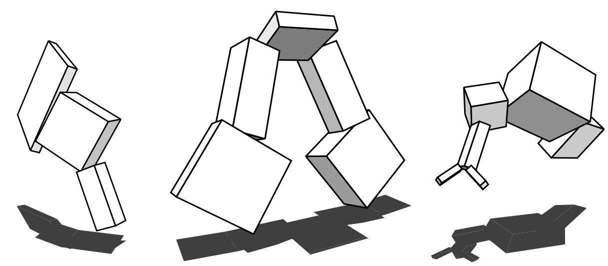


Figure 8: Creatures evolved for jumping.

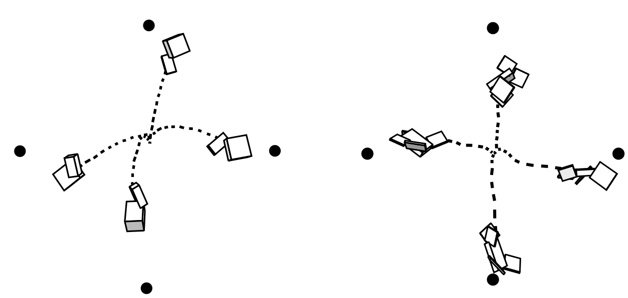


Figure 9: Following behavior. For each creature, four separate trials are shown from the same starting point toward different light source goal locations.

8 Future Work

One direction of future work would be to experiment with additional types of fitness evaluation methods. More complex behaviors might be evolved by defining fitness functions that could measure the level of success at performing more difficult tasks, or even multiple tasks. Fitness could also include the efficiency at which a behavior was achieved. For example, a fitness measure might be the distance traveled divided by the amount of energy consumed to move that distance.

Alternatively, fitness could be defined in a more biologically realistic way by allowing populations of virtual creatures to compete against each other within a physically simulated changing world. Competition has been shown to facilitate complexity, specialization, or even social interactions [17,22]. It becomes difficult to define explicit evaluations that can select for "interesting" behavior, but perhaps systems like these could help produce such results.

Another direction of future work might be to adjust the genetic language of possible creatures to describe only those that could actually be built as real robots. The virtual robots that can best perform a given task in simulation would then be assembled, and would hopefully also perform well in reality.

Much work could be done to dress up these virtual creatures to give them different shapes and improved rendered looks. Flexible skin could surround or be controlled by the rigid components. Various materials could be added such as scales, hair, fur, eyes, or tentacles, and they might flow or bounce using simple local dynamic simulations, even if they did not influence the overall dynamics. The shape details and external materials could also be included in the creatures' genetic descriptions and be determined by artificial evolution.

9 Conclusion

In summary, a system has been described that can generate autonomous three-dimensional virtual creatures without requiring cumbersome user specifications, design efforts, or knowledge of algorithmic details. A genetic language for representing virtual creatures with directed graphs of nodes and connections allows an unlimited hyperspace of possible creatures to be explored. It is believed that these methods have potential as a powerful tool for the creation of desirable complexity for use in virtual worlds and computer animation.

As computers become more powerful, the creation of virtual actors, whether animal, human, or completely unearthly, may be limited mainly by our ability to design them, rather than our ability to satisfy their computational requirements. A control system that someday actually generates "intelligent" behavior might tend to be a complex mess beyond our understanding. Artificial evolution permits the generation of complicated virtual systems without requiring design, and the use of unbounded genetic languages allows evolving systems to increase in complexity beyond our understanding. Perhaps methods such as those presented here will provide a practical pathway toward the creation of intelligent behavior.

Acknowledgments

Thanks to Gary Oberbrunner and Matt Fitzgibbon for Connection Machine and software help. Thanks to Lew Tucker and Thinking Machines Corporation for supporting this research. Thanks to Bruce Blumberg and Peter Schröder for dynamic simulation help and suggestions. And special thanks to Pattie Maes.

References

1. Cramer, N.L., "A Representation for the Adaptive Generation of Simple Sequential Programs," *Proceedings of the First International Conference on Genetic Algorithms*, ed. by J. Grefenstette, 1985, pp.183-187.
2. Dawkins, R., *The Blind Watchmaker*, Harlow Longman, 1986.
3. Featherstone, R., *Robot Dynamics Algorithms*, Kluwer Academic Publishers, Norwell, MA, 1987.
4. de Garis, H., "Genetic Programming: Building Artificial Nervous Systems Using Genetically Programmed Neural Network Modules," *Proceedings of the 7th International Conference on Machine Learning*, 1990, pp.132-139.
5. Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.
6. Hart, J., "The Object Instancing Paradigm for Linear Fractal Modeling," *Graphics Interface*, 1992, pp.224-231.
7. Holland, J.H., *Adaptation in Natural and Artificial Systems*, Ann Arbor, University of Michigan Press, 1975.
8. Kitano, H., "Designing neural networks using genetic algorithms with graph generation system," *Complex Systems*, Vol.4, pp.461-476, 1990.
9. Koza, J., *Genetic Programming: on the Programming of Computers by Means of Natural Selection*, MIT Press, 1992.
10. Lindenmayer, A., "Mathematical Models for Cellular Interactions in Development, Parts I and II," *Journal of Theoretical Biology*, Vol.18, 1968, pp.280-315.
11. McKenna, M., and Zeltzer, D., "Dynamic Simulation of Autonomous Legged Locomotion," *Computer Graphics*, Vol.24, No.4, July 1990, pp.29-38.
12. Miller, G., "The Motion Dynamics of Snakes and Worms," *Computer Graphics*, Vol.22, No.4, July 1988, pp.169-178.
13. Mjolsness, E., Sharp, D., and Alpert, B., "Scaling, Machine Learning, and Genetic Neural Nets," *Advances in Applied Mathematics*, Vol.10, pp.137-163, 1989.
14. Ngo, J.T., and Marks, J., "Spacetime Constraints Revisited," *Computer Graphics*, Annual Conference Series, 1993, pp.343-350.
15. van de Panne, M., and Fiume, E., "Sensor-Actuator Networks," *Computer Graphics*, Annual Conference Series, 1993, pp.335-342.
16. Raibert, M., and Hodgins, J.K., "Animation of Dynamic Legged Locomotion," *Computer Graphics*, Vol.25, No.4, July 1991, pp.349-358.
17. Ray, T., "An Approach to the Synthesis of Life," *Artificial Life II*, ed. by Langton, Taylor, Farmer, & Rasmussen, Addison-Wesley, 1991, pp.371-408.
18. Sims, K., "Artificial Evolution for Computer Graphics," *Computer Graphics*, Vol.25, No.4, July 1991, pp.319-328.
19. Sims, K., "Interactive Evolution of Dynamical Systems," *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, ed. by Varela, Francisco, & Bourgine, MIT Press, 1992, pp.171-178.
20. Smith, A.R., "Plants, Fractals, and Formal Languages," *Computer Graphics*, Vol.18, No.3, July 1984, pp.1-10.
21. Todd, S., and Latham, W., *Evolutionary Art and Computers*, London, Academic Press, 1992.
22. Yaeger, L., "Computational Genetics, Physiology, Metabolism, Neural Systems, Learning, Vision, and Behavior or PolyWorld: Life in a New Context," *Artificial Life III*, ed. by C. Langton, Santa Fe Institute Studies in the Sciences of Complexity, Proceedings Vol. XVII, Addison-Wesley, 1994, pp.263-298.

Evolving 3D Morphology and Behavior by Competition

Karl Sims

Thinking Machines Corporation
(No longer there)

Abstract

This paper describes a system for the evolution and co-evolution of virtual creatures that compete in physically simulated three-dimensional worlds. Pairs of individuals enter one-on-one contests in which they contend to gain control of a common resource. The winners receive higher relative fitness scores allowing them to survive and reproduce. Realistic dynamics simulation including gravity, collisions, and friction, restricts the actions to physically plausible behaviors.

The morphology of these creatures and the neural systems for controlling their muscle forces are both genetically determined, and the morphology and behavior can adapt to each other as they evolve simultaneously. The genotypes are structured as directed graphs of nodes and connections, and they can efficiently but flexibly describe instructions for the development of creatures' bodies and control systems with repeating or recursive components. When simulated evolutions are performed with populations of competing creatures, interesting and diverse strategies and counter-strategies emerge.

1 Introduction

Interactions between evolving organisms are generally believed to have a strong influence on their resulting complexity and diversity. In natural evolutionary systems the measure of fitness is not constant: the reproducibility of an organism depends on many environmental factors including other evolving organisms, and is continuously in flux. Competition between organisms is thought to play a significant role in preventing static fitness landscapes and sustaining evolutionary change.

These effects are a distinguishing difference between natural evolution and optimization. Evolution proceeds with no explicit goal, but optimization, including the genetic algorithm, usually aims to search for individuals with the highest possible fitness values where the fitness measure has been predefined, remains constant, and depends only on the individual being tested.

The work presented here takes the former approach. The fitness of an individual is highly dependent on the specific behaviors of other individuals currently in the population. The hope is that virtual creatures with higher complexity and more interesting behavior will evolve than when applying the selection pressures of optimization alone.

Many simulations of co-evolving populations have been performed which involve competing individuals [1,2]. As examples, Lindgren has studied the evolutionary dynamics of competing game strategy rules [14], Hillis has demonstrated that co-evolving parasites can enhance evolutionary optimization [9], and Reynolds evolves vehicles for competition in the game of tag [19]. The work presented here involves similar evolutionary dynamics to help achieve interesting results when phenotypes have three-dimensional bodies and compete in physically simulated worlds.

In several cases, optimization has been used to automatically generate dynamic control systems for given two-dimensional articulated structures: de Garis has evolved weight values for neural networks [6], Ngo and Marks have applied genetic algorithms to generate stimulus-response pairs [16], and van de Panne and Fiume have optimized sensor-actuator networks [17]. Each of these methods has resulted in successful locomotion of two-dimensional stick figures.

The work presented here is related to these projects, but differs in several respects. Previously, control systems were generated for fixed structures that were user-designed, but here entire creatures are evolved: the evolution determines the creature morphologies as well as their control systems. The physical structure of a creature can adapt to its control system, and vice versa, as they evolve together. Also, here the creatures' bodies are three-dimensional and fully physically based. In addition, a developmental process is used to generate the creatures and their control systems, and allows similar components including their local neural circuitry to be defined once and then replicated, instead of requiring each to be separately specified. This approach is related to L-systems, graftal grammars, and object instancing techniques [8,11,13,15,23]. Finally, the previous work on articulated structures relies only on optimization, and competitions between individuals were not considered.

Published in: *Artificial Life IV Proceedings*, ed. by R. Brooks & P. Maes, MIT Press, 1994, pp28-39.

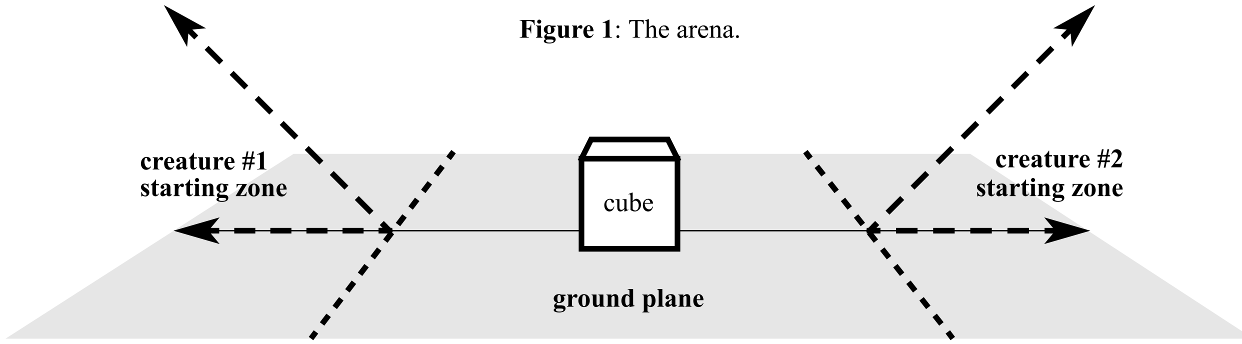


Figure 1: The arena.

A different version of the system described here has also been used to generate virtual creatures by optimizing for specific defined behaviors such as swimming, walking, and following [22].

Genotypes used in simulated evolutions and genetic algorithms have traditionally consisted of strings of binary digits [7,10]. Variable length genotypes such as hierarchical Lisp expressions or other computer programs can be useful in expanding the set of possible results beyond a predefined genetic space of fixed dimensions. Genetic languages such as these allow new parameters and new dimensions to be added to the genetic space as an evolution proceeds, and therefore define rather a *hyperspace* of possible results. This approach has been used to genetically program solutions to a variety of problems [3,12], as well as to explore procedurally generated images and dynamical systems [20,21].

In the spirit of unbounded genetic languages, *directed graphs* are presented here as an appropriate basis for a grammar that can be used to describe both the morphology and neural systems of virtual creatures. The level of complexity is variable for both genotype and phenotype. New features and functions can be added to creatures or existing ones removed, as they evolve.

The next section of this paper describes the environment of the simulated contest and how the competitors are scored. Section 3 discusses different simplified competition patterns for approximating competitive environments. Sections 4 and 5 present the genetic language that is used to represent creatures with arbitrary structure and behavior, and section 6 summarizes the physical simulation techniques used. Section 7 discusses the evolutionary simulations including the methods used for mutating and mating directed graph genotypes, and finally sections 8 and 9 provide results, discussion, and suggestions for future work.

2 The Contest

Figure 1 shows the arena in which two virtual creatures will compete to gain control of a single cube. The cube is placed in the center of the world, and the creatures start on opposite sides of the cube. The second contestant is initially turned by 180 degrees so the relative position of the cube to the creature is consistent from contest to contest no matter which starting side it is assigned. Each creature starts on the ground and behind a diagonal plane slanting up and away from the cube. Creatures are wedged into these "starting zones" until they contact both the ground plane and the diagonal plane, so taller creatures must start further back. This helps prevent the inelegant strategy of simply falling over onto the cube. Strategies like this that utilize only potential energy are further discouraged by relaxing a creature's body before it is placed in the starting zone. The effect of gravity is simulated until the creature reaches a stable minimum state.

At the start of the contest the creatures' nervous systems are activated, and a physical simulation of the creatures' bodies, the cube, and the ground plane begins. The winner is the creature that has the most control over the cube after a certain duration of simulated time (8 seconds were given). Instead of just defining a winner and loser, the margin of victory is determined in the form of a relative fitness value, so there is selection pressure not just to win, but to win by the largest possible margin.

The creatures' final distances to the cube are used to calculate their fitness scores. The shortest distance from any point on the surface of a creatures's parts to the center of the cube is used as its distance value. A creature gets a higher score by being closer to the cube, but also gets a higher score when its opponent is further away. This encourages creatures to reach the cube, but also gives points for keeping the opponent away from it. If d_1 and d_2 are the final shortest distances of each creature to the cube, then the fitnesses for each creature, f_1 and f_2, are given by:

$$f_1 = 1.0 + \frac{d_2 - d_1}{d_1 + d_2}$$

$$f_2 = 1.0 + \frac{d_1 - d_2}{d_1 + d_2}$$

This formulation puts all fitness values in the limited range of 0.0 to 2.0. If the two distances are equal the contestants receive tie scores of 1.0 each, and in all cases the scores will average 1.0.

Credit is also given for having “control” over the cube, beyond just as measured by the minimum distance to it. If both creatures end up contacting the cube, the winner is the one that surrounds it the most. This is approximated by further decreasing the distance value, as used above, when a creature is touching the cube on the side that opposes its center of mass. Since the initial distances are measured from the center of the cube they can be adjusted in this way and still remain positive.

During the simulated contest, if neither creature shows any movement for a full second, the simulation is stopped and the scores are evaluated early to save unnecessary computation.

3 Approximating Competitive Environments

There are many trade-offs to consider when simulating an evolution in which fitness is determined by discrete competitions between individuals. In this work, pairs of individuals compete one-on-one. At every generation of a simulated evolution the individuals in the population are paired up by some pattern and a number of competitions are performed to eventually determine a fitness value for every individual. The simulations of the competitions are by far the dominant computational requirement of the process, so the total number of competitions performed for each generation and the effectiveness of the pattern of competitions are important considerations.

In one extreme, each individual competes with all the others in the population and the average score determines the fitness (figure 2a). However, this requires $(N^2 - N)/2$ total competitions for a single-species population of N individuals. For large populations this is often unacceptable, especially if the competition time is significant, as it is in this work.

In the other extreme, each individual competes with just a single opponent (figure 2b). This requires only N/2 total competitions, but can cause inconsistency in the fitness values since each fitness is often highly dependent on the specific individual that happens to be assigned as the opponent. If the pairing is done at random, and especially if the mutation rate is high, fitness can be more dependent on the luck of receiving a poor opponent than on an individual’s actual ability.

One compromise between these extremes is for each individual to compete against several opponents chosen at random for each generation. This can somewhat dilute the fitness inconsistency problem, but at the expense of more competition simulations.

A second compromise is a tournament pattern (figure 2c) which can efficiently determine a single overall winner with N - 1 competitions. But this also does not necessarily give all individuals fair scores because of the random initial opponent assignments. Also, this pattern does not easily apply to multi-species evolutions where competitions are not performed between individuals within the same species.

A third compromise is for each individual to compete once per generation, but all against the same opponent. The individual with the highest fitness from the previous generation is chosen as this one-to-beat (figure 2d). This also requires N - 1 competitions per generation, but effectively gives fair relative fitness values since all are playing against the same opponent which has proven to be competent. Various interesting instabilities can still occur over generations

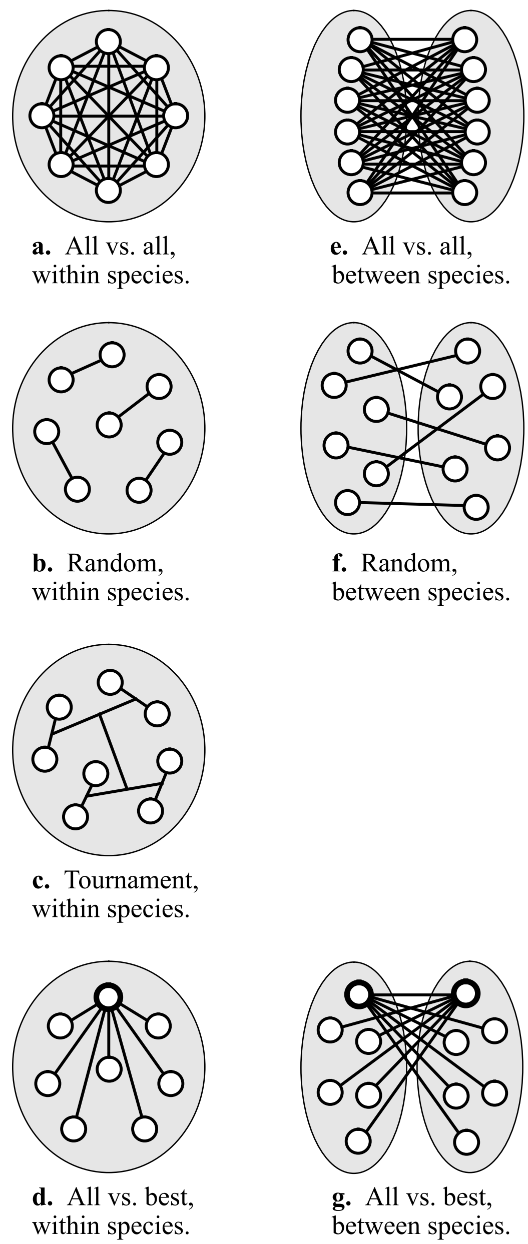


Figure 2: Different pair-wise competition patterns for one and two species. The gray areas represent species of interbreeding individuals, and lines indicate competitions performed between individuals.

however, since the strategy of the “best” individual can change suddenly between generations.

The number of species in the population is another element to consider when simulating evolutions involving competition. A species may be described as an interbreeding subset of individuals in the population. In single-species evolutions individuals will compete against their relatives, but in multi-species evolutions individuals can optionally compete only against individuals from other species. Figure 2 shows graphical representations of some of the different competition patterns described above for both one and two species.

The resulting effects of using these different competition patterns is unfortunately difficult to quantify in this work, since by its nature a simple overall measure of success is absent. Evolutions were performed using several of the methods described above with both one and two species, and the results were subjectively judged. The most “interesting” results occurred when the all vs. best competition pattern was used. Both one and two species evolutions produced some intriguing strategies, but the multi-species simulations tended to produce more interesting interactions between the evolving creatures.

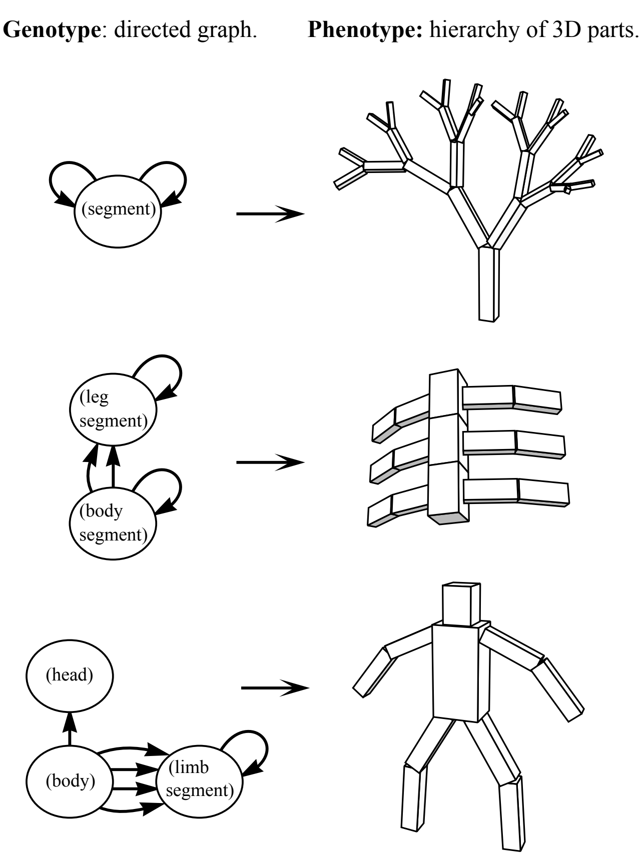


Figure 3: Designed examples of genotype graphs and corresponding creature morphologies.

4 Creature Morphology

In this work, the phenotype embodiment of a virtual creature is a hierarchy of articulated three-dimensional rigid parts. The genetic representation of this morphology is a directed graph of nodes and connections. Each graph contains the developmental instructions for growing a creature, and provides a way of reusing instructions to make similar or recursive components within the creature. A phenotype hierarchy of parts is made from a graph by starting at a defined *root-node* and synthesizing parts from the node information while tracing through the connections of the graph. The graph can be recurrent. Nodes can connect to themselves or in cycles to form recursive or fractal like structures. They can also connect to the same child multiple times to make duplicate instances of the same appendage.

Each node in the graph contains information describing a rigid part. The *dimensions* determine the physical shape of the part. A *joint-type* determines the constraints on the relative motion between this part and its parent by defining the number of degrees of freedom of the joint and the movement allowed for each degree of freedom. The different joint-types allowed are: *rigid, revolute, twist, universal, bend-twist, twist-bend*, or *spherical*. *Joint-limits* determine the point beyond which restoring spring forces will be exerted for each degree of freedom. A *recursive-limit* parameter determines how many times this node should generate a phenotype part when in a recursive cycle. A set of local *neurons* is also included in each node, and will be explained further in the next section. Finally, a node contains a set of *connections* to other nodes.

Each connection also contains information. The placement of a child part relative to its parent is decomposed into *position, orientation, scale*, and *reflection*, so each can be mutated independently. The position of attachment is constrained to be on the surface of the parent part. Reflections cause negative scaling, and allow similar but symmetrical sub-trees to be described. A *terminal-only* flag can cause a connection to be applied only when the recursive limit is reached, and permits tail or hand-like components to occur at the end of chains or repeating units.

Figure 3 shows some simple hand-designed graph topologies and resulting phenotype morphologies. Note that the parameters in the nodes and connections such as *recursive-limit* are not shown for the genotype even though they affect the morphology of the phenotype. The nodes are anthropomorphically labeled as “body,” “leg segment,” etc. but the genetic descriptions actually have no concept of specific categories of functional components.

5 Creature Behavior

A virtual “brain” determines the behavior of a creature. The brain is a dynamical system that accepts input sensor values and provides output effector values. The output values are applied as forces or torques at the degrees of freedom of the body's joints. This cycle of effects is shown in Figure 4.

Sensor, effector, and internal neuron signals are represented here by continuously variable scalars that may be positive or negative. Allowing negative values permits the implementation of single effectors that can both push and pull. Although this may not be biologically realistic, it simplifies the more natural development of muscle pairs.

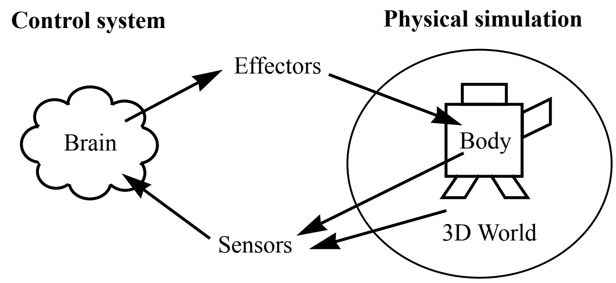


Figure 4: Cycle of effects between brain, body and world.

5.1 Sensors

Each sensor is contained within a specific part of the body, and measures either aspects of that part or aspects of the world relative to that part. Three different types of sensors were used for these experiments:

1. *Joint angle sensors* give the current value for each degree of freedom of each joint.

2. *Contact sensors* activate (1.0) if a contact is made, and negatively activate (-1.0) if not. Each contact sensor has a sensitive region within a part's shape and activates when any contacts occur in that area. In this work, contact sensors are made available for each face of each part. No distinction is made between self-contact and environmental contact.

3. *Photosensors* react to a global light source position. Three photosensor signals provide the coordinates of the normalized light source direction relative to the orientation of the part. Shadows are not simulated, so photosensors continue to sense a light source even if it is blocked. Photosensors for two independent colors are made available. The source of one color is located in the desirable cube, and the other is located at the center of mass of the opponent. This effectively allows evolving nervous systems to incorporate specific "cube sensors" and "opponent sensors."

Other types of sensors, such as accelerometers, additional proprioceptors, or even sound or smell detectors could also be implemented, but these basic three are enough to allow some interesting and adaptive behaviors to occur.

5.2 Neurons

Internal neural nodes are used to give virtual creatures the possibility of arbitrary behavior. They allow a creature to have an internal state beyond its sensor values, and be affected by its history.

In this work, different neural nodes can perform diverse functions on their inputs to generate their output signals. Because of this, a creature's brain might resemble a dataflow computer program more than a typical artificial neural network. This approach is probably less biologically realistic than just using sum and threshold functions, but it is hoped that it makes the evolution of interesting behaviors more likely. The set of functions that neural nodes can have is: *sum, product, divide, sum-threshold, greater-than, sign-of, min, max, abs, if, interpolate, sin, cos, atan, log, expt, sigmoid, integrate, differentiate, smooth, memory, oscillate-wave,* and *oscillate-saw.*

Some functions compute an output directly from their inputs, while others such as the oscillators retain some state and can give time varying outputs even when their inputs are constant. The number of inputs to a neuron depends on its function, and here is at most three. Each input contains a connection to another neuron or a sensor from which to receive a value. Alternatively, an input can simply receive a constant value. The input values are first scaled by weights before being operated on. The genetic parameters for each neural node include these weights as well as the function type and the connection information.

For each simulated time interval, every neuron computes its output value from its inputs. In this work, two brain time steps are performed for each dynamic simulation time step so signals can propagate through multiple neurons with less delay.

5.3 Effectors

Each effector simply contains a connection from a neuron or a sensor from which to receive a value. This input value is scaled by a constant weight, and then exerted as a joint force which affects the dynamic simulation and the resulting behavior of the creature. Different types of effectors, such as sound or scent emitters, might also be interesting, but only effectors that exert simulated muscle forces are used here.

Each effector controls a degree of freedom of a joint. The effectors for a given joint connecting two parts, are contained in the part further out in the hierarchy, so that each non-root part operates only a single joint connecting it to its parent. The angle sensors for that joint are also contained in this part.

Each effector is given a *maximum-strength* proportional to the maximum cross sectional area of the two parts it joins. Effector forces are scaled by these strengths and not permitted to exceed them. This is similar to the strength limits of natural muscles. As in nature, mass scales with volume but strength scales with area, so behavior does not always scale uniformly.

5.4 Combining Morphology and Control

The genotype descriptions of virtual brains and the actual phenotype brains are both directed graphs of nodes and connections. The nodes contain the sensors, neurons, and effectors, and the connections define the flow of signals between these nodes. These graphs can also be recurrent, and as a result the final control system can have feedback loops and cycles.

However, most of these neural elements exist within a specific part of the creature. Thus the genotype for the nervous system is a nested graph: the morphological nodes each contain graphs of the neural nodes and connections. Figure 5 shows an example of an evolved nested graph which describes a simple three-part creature as shown in figure 6.

When a creature is synthesized from its genetic description, the neural components described within each part are generated along with the morphological structure. This causes blocks of neural control circuitry to be replicated along with each instanced part, so each duplicated segment or appendage of a creature can have a similar but independent local control system.

These local control systems can be connected to enable the possibility of coordinated control. Connections are allowed between adjacent parts in the hierarchy. The neurons and effectors within a part can receive signals from sensors or neurons in their parent part or in their child parts.

Creatures are also allowed a set of neurons that are not associated with a specific part, and are copied only once into the phenotype. This gives the opportunity for the development of global synchronization or centralized control. These neurons can receive signals from each other or from sensors or neurons in specific instances of any of the creature's parts, and the neurons and effectors within the parts can optionally receive signals from these unassociated-neuron outputs.

In this way the genetic language for morphology and control is merged. A local control system is described for each type of part, and these are copied and connected into the hierarchy of the creature's body to make a complete distributed nervous system. Figure 6a shows the creature morphology resulting from the genotype in figure 5. Again, parameters describing shapes and weight values are not shown for the genotype even though they affect the pheno-

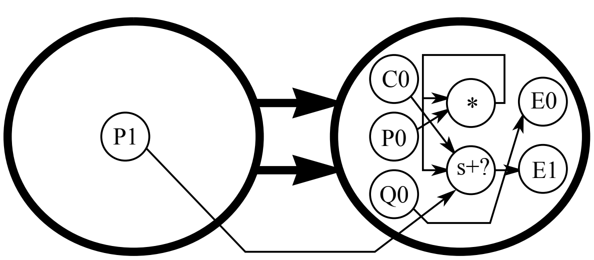


Figure 5: Example evolved nested graph genotype. The outer graph in bold describes a creature's morphology. The inner graph describes its neural circuitry. C0, P0, P1, and Q0 are contact and photosensors, E0 and E1 are effector outputs, and those labeled "*" and "s+?" are neural nodes that perform *product* and *sum-threshold* functions.

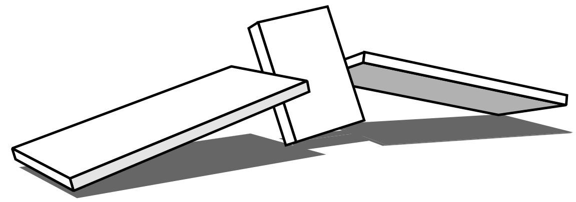


Figure 6a: The phenotype morphology generated from the evolved genotype shown in figure 5.

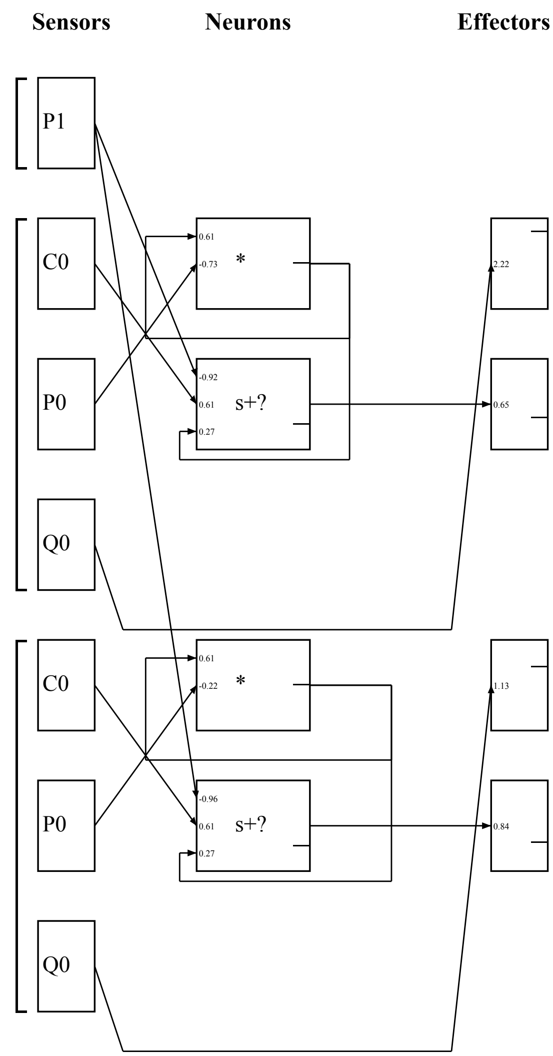


Figure 6b: The phenotype "brain" generated from the evolved genotype shown in figure 5. The effector outputs of this control system cause the morphology above to roll forward in tumbling motions.

type. Figure 6b shows the corresponding brain of this creature. The brackets on the left side of figure 6b group the neural components of each part. Two groups have similar neural systems because they were synthesized from the same genetic description. This creature can roll over the ground by making cyclic tumbling motions with its two arm-like appendages. Note that it can be difficult to analyze exactly how a control system such as this works, and some components may not actually be used at all. Fortunately, a primary benefit of using artificial evolution is that understanding these representations is not necessary.

6 Physical Simulation

Dynamics simulation is used to calculate the movement of creatures resulting from their interaction with a virtual three-dimensional world. There are several components of the physical simulation used in this work: articulated body dynamics, numerical integration, collision detection, and collision response with friction. These are only briefly summarized here, since physical simulation is not the emphasis of this paper.

Featherstone's recursive O(N) articulated body method is used to calculate the accelerations from the velocities and external forces of each hierarchy of connected rigid parts [5]. Integration determines the resulting motions from these accelerations and is performed by a Runge-Kutta-Fehlberg method which is a fourth order Runge-Kutta with an additional evaluation to estimate the error and adapt the step size. Typically between 1 and 5 integration time steps are performed for each frame of 1/30 second.

The shapes of parts are represented here by simple rectangular solids. Bounding box hierarchies are used to reduce the number of collision tests between parts from $O(N^2)$. Pairs whose world-space bounding boxes intersect are tested for penetrations, and collisions with a ground plane are also tested. If necessary, the previous time-step is reduced to keep any new penetration depths below a certain tolerance. Connected parts are permitted to interpenetrate but not rotate completely through each other. This is achieved by using adjusted shapes when testing for collisions between connected parts. The shape of the smaller part is clipped halfway back from its point of attachment so it can swing freely until its remote end makes contact.

Collision response is accomplished by a hybrid model using both impulses and penalty spring forces. At high velocities, instantaneous impulse forces are used, and at low velocities springs are used, to simulate collisions and contacts with arbitrary elasticity and friction parameters.

It is important that the physical simulation be reasonably accurate when optimizing for creatures that can move within it. Any bugs that allow energy leaks from non-conservation, or even round-off errors, will inevitably be discovered and exploited by the evolving creatures. Although this can be a lazy and often amusing approach for debugging a physical modeling system, it is not necessarily the most practical.

7 Creature Evolution

An evolution of virtual creatures is begun by first creating an initial population of genotypes. Seed genotypes are synthesized "from scratch" by random generation of sets of nodes and connections. Alternatively, an existing genotype from a previous evolution can be used to seed an initial population.

Before creatures are paired off for competitions and fitness evaluation, some simple viability checks are performed, and inappropriate creatures are removed from the population by giving them zero fitness values. Those that have more than a specified number of parts are removed. A subset of genotypes will generate creatures whose parts initially interpenetrate. A short simulation with collision detection and response attempts to repel any intersecting parts, but those creatures with persistent interpenetrations are also discarded.

A *survival-ratio* determines the percentage of the population that will survive each generation. In this work, population sizes were typically 300, and the survival-ratio was 1/5. If the initially generated population has fewer individuals with positive fitness than the number that should survive, another round of seed genotypes is generated to replace those with zero fitness.

For each generation, creatures are grown from their genotypes, and their fitness values are measured by simulating one or more competitions with other individuals as described. The individuals whose fitnesses fall within the survival percentile are then reproduced, and their offspring fill the slots of those individuals that did not survive. The number of offspring that each surviving individual generates is proportional to its fitness. The survivors are kept in the population for the next generation, and the total size of the population is maintained. In multi-species evolutions, each sub-population is independently treated in this way so the number of individuals in each species remains constant and species do not die out.

Offspring are generated from the surviving creatures by copying and combining their directed graph genotypes. When these graphs are reproduced they are subjected to probabilistic variation or mutation, so the corresponding phenotypes are similar to their parents but have been altered or adjusted in random ways.

7.1 Mutating Directed Graphs

A directed graph is mutated by the following sequence of steps:

1. The internal parameters of each node are subjected to possible alterations. A mutation frequency for each parameter type determines the probability that a mutation will be applied to it at all. Boolean values are mutated by simply flipping their state. Scalar values are mutated by adding several random numbers to them for a Gaussian-like distribution

so small adjustments are more likely than drastic ones. The scale of an adjustment is relative to the original value, so large quantities can be varied more easily and small ones can be carefully tuned. A scalar can also be negated. After a mutation occurs, values are clamped to their legal bounds. Some parameters that only have a limited number of legal values are mutated by simply picking a new value at random from the set of possibilities.

2. A new random node is added to the graph. A new node normally has no effect on the phenotype unless a connection also mutates a pointer to it. Therefore a new node is always initially added, but then garbage collected later (in step 5) if it does not become connected. This type of mutation allows the complexity of the graph to grow as an evolution proceeds.

3. The parameters of each connection are subjected to possible mutations in the same way the node parameters were in step 1. With some frequency the connection pointer is moved to point to a different node which is chosen at random.

4. New random connections may be added and existing ones may be removed. In the case of the neural graphs these operations are not performed because the number of inputs for each element is fixed, but the morphological graphs can have a variable number of connections per node. Each existing node is subject to having a new random connection added to it, and each existing connection is subject to possible removal.

5. Unconnected elements are garbage collected. Connectedness is propagated outwards through the connections of the graph, starting from the root node of the morphology, and from the effector nodes of the neural graphs. Although leaving the disconnected nodes for possible reconnection might be advantageous, and is probably biologically analogous, at least the unconnected newly added ones are removed to prevent unnecessary growth in graph size.

Since mutations are performed on a per element basis, genotypes with only a few elements might not receive any mutations, where genotypes with many elements would receive enough mutations that they would rarely resemble their parents. This is compensated for by scaling the mutation frequencies by an amount inversely proportional to the size of the current graph being mutated, such that on the average at least one mutation occurs in the entire graph.

Mutation of nested directed graphs, as are used here to represent creatures, is performed by first mutating the outer graph and then mutating the inner layer of graphs. The inner graphs are mutated last because legal values for some of their parameters (inter-node neural input sources) can depend on the topology of the outer graph.

7.2 Mating Directed Graphs

Sexual reproduction allows components from more than one parent to be combined into new offspring. This permits features to evolve independently and later be merged into a single individual. Two different methods for mating directed graphs are used in this work.

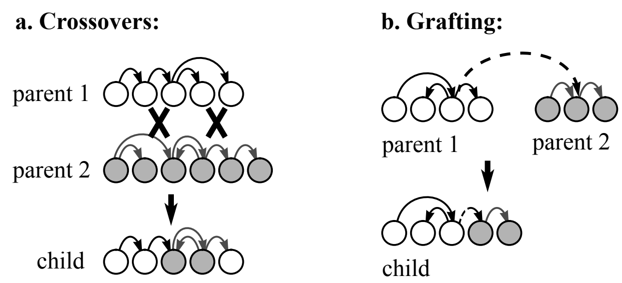


Figure 7: Two methods for mating directed graphs.

The first is a *crossover* operation (figure 7a). The nodes of two parents are each aligned in a row as they are stored, and the nodes of the first parent are copied to make the child, but one or more crossover points determine when the copying source should switch to the other parent. The connections of a node are copied with it and simply point to the same relative node locations as before. If the copied connections now point out of bounds because of varying node numbers they are randomly reassigned.

A second mating method *grafts* two genotypes together by connecting a node of one parent to a node of another (figure 7b). The first parent is copied, and one of its connections is chosen at random and adjusted to point to a random node in the second parent. Newly unconnected nodes of the first parent are removed and the newly connected node of the second parent and any of its descendants are appended to the new graph.

A new directed graph can be produced by either of these two mating methods, or asexually by using only mutations. Offspring from matings are sometimes subjected to mutations afterwards, but with reduced mutation frequencies. In this work a reproduction method is chosen at random for each child to be produced by the surviving individuals using the ratios: 40% asexual, 30% crossovers, and 30% grafting. A second parent is chosen from the survivors if necessary, and a new genotype is produced from the parent or parents.

After a new generation of genotypes is created, a phenotype creature is generated from each, and again their fitness values are evaluated. As this cycle of variation and selection continues, the population is directed towards creatures with higher fitness.

7.3 Parallel Implementation

This process has been implemented to run in parallel on a Connection Machine® CM-5 in a master/slave message passing model. A single processing node contains the population and performs all the selection and reproduction operations. It farms out pairs of genotypes to the other nodes to be fitness tested, and gathers back the fitness values after they have been determined. The fitness tests each include a dynamics simulation for the competition and although many can execute in nearly real-time, they are still the dominant computational requirement of the system. Performing a fitness test per processor is a simple but effective way to parallelize this process, and the overall performance scales quite linearly with the number of processors, as long as the population size is somewhat larger than the number of processors.

Each fitness test takes a different amount of time to compute depending on the complexity of the creatures and how they attempt to move. To prevent idle processors from just waiting for others to finish, the slowest few simulations at the end of a generation are suspended and those individuals are removed from the population by giving them zero fitness. With this approach, an evolution with population size 300, run for 100 generations, might take about four hours to complete on a 32 processor CM-5.

8 Results and Discussion

Many independent evolutions were performed using the "all vs. best" competition pattern as described in section 3. Some single-species evolutions were performed in which all individuals both compete and breed with each other, but most included two species where individuals only compete with members of the opponent species.

Some examples of resulting two-species evolutionary dynamics are shown in Figure 8. The relative fitness of the best individuals of each species are plotted over 100 generations. The rate of evolutionary progress varied widely in different runs. Some species took many generations before they could even reach the cube at all, while others discovered a fairly successful strategy in the first 10 or 20 generations. Figure 8c shows an example where one species was successful fairly quickly and the other species never evolved an effective strategy to challenge it. The other three graphs in figure 8 show evolutions where more interactions occurred between the evolving species.

A variety of methods for reaching the cube were discovered. Some extended arms out onto the cube, and some reached out while falling forward to land on top of it. Others could crawl inch-worm style or roll towards the cube, and a few even developed leg-like appendages that they used to walk towards it.

The most interesting results often occurred when both species discovered methods for reaching the cube and then further evolved strategies to counter the opponent's behavior. Some creatures pushed their opponent away from the cube, some moved the cube away from its initial location and then followed it, and others simply covered up the cube to block the opponent's access. Some counter-strategies took advantage of a specific weakness in the original strategy and could be easily foiled in a few generations by a minor adaptation to the original strategy. Others permanently defeated the original strategy and required the first species to evolve another level of counter-counter-strategy to regain the lead.

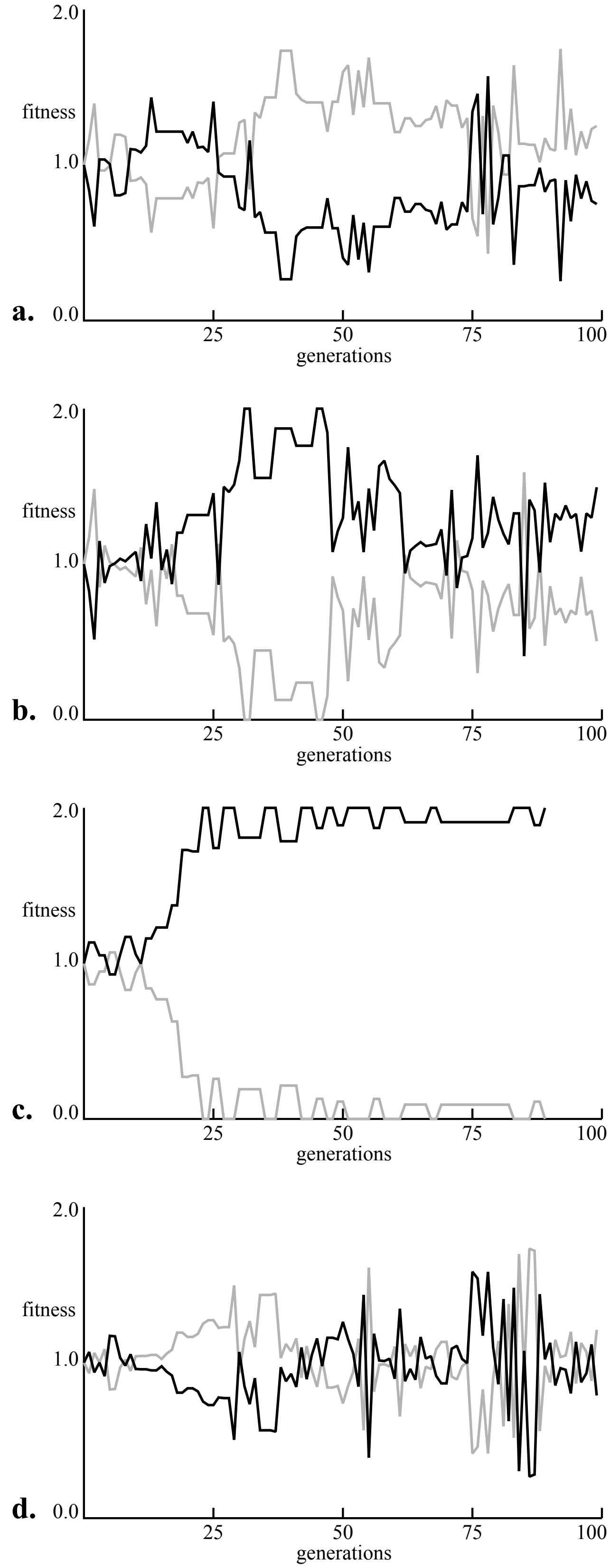


Figure 8: Relative fitness between two co-evolving and competing species, from four independent simulations.

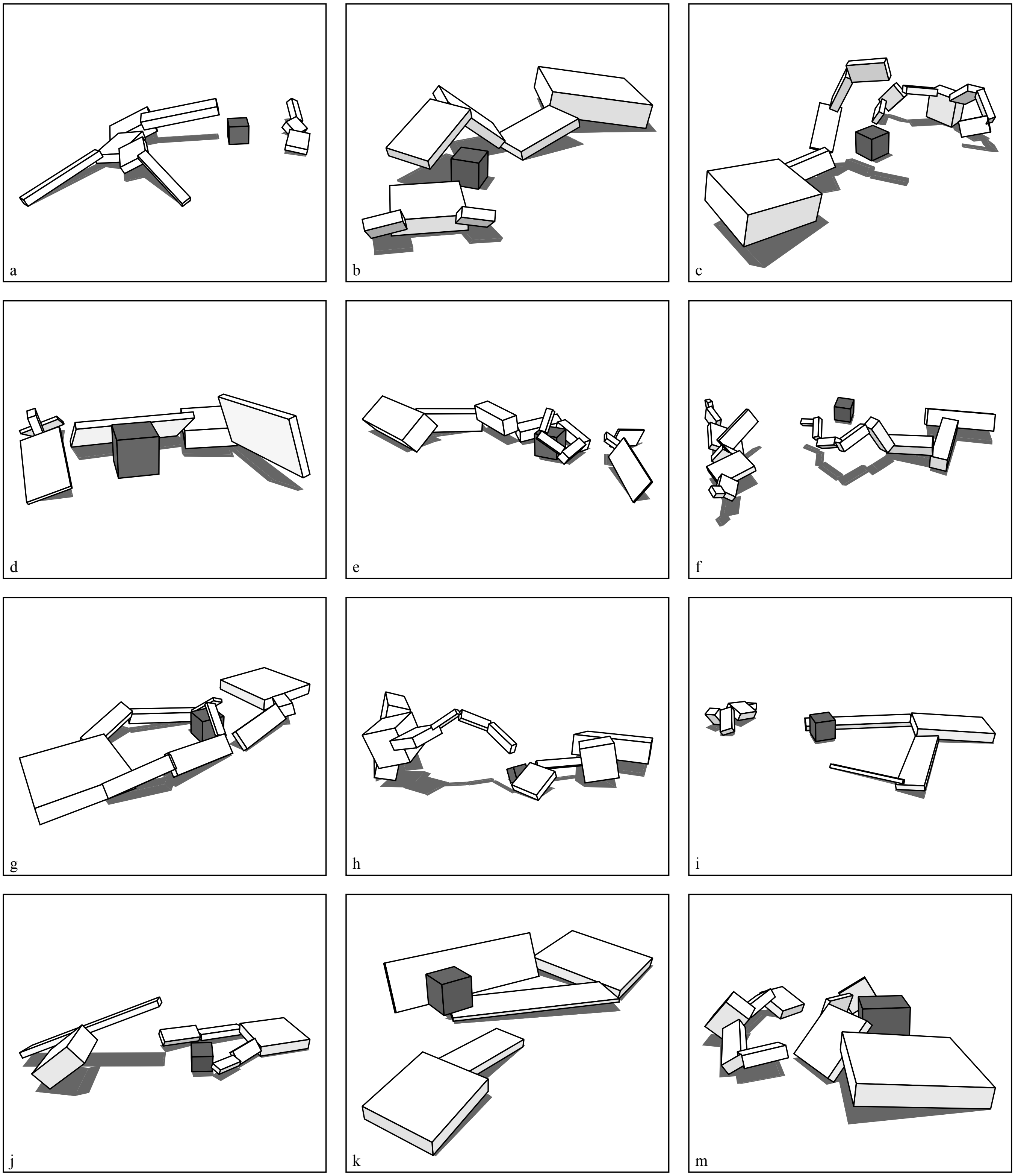


Figure 9: Evolved competing creatures.

In some evolutions the winners alternated between species many times with new strategies and counter-strategies. In other runs one species kept a consistent lead with the other species only providing temporary challenges.

After the results from many simulations were observed, the best were collected and then played against each other in additional competitions. The different strategies were compared, and the behavior and adaptability of creatures were observed as they faced new types of opponents that were not encountered during their evolutions. A few evolutions were also performed starting with an existing creature as a seed genotype for each species so they could further evolve to compete against a new type of opponent.

Figure 9 shows some examples of evolved competing creatures and demonstrates the diversity of the different strategies that emerged. Some of the behaviors and interactions of these specific creatures are described briefly here. The larger creature in figure 9b nudges the cube aside and then pins down his smaller opponent. The crab-like creature in 9c can successfully walk forward, but then continues blindly past the cube and over the opponent. Figure 9d shows a creature that has just pushed its opponent away from the cube, and the arm-like creature in 9e also jabs at its opponent before curling around the cube.

Most creatures perform similar behavior independently of the opponent's actions, but a few are adaptive in that they can reach towards the cube wherever it moves. For example the arm-like creature in figure 9f pushes the cube aside and then uses photosensors to adaptively follow it. If its opponent moves the cube in a different direction it will successfully grope towards the new location.

The two-armed creature in figure 9g blocks access to the cube by covering it up. Several other two-armed creatures in 9i, 9j, and 9k use the strategy of batting the cube to the side with one arm and catching it with the other arm. This seemed to be the most successful strategy of the creatures in this group, and the one in 9k was actually the overall winner because it could whisk the cube aside very quickly. However, it was a near tie between this and the photosensitive arm in 9f. The larger creature in 9m wins by a large margin against some opponents because it can literally walk away with the cube, but it does not initially reach the cube very quickly and tends to loose against faster opponents.

It is possible that adaptation on an evolutionary scale occurred more easily than the evolution of individuals that were themselves adaptive. Perhaps individuals with adaptive behavior would be significantly more rewarded if evolutions were performed with many species instead of just one or two. To be successful, a single individual would then need to defeat a larger number of different opposing strategies.

9 Future Work

Several variations on this system could be worth further experimentation. Other types of contests could be defined in which creatures compete in different environments and different rules determine the winners. Creatures might also be rewarded for cooperative behavior somehow as well as competitive, and teams of interacting creatures could be simulated.

Evolutions containing larger numbers of species should certainly be performed, with the hope of increasing the chances for emergence of more adaptive individuals as hypothesized above.

An additional extension to this work would be to simulate a more complex but more realistic environment in which many creatures simultaneously compete and/or cooperate with each another, instead of pairing off in one-on-one contests. Speciation, mating patterns, competing patterns, and even offspring production could all be determined by one long ecological simulation. Experiments like this have been performed with simpler organisms and have produced interesting results including specialization and various social interactions [18,24].

Perhaps the techniques presented here should be considered as an approach toward creating artificial intelligence. When a genetic language allows virtual entities to evolve with increasing complexity, it is common for the resulting system to be difficult to understand in detail. In many cases it would also be difficult to design a similar system using traditional methods. Techniques such as these have the potential of surpassing those limits that are often imposed when human understanding and design is required. The examples presented here suggest that it might be easier to evolve virtual entities exhibiting intelligent behavior than it would be for humans to design and build them.

10 Conclusion

In summary, a system has been described that can automatically generate autonomous three-dimensional virtual creatures that exhibit diverse competitive strategies in physically simulated worlds. A genetic language that uses directed graphs to describe both morphology and behavior defines an unlimited hyperspace of possible results, and a variety of interesting virtual creatures have been shown to emerge when this hyperspace is explored by populations of evolving and competing individuals.

Acknowledgments

Thanks to Gary Oberbrunner and Matt Fitzgibbon for Connection Machine and software support. Thanks to Thinking Machines Corporation and Lew Tucker for supporting this research. Thanks to Bruce Blumberg and Peter Schröder for dynamic simulation help and suggestions. And special thanks to Pattie Maes.

References

1. Angeline, P.J., and Pollack, J.B., "Competitive Environments Evolve Better Solutions for Complex Tasks," in *Proceedings of the 5th International Conference on Genetic Algorithms*, ed. by S. Forrest, Morgan Kaufmann 1993, pp.264-270.

2. Axelrod, R., "Evolution of Strategies in the Iterated Prisoner's Dilemma", in *Genetic Algorithms and Simulated Annealing*, ed. by L. Davis, Morgan Kaufmann, 1989.

3. Cramer, N.L., "A Representation for the Adaptive Generation of Simple Sequential Programs," *Proceedings of the First International Conference on Genetic Algorithms*, ed. by J. Grefenstette, 1985, pp.183-187.

4. Dawkins, R., *The Blind Watchmaker*, Harlow Longman, 1986.

5. Featherstone, R., *Robot Dynamics Algorithms*, Kluwer Academic Publishers, Norwell, MA, 1987.

6. de Garis, H., "Genetic Programming: Building Artificial Nervous Systems Using Genetically Programmed Neural Network Modules," *Proceedings of the 7th International Conference on Machine Learning*, 1990, pp.132-139.

7. Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.

8. Hart, J., "The Object Instancing Paradigm for Linear Fractal Modeling," *Graphics Interface*, 1992, pp.224-231.

9. Hillis, W.D., "Co-evolving parasites improve simulated evolution as an optimization procedure," *Artificial Life II*, ed. by Langton, Taylor, Farmer, & Rasmussen, Addison-Wesley, 1991, pp313-324.

10. Holland, J.H., *Adaptation in Natural and Artificial Systems*, Ann Arbor, University of Michigan Press, 1975.

11. Kitano, H., "Designing neural networks using genetic algorithms with graph generation system," *Complex Systems*, Vol.4, pp.461-476, 1990.

12. Koza, J., *Genetic Programming: on the Programming of Computers by Means of Natural Selection*, MIT Press, 1992.

13. Lindenmayer, A., "Mathematical Models for Cellular Interactions in Development, Parts I and II," *Journal of Theoretical Biology*, Vol.18, 1968, pp.280-315.

14. Lindgren, K., "Evolutionary Phenomena in Simple Dynamics," in *Artificial Life II*, ed. by Langton, Taylor, Farmer, & Rasmussen, Addison-Wesley, 1991, pp.295-312.

15. Mjolsness, E., Sharp, D., and Alpert, B., "Scaling, Machine Learning, and Genetic Neural Nets," *Advances in Applied Mathematics*, Vol.10, 1989, pp.137-163.

16. Ngo, J.T., and Marks, J., "Spacetime Constraints Revisited," *Computer Graphics*, Annual Conference Series, 1993, pp.343-350.

17. van de Panne, M., and Fiume, E., "Sensor-Actuator Networks," *Computer Graphics*, Annual Conference Series, 1993, pp.335-342.

18. Ray, T., "An Approach to the Synthesis of Life," *Artificial Life II*, ed. by Langton, Taylor, Farmer, & Rasmussen, Addison-Wesley, 1991, pp.371-408.

19. Reynolds, C., "Competition, Coevolution and the Game of Tag," to be published in: *Artificial Life IV Proceedings*, ed. by R. Brooks & P. Maes, MIT Press, 1994.

20. Sims, K., "Artificial Evolution for Computer Graphics," *Computer Graphics*, Vol.25, No.4, July 1991, pp.319-328.

21. Sims, K., "Interactive Evolution of Dynamical Systems," *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, ed. by Varela, Francisco, & Bourgine, MIT Press, 1992, pp.171-178.

22. Sims, K., "Evolving Virtual Creatures," *Computer Graphics,* Annual Conference Series, July 1994, pp.15-22.

23. Smith, A.R., "Plants, Fractals, and Formal Languages," *Computer Graphics*, Vol.18, No.3, July 1984, pp.1-10.

24. Yaeger, L., "Computational Genetics, Physiology, Metabolism, Neural Systems, Learning, Vision, and Behavior or PolyWorld: Life in a New Context," *Artificial Life III*, ed. by C. Langton, Santa Fe Institute Studies in the Sciences of Complexity, Proceedings Vol. XVII, Addison-Wesley, 1994, pp.263-298.

Automatic Creation of an Autonomous Agent: Genetic Evolution of a Neural-Network Driven Robot

Dario Floreano
Laboratory of Cognitive Technology
AREA Science Park
Trieste, Italy
dario@psicosun.univ.trieste.it

Francesco Mondada
Laboratory of Microcomputing
Swiss Federal Institute of Technology
Lausanne, Switzerland
mondada@di.epfl.ch

Abstract

The paper describes the results of the evolutionary development of a real, neural-network driven mobile robot. The evolutionary approach to the development of neural controllers for autonomous agents has been successfully used by many researchers, but most -if not all- studies have been carried out with computer simulations. Instead, in this research the whole evolutionary process takes places *entirely* on a real robot without human intervention. Although the experiments described here tackle a simple task of navigation and obstacle avoidance, we show a number of emergent phenomena that are characteristic of autonomous agents. The neural controllers of the evolved best individuals display a full exploitation of non-linear and recurrent connections that make them more efficient than analogous man-designed agents. In order to fully understand and describe the robot behavior, we have also employed quantitative ethological tools [13], and showed that the adaptation dynamics conform to predictions made for animals.

1 Introduction

A mechanical device that can operate without being attached to a power supply or an external computer is not necessarily an autonomous robot. Although this may be an additional desirable feature, autonomous robots are rather identified by their ability to adapt to an environment by finding optimal solutions, develop a suitable control system, define their own goals, and, possibly, perform some self-monitoring [19]. All these capabilities cannot be pre-defined, but should rather emerge from the interaction between the robot and its own environment. A possible solution for building autonomous systems consists in using simple components and primitive structures for the control system; in this case, articulated and complex behaviors would be the spontaneous result of the interactions among all these parts through a process of self-organization guided by a continuous exchange of information with the environment. Major steps in this direction have already been taken. Brooks's subsumption architecture [4] is indeed a case of constructive, bottom-up approach toward building autonomous robots that display emergent behaviors. His approach consists of providing the robot with a set of simple behaviors; further behavior-modules can be added on the top of these primitives and connected to them via simple excitatory or inhibitory links. A similar approach has been formulated by Steels [19], who is pursuing the goal of building intelligent agents by focusing on action-centered skills, autonomy, behavior-oriented decomposition, emergent functionality, and layered architectures. In a more general context, Maes [11] has tried to define the theory, methodology, and goals of a new Behavior-Based Artificial Intelligence, as contrasted to the Knowledge-Based Artificial Intelligence. Beside these solutions, some other researchers have fulfilled the requirements of learning and adaptation by employing various sorts of neural networks to control a robotic system [2], [20]; whether pre-wired or plastic, these neural controllers exhibit characteristics of generalization, flexibility, robustness, and, possibly, plastic adaptation. All these features are indeed important prerequisites of autonomous agents. A somehow different step toward design automatization of autonomous robots is taken by those researchers that try to evolve the robot control system. Rather than starting from a designed solution, they describe the primitives of the robot in the form of an artificial chromosome, build many of these chromosome with some random arrangement of the genes, test the control system generated with every chromosome on a robot, select and reproduce only those chromosomes that guarantee the robot a better fitness according to some survival criterion; this process is repeated until the average population performance is good enough or some mutant with exceptional characteristics is born. Although the evolutionary procedure [9], [7] is well known to a vast community of researchers, it is not a straightforward task to apply it to real robots, as we will see later. Our work concerns the evolution of a neural-network-controlled mobile robot. What is really important in our experiments is that the whole evolutionary process takes places *entirely* on a real robot without human intervention. Before going into the description of our results and the following discussion, let us stress two points that we think to be of general relevance, namely the choice of a neural architecture and the role of simulations versus real implementations.

2 Neural Architectures

Artificial neural networks seem to us to be particularly good candidates for the control system of artificial autonomous agents because they possess many desirable features required by the principles of autonomy in real environments (see also [8]). Let us list some of these properties.

- Neural networks are flexible. The ability to learn enables dynamic adaptation of robot behavior to changes in the environment. Even when the synapses are not modifiable, a neural network still exhibits a reasonable degree of flexibility, i.e., it is able to produce appropriate behaviors in response to a range of possible variations of the physical stimulation.
- Artificial neural networks are robust: missing links or malfunctioning of some hardware components do not strongly impair the robot's behavior.
- A neural network deals with the micro-structure of the robot: this means that it can either shape its own structure to exploit at its best the sensory-motor features of the robot [5], or actively select and use only those sensors and motors that are best suited for performing the task [16].
- The well known tolerance to noise (in some cases noise enhances performance [17], or is an essential component for learning, such as in self-organizing neural networks) makes them good candidates for mediating between physical sensors and actuators with intrinsic noise.
- If we do not put limits to the network architecture, and thus have recurrent and lateral connections, and non-linear transfer functions, we have a potentially powerful device that could cope with the temporal structure and complex mappings required by real-world operations.

Finally, artificial neural networks are well-suited structures also for artificial evolution. Small changes in a neural network usually correspond to small changes in its behavior, at least for feed-forward architectures. Genetic algorithms find their way toward a maximum by sampling new solutions obtained by random combinations and mutations, and thus take advantage of the intrinsic "gradualism" of the neural network structure.

3 Simulation versus Implementation

There is currently a hot debate among people trying to understand and reproduce intelligent agents, that could be stated as follows: "Is the simulation a powerful enough tool to draw sound conclusions, or should a theory or an approach be tested on a real agent, i.e., a robot¿' Although both numerical simulations and physical implementations have their own merits in different fields of research, the issue becomes important when we investigate autonomous and intelligent agents. Let us examine in more detail the respective advantages and drawbacks of these two methodologies in our particular case. It is usually argued that computer simulations are fast. High performance serial machines and massively parallel computers nowadays are powerful tools for the virtual reproduction and analysis of complex-system dynamics. In a few days of computation the scientist can reproduce birth and death of whole populations of organisms (see, e.g., [1]). But this holds only to a limited level of sophistication. It is still much faster to have a real camera acquiring images from a real world than simulating the world, the camera, and the image acquisition process (see [8]). This is not a problem of "bottlenecks", but it is due to the fact that enormous calculations are sometimes necessary for simulating a very trivial [1] physical phenomenon, partly because computers are general-purpose machines whereas natural devices are "dedicated hardware". Another common belief is that computer simulations are cheaper. The researcher may think that it is worth exploring a hypothesis or a new algorithm by computer simulations before investing money and time in a robot. Although this may be true in many situations, in some cases it is not. It all depends on the degree of plausibility and "reality" of the simulation. If the standard is intended to be high, then it is very likely that it will involve one or more specialized programmers on the project for many months. Sometimes, this may cost more than building, purchasing, or modifying real robots. It is widely accepted that numerical simulations allow complete control and record of all the variables; it is thus possible to replicate results, analyze phenomena, accelerate or slow down processes. This is certainly true. But why should we have complete control over an autonomous agent? After all, an artificial agent will never be truly autonomous while there is an umbilical chord that limits its field of action Autonomous agents living within a computer are limited by the necessarily-predefined number of experiences and levels of interactions with the environment. If we are to build intelligent autonomous agents, then we will have to

[1] Here "trivial" is meant in a naive sense, as opposed to number of processes or complexity of the dynamics involved

give up -sooner or later- with the obsession of controlling and replicating every possible accident. Computer simulations are still a powerful tool for our research. They provide a viable solution for experimenting non-existing devices or non feasible (with physical tools) hypotheses about the nature and characteristics of artificial agents. "Life as it could be" is indeed one of the major topics of Artificial Life [10], a field of investigation that much inspires research in autonomous agents. Computers leave us free to use our imagination and test the most bizarre hypotheses to recreate new autonomous organisms living and behaving within worlds with different physical laws. However, when we simulate something we must always be aware that we are putting some constraint somewhere at some level. It is not anymore the real world that we are dealing with. And this may be a crucial point when trying to create an autonomous system. By definition, an autonomous agent itself will define the level of interaction with its own environment and alone will choose the relevant information to take into consideration. If we are to restrict at some point the range of available possibilities, we may hamper or greatly reduce the potentiality of our agent. One of the strongest critics made against the simulative approach is that numerical simulations do not consider all the physical laws of the interaction of a real agent with its own environment, such as mass, weight, friction, inertia, etc. Although this may be questionable, it is certainly true that simulations do not take into account Murphy's Laws, such as malfunctioning, component failures, and consumption that govern both artificial and biological organisms. Finally, a real danger with computer simulations is that it cannot be guaranteed that a transfer to the implementation phase will be smooth, if feasible at all. But, let us imagine this to be possible. Who will guarantee to us, then, that the robot is actually doing what it was doing in the simulations? How to compare precise numerical values with behavioral data collected in a noisy world? This is especially important for those researchers who develop the control system of the robot with a computer simulation, and then "inject" the resulting "brain" into the processor of their physical agent and leave it free to move. The analysis and discussion of the reasons why one method should be preferred over the other may take much longer; here, we have only tried to outline a few important topics that we felt relevant for our methodology.

3.1 Evolutionary Development of a Physical Robot

But, for what concerns our specific research, there is a more compelling question. Why are several groups working on simulations, but it is hard -or even impossible- to find cases of generational development of populations of real robots, that is, robots that must survive in a real world on the basis of some fitness criterion, where only the fittest can mate and reproduce through a generational and cyclic process? We believe that the reason is not the cost and waste of material (not fitted robots), or difficulties with the mating procedure, but it is rather based on the construction principles of robots. Most of the available robots are not suited for evolution, in terms of mechanical robustness, design concepts, and automatic evaluation of the robot performance:

- Evolution (Genetic Algorithms) takes a long time; it may require hours, days, weeks, or even months, of continuous functioning of the hardware. Most of available robots tend to break down in these conditions and are not capable of self-repair, as biological organisms often do.

- The common philosophy underlying the construction of robots designed for operating in autonomy dictates that many precise and sophisticated sensory devices should be mounted on the main board. The mechanical solutions for moving around and performing other actions are taken either from well established engineering solutions (three wheel synchronous drive, for instance), or from successful biological organisms (stick insects, ants, etc.). This leads to the construction of complex, highly structured, and fragile mechanisms. For this reason, such robots would easily get trapped in corners and local minima during the first generations of the evolutionary process. Whereas there is no reason in principle why the evolutionary technique should not be applicable to complex robots (and indeed it will have to, at least to some extent), it is definitely true that biological evolution did not start with a structured and sophisticated body coupled with a virtually non-existing brain. Evolutionary studies have shown that there is a gradual coevolution of body and "mind" in biological organisms. Thus, either we start with a robot designed with new principles (simple components and geometry, robust and reliable hardware, only necessary and elementary sensors and actuators), or we provide a complex robot with a set of "basic instincts" (but which?) and let evolution work on higher control structures. We have chosen the first approach because we consider it to be chronologically and logically the first thing to try, and also because the second solution, although viable in principle, may still be problematic at this stage.

- In order to get a sensible behavior out of a *tabula rasa* (whatever type of architecture we use), Genetic Algorithms require a fitness function, i.e., a survival criterion against which each individual of the population is tested. As long as the artificial agent is a virtual entity within a computer it is fairly easy to precisely evaluate its performance. However, when the agent takes form into a physical and mobile body free to wander in our world, automatic fitness evaluation becomes a non-trivial task. We will take into consideration this issue in a later section.

4 Navigation and Obstacle Avoidance

Because of all the reasons outlined in the section above, we were not certain that the evolutionary approach would have worked with a real robot. Mainly, we did not know how to assess and compare the results that we could have obtained with this approach. Thus, we have decided to start from a classic task, a sort of exercise and test for all people working with mobile robots. The robot had to learn to move in an environment and avoid obstacles. For its simplest formulation, there is already a well-known, optimal, and simple distributed solution for this task: the Braitenberg's vehicle [3], with which we have compared our results. Thus, we have put our robot in an arbitrary environment, set a few parameters concerning the fitness function and the network structure, and let it free to evolve. We have run this experiment many times in order to obtain reliable data and draw sound conclusions. Each time, we have kept track of a some relevant variables during the evolutionary process, analyzed the best organisms, and compared the solutions obtained by evolution with those designed by man.

4.1 Experimental Setup

The robot employed in our experiments is Khepera, a miniature mobile robot [14]. Khepera has many of the characteristics required by the evolutionary approach to autonomous robots. It has a circular shape (Figure 1), with diameter of 55 mm, height of 30 mm, and weight of 70 g, and is supported by two wheels and two small Teflon balls.

The wheels are controlled by two DC motors with incremental encoder (10 pulses per mm of advancement of the robot), and can move in both directions. The simple

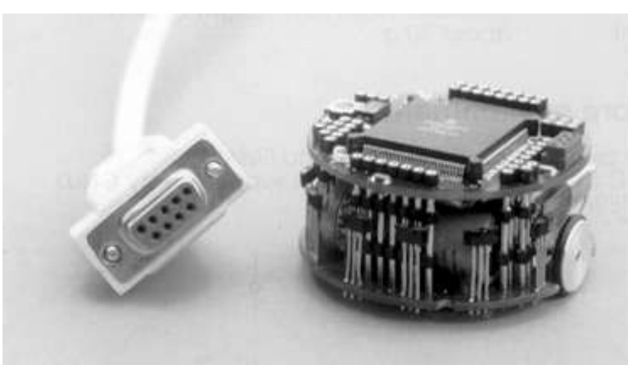


Figure 1: Khepera, the miniature mobile robot.

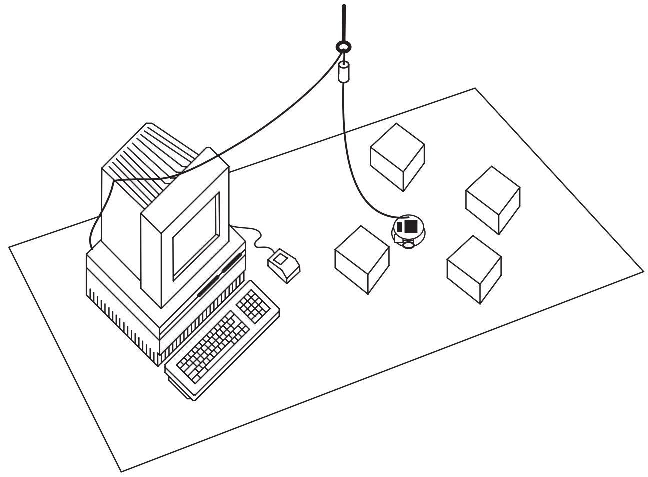


Figure 2: Operating methodology.

geometrical shape and the motor layout allow Khepera to potentially negotiate any type of obstacle and corner. These characteristics, together with many other mechanical solutions, have resulted in a robot that has continuously and reliably operated for weeks and weeks, most of the time crashing into walls and obstacles (due to the functioning principles of Genetic Algorithms). In the basic configuration used here, the robot is provided with eight Infra-Red proximity sensors. Six sensors are positioned on one side of the robot (front), the remaining two on the other side (back). A Motorola 68331 controller with 256 Kbytes of RAM and 512 Kbytes ROM manages all the input-output routines and can communicate via a serial port with a host computer.

Because of its size and design principles, Khepera is well-suited for laboratory experiments. Its communication protocol can exploit all the power and disk size available in a workstation by letting high-level control processes (genetic operators, neural network activation, variables recordings) run on the main station while low-level processes (sensor-reading, motor control, and other real time tasks) run on the on-board processor (Figure 2).

We have adopted this solution for our experiments. Khepera was attached via a serial port to a Sun SparcStation 2 by means of a lightweight aerial cable and specially designed rotating contacts. In this way, while the robot was running, we could keep track of all the populations of organisms that were born, tested, and passed to the genetic operators, together with their "personal life files". At the same time, we could also take advantage of specific software designed for graphic visualization of trajectories and sensory-motor status while the robot was evolving. Skeptics should not consider this methodology as an attempt on the very hearth of autonomy: as stated in the very beginning of this paper, running with

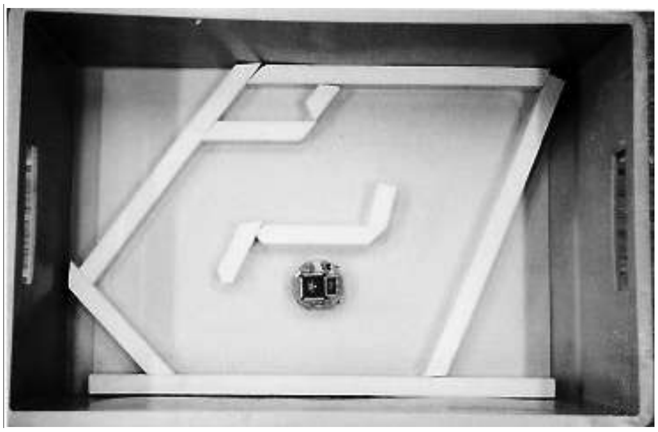


Figure 3: Environment of the experiment.

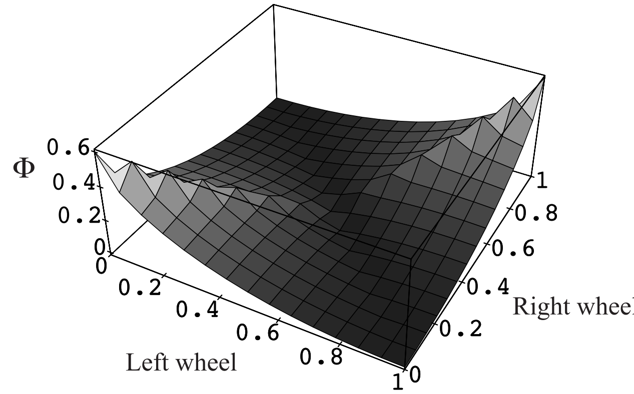


Figure 4: Function surface for i = 0.4. Wheel speed values have already been transformed into positive range where 0.5 is the point of direction inversion. Please note that this is not a full picture of the fitness function maximized by the genetic algorithm, which is instead n-dimensional (n = number of neural network free parameters). Furthermore, it does not take into account the physical characteristics of the environment.

its own batteries is only an optional feature of an autonomous agent. For what concerns Khepera, the robot is not aware of where its own "brain" is located, and this is indeed not important in this experiment of navigation and obstacle avoidance. However, it should be noted that the software that implements the genetic development of neural networks [6] could be slimmed down and downloaded into the robot processor.

The robot was put in an environment consisting in a sort of circular corridor whose external size was approx. 80x50 cm large (Figure 3). The walls were made of light-blue polystyrene and the floor was a gray thick paper. The robot could sense the walls with the IR proximity sensors. Since the corridors were rather narrow (8-12 cm), some sensors were slightly active most of the time. The environment was within a portable box positioned in a room always illuminated from above by a 60-watt bulb light. A serial cable connected the robot to the workstation in our office, a few rooms away from it. Our goal was to develop a robot that could learn to maximize some sort of exploration measure while accurately avoiding all the obstacles on its way. This statement was also the base for the fitness criterion used in the experiments. One of the desirable features of autonomous robots is the independence from an external operator, also during the development process of the control system. This would mean that the performance criterion for an autonomous agent should rely solely on a set of variables that can be measured within the frame of interaction between the robot and the environment. If this constraint is satisfied, we achieve a practical advantage, because the robot could eventually learn to operate in any environment by a continuous self-assessment of its own performance without external controllers. Hence, our fitness criterion Φ was function of three variables, directly measured on the robot, as follows,

$$\Phi = V\left(1-\sqrt{\Delta v}\right)(1-i) \tag{1}$$

$$\begin{array}{ccccc} 0 & \leq & V & \leq & 1 \\ 0 & \leq & \Delta v & \leq & 1 \\ 0 & \leq & i & \leq & 1 \end{array}$$

where V is a measure of the average rotation speed of the two wheels, Δv is the algebraic difference between the signed speed values of the wheels (positive is one direction, negative the other) transformed into positive values, and i is the activation value of the proximity sensor with the highest activity. The function Φ has three components: the first one is maximized by speed, the second by straight direction, and the third by obstacle avoidance. Since the robot has a circular shape and the wheels can rotate in both directions, this function has a symmetric surface with two equal maxima, each corresponding to one motion direction (Figure 4).

The evolutionary training was a standard genetic algorithm as described by Goldberg [7] with fitness scaling and roulette wheel selection, biased mutations [15], and one-point crossover (experiment parameters are given in the Appendix). The neural network architecture was fixed and consisted of a single layer of synaptic weights from eight input units (clamped to the sensors) to two output units (directly connected to the motors)

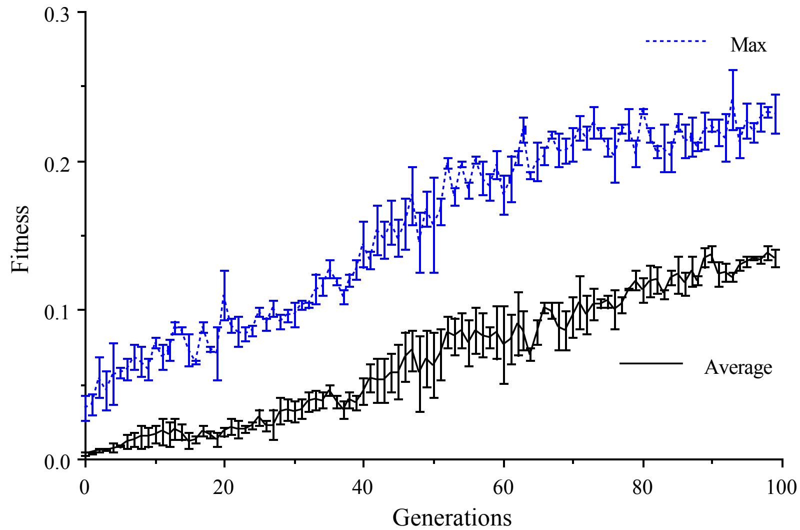


Figure 5: Population average fitness and best individual fitness at each generation. Values are averaged over three runs (S.E. displayed).

with mobile thresholds, logistic activation functions, and discrete-time recurrent connections only within the output layer. Given the small network size, each synaptic connection and each threshold was coded as a floating point number on the chromosome [21]. Each individual of a population was in turn decoded into the corresponding neural networks, the input nodes connected to the robot sensors, the output nodes to the motors, and the robot was left free to move for a given number of steps (motor actions) while its performance Φ was automatically recorded. Each motor action lasted 300 ms. Between one individual and the next, a pair of random velocities was applied to the wheels for 5 seconds. This procedure was aimed at limiting the artifactual inheritance of particular locations between adjacent individuals in the populations.

4.2 Results

Khepera genetically learns to navigate and avoid obstacles in less than 100 generations (Figure 5), each generation taking approximatively 39 minutes. However, around the 50th generation the best individuals already exhibit a near to optimal behavior. Their navigation is extremely smooth, they never bump into walls and corners, and try to keep a straight trajectory. This allows them to perform complete laps of the corridor without turning back. These results are highly reliable and have been replicated in many runs of the experiment.

It is interesting to analyze a single run of the evolutionary development of Khepera by looking at the values of the three fitness components for the best individuals in the population at each generation (Figure 6).

During the initial generations the best individuals are those that move straight at very low velocities (about 2 mm/s). High oscillations of the sensory component indicates that they cannot yet discriminate between walls and empty spaces: it is still much up to individual "luck" (starting location) to avoid crashing into an obstacle. Most of the remaining individuals in the initial generations spend their life by rotating in place. Maximizing the fitness function Φ means to find a balance among the three components because none of them can assume the maximum value without lowering one of the other two. A stable balance is found around the 50th generation. In the remaining 50 generations the robot increases only the global motion speed. However, the global speed never reaches the maximum value (80 mm/s), not even when the evolutionary process is continued until the 200th generation. For all the best individuals, the robot speed peaks at 48 mm/s when positioned in zones free of obstacles. This self-adjustment of the maximum cruising speed has an adaptive meaning. Since sensors and motors are updated only every 300 ms and many passages in the environment are rather narrow, if Khepera had moved faster it would have often crashed into walls without the possibility to detect them. Thus, the system has adapted its own behavior to the physical characteristics of its own sensory system and of the environment where it lives. We have tested some of the best individuals of the last generations in new environments with a variety of objects (differing in shape, color, texture, and light absorbency) and new light conditions (full sunlight, new rooms with different artificial light). We have

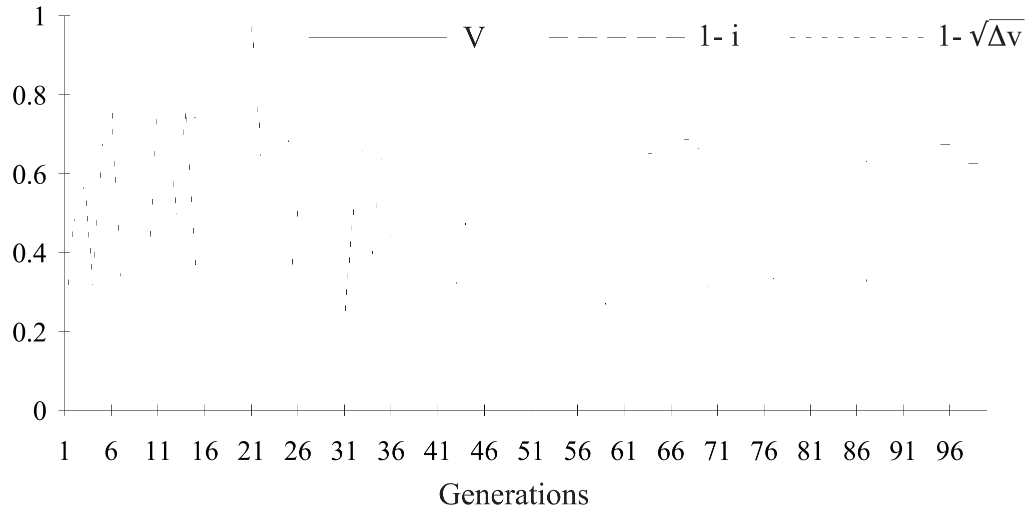


Figure 6: Fitness components values for the best individual of each generation. V is maximized by robot velocity; $1 - i$ is maximized by obstacle avoidance; $1 - \sqrt{\Delta v}$ is maximized by straight trajectory.

also tested the best individuals with other robot bodies (same model, but obviously with slight variations of the sensor responses). In all these cases Khepera navigates successfully without touching any of the objects and trying to keep a straight trajectory. All the individuals tested show a preferential turning direction which solely depends on the initial conditions of the evolutionary run (initial weight values, interaction with the environment), but they can turn in both directions when required by the environment.

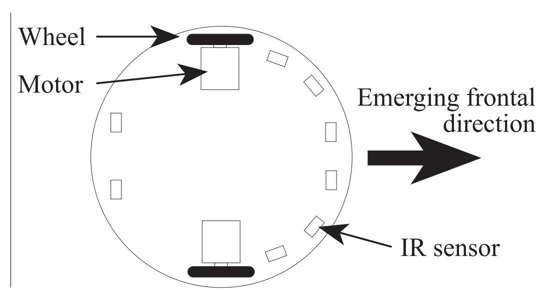


Figure 7: IR sensors and motors layout in Khepera. Diameter size is 55 mm, maximum speed in either direction is 80 mm/s.

4.3 Discussion

A basic characteristic of autonomous systems is the ability to self-regulate their own behavior in order to maximize the probability of survival and reproduction. In this sense adaptation is function of the interaction between two variables, the physical properties of the environment and the characteristics of the organism's body. The success of any plan, strategy, or single action, depends not only on the affordances of the environment, but also on the capacity to detect them and adequately respond. In nature we can observe a continuous evolutionary co-adaptation of body structures and behavioral repertoire. Although we cannot yet expect changes in the hardware structure of an autonomous robot, still we should observe self-selection of the behavioral strategies that exploit at best the physical features of the robot's body and sensory-motor apparatus. We have already seen an example of such a behavioral adaptation in the case of the speed self-regulation of our robot. Another significative example of autonomous adaptation is given by the direction of motion.

Khepera has a perfectly circular and symmetric body shape and the wheels can rotate at equal speeds in both directions. In terms of pure kinematics, thus, it is logical to expect that the robot will equally move in either direction, depending on initial internal and external conditions. However, in all our experiments, early during the evolution the robots develop a frontal direction of motion that corresponds to the side with more sensors (Figure 7). The development of this frontal direction of motion allows the robot to face obstacles with the side that provides a finer resolution and a larger visual angle. Those individuals that move "backward" are very likely to get stuck in convex and sharp corners or fail to detect a lateral collision with a wall; hence, they disappear very soon from the population. (Analogous phenomena of behavioral adaptation to the visual configuration of a simple simulated organism have been shown by [5].) However, rear sensors do not go out of use. The neural networks of the best individuals of the final generations still make use of that information to change trajectory if

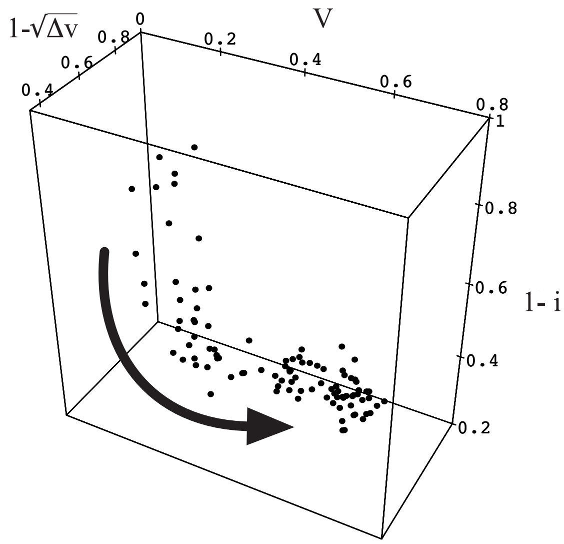


Figure 8: State-space representation of the adaptation process. Each dot is the state of the best individual of a generation. The arrow shows the direction of motion during evolution. The dots concentrate in a sub-space indicated by the arrow tip in the last 20 generations. Axes range spans from 0 to 1 (only covered space is shown in the picture).

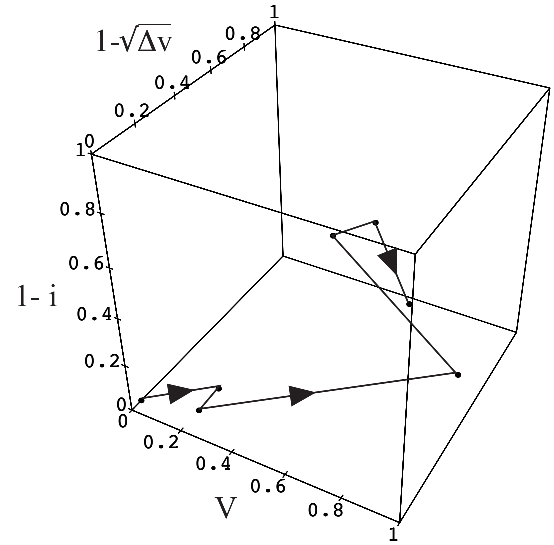


Figure 9: State-space representation of the behavior of the best individual of the last generation, when pulled apart from its equilibrium state.

something is approaching the robot from the back. As for any dynamic system, also in the case of evolved robots it is important to understand and try to describe the state-transition phase. But an autonomous system is not completely controllable and observable [12]. This holds also for our robot, both because the dynamics and results of the evolutionary process cannot be controlled, and because the inner functioning of the neural network, as we will see later, is not linear and each state depends upon a previous history of states. However, as in the case of animals, the activity of an autonomous agent depends on the state of the agent itself, such as its level of energy, the perception of the environment, and the memory of previous states. This analysis yields to the construction of an n-dimensional state space, where the axes are provided by n state variables considered. This "state-space approach" has been used in ethology [13] to describe animal behavior in quantitative terms, and can be applied also to our agent. We can describe our agent as a point in a three dimensional space given by the values of the three fitness components and monitor its change in time.

Figure 8 is a state-space plot of the best individuals of each generation during evolution. The adaptation process is described by a reduction of oscillations and by a gradual displacement toward a sub-region of this space. This region of the adaptation space is compact and bounded, and represents the stability conditions of the system [12] that satisfy the survival criterion.

Our system is asymptotically stable because, when disturbed (by the recombination and mutation operators during the last 20 generations), it tends to stay within the same adaptation zone. This holds also when we analyze the behavior of a single individual. If we disturb the system by pulling it away from its equilibrium state, it will tend to return and stay in its original state (Figure 9). This analysis may be carried further on along the lines of the "Adaptation Theorem" of Sibly and McFarland [18], but there is not space enough here (we will consider this issue in further detail with more complex examples of evolved behaviors in another paper). A final consideration is deserved by the comparison between our agent and a distributed Braitenberg's vehicle designed to go straight and avoid obstacles. Braitenberg's vehicle (which has been implemented on Khepera too) is a linear reactive system that, basically, when some sensors are activated, gives more energy to the ipsilateral motor and inhibits the controlateral one. The pattern of synaptic connections is symmetrical about the front axis. This very simple system is very efficient, but is gets stuck as soon as two symmetric and controlateral sensors become equally activated. In this case the total amount of energy given to each motor is equal and tends to 0. Instead, our agent has developed a pattern of synaptic connections similar to Braitenberg's vehicle, but it has also accurately exploited the recurrent connections at the output layer and the non-linearities embedded in the activation functions. The best individuals of the last generations never get stuck in such cases because the state of the motors is not uniquely defined by the current state of the sensors, but also by the previous history of actions.

5 Conclusion

We have described and analyzed a working example of an artificial autonomous agent. Our robot satisfies most of the basic criteria that underlie the definition of autonomous agents. Through the evolutionary process Khepera has automatically and autonomously developed the optimal distributed control system to survive in the environment where it has been placed. The role of the human experimenter has been indeed rather small, specifically limited to formulate only the survival criterion and the global structure of the net. We have neither pre-designed the behaviors of the robot, nor have intervened during evolution. The robot itself and alone has developed -starting from a sort of *tabula rasa* - a set of strategies and behaviors as a result of the adaptation to the environment and its own body. Despite its simple components and the simple survival criterion, it is difficult to control and predict the robot behavior, due to the non-linearities and feedback connections exploited for optimal navigation and obstacle avoidance. We have tried to describe our agent's behavior with quantitative ethological tools, and we have also showed two emergent phenomena such as speed self-regulation and frontal direction. Our current work is aimed at using the same approach in more complex environments where the fitness criterion is not anymore fixed by the experimenter, but is the natural and logical result of the interaction between the physical characteristics of the robot and the type of environment. We have already obtained new significant results where homing for battery recharge is purely an emergent behavior. These data make us confident in thinking that our approach is a valid methodology for automatically creating complex autonomous agents. Future work will enable evolvability and more flexibility (through a major adherence to biological plausibility) in the neural network structure and will employ learning during life as well.

Appendix

Genetic algorithm parameters:

Population size	80
Generation number	100
Crossover probability	0.1
Mutation probability	0.2
Mutation range	± 0.5
Initial weight range	± 0.5
Final weight range	Not bounded
Life length	80 actions
Action duration	300 ms

Acknowledgments

We would like to thank Edo Franzi and André Guignard for the important work in the design of Khepera. Dario Floreano has been partially supported by the Italian National Research Council, grant n. 92.01065.PF67 (co-ordinator W. Gerbino). Francesco Mondada has been partially supported by the Swiss National Research Foundation (project PNR23).

References

[1] D. H. Ackley and M. L. Littman. Interactions between learning and evolution. In C.G. Langton, J.D. Farmer, S. Rasmussen, and C. Taylor, editors, *Artificial Life II: Proceedings Volume of Santa Fe Conference*, volume XI. Addison Wesley: series of the Santa Fe Institute Studies in the Sciences of Complexities, 1991.

[2] R. D. Beer and J. C. Gallagher. Evolving dynamical neural networks for adaptive behavior. *Adaptive Behavior*, 1:91–122, 1992.

[3] V. Braitenberg. *Vehicles. Experiments in Synthetic Psychology.* MIT Press, Cambridge, MA, 1984.

[4] R. A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47:139–59, 1991.

[5] D. T. Cliff and S. G. Bullock. Adding "foveal vision" to Wilson's animat. Technical report CSRP 263, University of Sussex, School of Cognitive and Computing Sciences, December 1992.

[6] D. Floreano. Robogen: A software package for evolutionary control systems. Release 1.1. Technical report LabTeCo No. 93-01, Cognitive Technology Laboratory, AREA Science Park, Trieste, Italy, 1993.

[7] D. E. Goldberg. *Genetic algorithms in search, optimization and machine learning.* Addison-Wesley, Reading, MA, 1989.

[8] I. Harvey, P. Husbands, and D. T. Cliff. Issues in evolutionary robotics. In J. Meyer, H. L. Roitblat, and S. W. Wilson, editors, *From Animals to Animats II: Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, Cambridge, MA, 1992. MIT Press-Bradford Books.

[9] J. H. Holland. *Adaptation in natural and artificial systems.* The University of Michigan Press, Ann Arbor, 1975.

[10] C. G. Langton. Preface. In C.G. Langton, J.D. Farmer, S. Rasmussen, and C. Taylor, editors, *Artificial Life II: Proceedings Volume of Santa Fe Conference*, volume XI. Addison Wesley: series of the Santa Fe Institute Studies in the Sciences of Complexities, 1991.

[11] P. Maes. Behavior-based artificial intelligence. In J. Meyer, H. L. Roitblat, and S. W. Wilson, editors, *From Animals to Animats II: Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, Cambridge, MA, 1992. MIT Press-Bradford Books.

[12] D. J. McFarland. Autonomy and self-sufficiency in robots. AI-Memo 92-03, Artificial Intelligence Laboratory, Vrije Universiteit Brussel, Belgium, 1992.

[13] D. J. McFarland and A. Houston. *Quantitative Ethology: the state-space approach.* Pitman Books, London, 1981.

[14] F. Mondada, E. Franzi, and P. Ienne. Mobile robot miniaturization: A tool for investigation in control algorithms. In *Proceedings of the Third International Symposium on Experimental Robotics*, Kyoto, Japan, 1993.

[15] D. Montana and L. Davis. Training feed forward neural networks using genetic algorithms. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, San Mateo, CA, 1989. Morgan Kaufmann.

[16] S. Nolfi. Genotypes for neural networks. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks.* MIT Press, Cambridge, MA, 1993.

[17] S. Nolfi, D. Parisi, and R. Pedone. How noise helps generalization in feed-forward networks. In E. R. Caianiello, editor, *Parallel Architectures and Neural Networks: Proceedings of the 6th Italian Workshop on Neural Networks*, Singapore, 1992. World Scientific P.

[18] R. M. Sibly and D. J. McFarland. A state-space approach to motivation. In D. J. McFarland, editor, *Motivational Control Systems Analysis.* Academic Press, London, 1974.

[19] L. Steels. Building agents out of autonomous behavior systems. In L. Steels and R. Brooks, editors, *The "artificial life" route to "artificial intelligence". Building situated embodied agents.* Lawrence Erlbaum, New Haven, 1993.

[20] P. F. M. J. Verschure, B. J. A. Kröse, and R. Pfeifer. Distributed adaptive control: The self-organization of structured behavior. *Robotics and Autonomous Agents*, 9:181–96, 1992.

[21] X. Yao. A review of evolutionary artificial neural networks. *International Journal of Intelligent Systems*, 4:203–222, 1993.

VII Fitness Function Formulation and Classification

Based on 2009 Nelson, Barlow, Doitsidis article in the journal "Robotics & Autonomous Systems".

- **Optimal Fitness Function** - provides the most novel control, while requiring the provision of least amount of domain-specific knowledge.

The fitness functions are classified based on the level of a priori knowledge they incorporate.

① Training Data Fitness Functions

These fitness functions use pre-existing datasets of input-output mappings to evaluate how well a controller replicates the expected outputs.

Mechanism:

- The fitness function minimizes the error between the system's output & the expected output for each input in the dataset
- Often used in supervised learning approaches, such as training neural networks via backpropagation.

Limitations:

Requires full a priori knowledge of the task in the form of a complete training dataset, which may not always be feasible.

② Behavioural Fitness Functions

The functions measure the robot's local behaviours, such as its responses to sensor inputs or how it moves in the environment, without directly evaluating task completion.

Mechanism:

- They evaluate specific behaviours believed to contribute to task success (e.g., obstacle avoidance or straight-line movement)

- Easy to design for well-defined behaviours

Limitation:

- Biases evolution toward behaviours the designer assumes are necessary for task completion.

③ Functional Incremental Fitness Functions

These functions evolve behaviours incrementally, starting with simpler tasks and progressing to more complex ones by modifying the fitness function over time.

Mechanism:

- Evolution begins with a fitness function rewarding basic capabilities
- After initial goals are met, the function adds more criteria to encourage complexity.

Example:

- In a navigation task, the function might first reward obstacle avoidance, then path-following, and finally goal-reaching.

Strength:

- Prevents stagnation in early evolution by solving the "sub-minimally competent population" problem (i.e, no initial solutions with measurable fitness)

Limitation:

- Reduces the likelihood of discovering unexpected solutions, as the evolutionary path is tightly constrained by the designer.

④ Tailored Fitness Functions

These are hybrid functions combining task-specific behavioural terms with aggregate measures of partial or overall task success.

Mechanism:

- Include behavioural terms (e.g, specific movements or sensor responses)
- Add aggregate terms measuring partial success (e.g, closeness to a goal or intermediate milestones)

Strength:

- Captures intermediate behaviours while incentivizing ultimate task success.

Limitation:

- Requires significant designer input and trial-and-error tuning.

⑤ Environmental Incremental Fitness Functions

The functions evolve behaviours by progressively increasing the complexity of the robot's environment instead of altering the fitness function.

Mechanism:

- Start with a simplified environment (e.g., no obstacle)
- Gradually introduce complexities (e.g., adding obstacles or adversaries) as the robot evolves.

Strength

- Encourages diverse strategies without heavily constraining the evolutionary process.

Limitation

- Designing the progression of environmental complexity can be challenging.

⑥ Competitive & Co-competitive Fitness Functions

Fitness is evaluated by direct competition, either within a single population (competitive) or between two distinct populations (co-competitive).

Mechanism:

- <u>Competitive</u>: Robots compete for a shared resource or task (eg, reaching a goal faster)
- <u>co-competitive</u>: Two populations evolve simultaneously to outperform each other (predator-prey dynamics).

Strength:

- Promotes adaptive behaviours through dynamic challenges.

Limitation:

- can lead to cyclical or overly specialized behaviours.

⑦ Aggregate Fitness Functions

These functions measure only high-level task success or failure without considering the methods used to achieve it.

Mechanism:

- Provide a single scalar evaluation of task success.
- often binary (success/failure) or quantitative (e.g., no. of objects collected)

Strength: Minimizes designer bias, enabling evolution of novel strategies

Limitation: Faces the bootstrap problem, where early populations fail to show any detectable success, leading to evolutionary stagnation.

- Comparison Table

	Fitness Function	A priori Knowledge	Flexibility	Bias	Applications
1	Training Data	Very High	Low	High	Supervised-learning
2	Behavioural	High	Moderate	Moderate -high	Basic- control behaviours
3	Functional Incremental	Moderate-high	Moderate	Moderate	complex task w/ sub-skills
4	Tailored	Moderate	High	Moderate	Multi-faceted tasks
5.	Environmental Incremental	Moderate	High	Low	Robust strategy evolution
6	competitive/ co-competitive	Low-moderate	High	Low-moderate	Adaptive behaviours, adversarial tasks
7	Aggregate	very low	High	Very Low	Novel solution for complex tasks

Robotics and Autonomous Systems ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Fitness functions in evolutionary robotics: A survey and analysis

Andrew L. Nelson [a,*], Gregory J. Barlow [b], Lefteris Doitsidis [c]

[a] Androtics, LLC, PO Box 44065, Tucson, AZ 85733-4065, USA
[b] The Robotics Institute, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA
[c] Intelligent Systems & Robotics Laboratory, Department of Production Engineering & Management, Technical University of Crete, 73132, Hania, Greece

ARTICLE INFO

Article history:
Received 13 March 2007
Received in revised form
16 April 2008
Accepted 29 September 2008
Available online xxxx

Keywords:
Evolutionary robotics
Fitness functions
Genetic algorithms
Autonomous learning robots
Artificial life

ABSTRACT

This paper surveys fitness functions used in the field of evolutionary robotics (ER). Evolutionary robotics is a field of research that applies artificial evolution to generate control systems for autonomous robots. During evolution, robots attempt to perform a given task in a given environment. The controllers in the better performing robots are selected, altered and propagated to perform the task again in an iterative process that mimics some aspects of natural evolution. A key component of this process – one might argue, *the* key component – is the measurement of fitness in the evolving controllers. ER is one of a host of machine learning methods that rely on interaction with, and feedback from, a complex dynamic environment to drive synthesis of controllers for autonomous agents. These methods have the potential to lead to the development of robots that can adapt to uncharacterized environments and which may be able to perform tasks that human designers do not completely understand. In order to achieve this, issues regarding fitness evaluation must be addressed. In this paper we survey current ER research and focus on work that involved real robots. The surveyed research is organized according to the degree of *a priori* knowledge used to formulate the various fitness functions employed during evolution. The underlying motivation for this is to identify methods that allow the development of the greatest degree of novel control, while requiring the minimum amount of *a priori* task knowledge from the designer.

1. Introduction

The primary goal of evolutionary robotics (ER) is to develop methods for automatically synthesizing intelligent autonomous robot systems. Although the greater part of current research is applied to control systems alone, ER also applies this ideal of automatic design to the creation of robot bodies (morphology) and also to the simultaneous evolution of robot control and morphology. This is often stated in terms of co-evolution of body and mind.

Automatic robot controller development methods that do not require hand coding or in-depth human knowledge are potentially of great value because it may be possible to apply them to domains in which humans have insufficient knowledge to develop adequate controllers directly. Advanced autonomous robots may someday be required to negotiate environments and situations that their designers had not anticipated. The future designers of these robots may not have adequate expertise to provide appropriate control algorithms in the case that an unforeseen situation is encountered in a remote environment in which a robot cannot be accessed. It is not always practical or even possible to define every aspect of an autonomous robot's environment, or to give a tractable dynamical systems-level description of the task the robot is to perform. The robot must have the ability to learn control without human supervision.

In contrast to intelligent autonomous mobile robots, most industrial robots perform precisely defined tasks, using methods that are well defined at a low level. For example, an industrial robot (even a very complex one) is usually described by a dynamical model, and the task it is intended to perform can be achieved by a well-defined method or procedure. Often, the task itself can also be described by a dynamical model. Arriving at a mathematical description of an optimal or near-optimal control strategy to perform the task becomes a matter of mathematical and sometimes heuristic optimization of well-defined procedures [91].

The situation is quite different for autonomous robots that must interact dynamically with complex environments. While the overall task may remain well defined at a high level, an effective solution algorithm is usually not well defined. Most non-trivial tasks for intelligent autonomous robots cannot be described adequately by tractable dynamical models. Essentially, autonomous robot control designers know what task they want a given robot to perform, but they do not know how the robot will perform the task.

* Corresponding author. Tel.: +1 520 822 6921.
E-mail addresses: alnelson@ieee.org (A.L. Nelson), gjb@cmu.edu (G.J. Barlow), ldoitsidis@dpem.tuc.gr (L. Doitsidis).

0921-8890/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.robot.2008.09.009

Control systems for autonomous robots are often programmed directly by researchers or designers. Such control programs can be very complex. Researchers must anticipate which abilities a given robot will need, and then formulate these into a control program or control hierarchy. Many researchers in the field of autonomous robot control rely on sophisticated control architectures to facilitate overall control design [88,89].

As the complexity of an environment and task for a given autonomous robot increases, the difficulty of designing an adequate control system by hand becomes a limiting factor in the degree of functional complexity that can be achieved. A potential solution to this problem is to develop methods that allow robots to learn how to perform complex tasks automatically. Developing machine learning methods for use in robotic systems has in fact become a major focus of contemporary autonomous robotics research. Some of these methods, including evolutionary robotics, focus on the ground-up learning of complete control systems. The goal of these methods is to learn the entirety of the control structure, rather than simply learning particular components, such as object classification or instances of path planning.

Learning intelligent control for autonomous agents is in some ways very different from other forms of machine learning or optimization (see [90] for an introduction to machine learning). In particular, it is often not possible to generate a succinct training data set that might be used to train controllers using batch methods or error back propagation. Defining discrete states for complex autonomous robot-environment systems is also problematic and traditional temporal difference (TD) methods such as Q-learning are not easily applied to intelligent autonomous control learning problems in dynamic continuous environments. Evolutionary robotics approaches the problem of intelligent control learning by applying population-based artificial evolution to evolve robot control systems directly. This evolutionary process represents a form of machine learning that does not necessarily require complete knowledge of environment, robot morphology, or task dynamics.

The field of evolutionary robotics is situated within a broader area of research focused on automatic methods of environment-based learning and autonomous systems development. This broader area of inquiry includes developmental robotics [108,109], artificial life, and a variety of other non-evolutionary computation-based machine learning specialties applied to fully autonomous systems. Although this survey focuses specifically on objective functions used in evolutionary robotics research, objective functions are a central component of many control learning methods applied to intelligent autonomous agents.

The field of automatic intelligent control learning for autonomous robots is in its infancy. Much of the research surveyed in this paper focused on learning how to perform relatively simple tasks. Phototaxis, for instance, is a well-studied task in ER and is representative of the complexity of tasks studied in much of the current and past research. To perform this task, a robot in an environment must identify and home in on a light source.

The current focus of ER is on developing methods for evolving controllers capable of performing more difficult and complex tasks, rather than optimizing the evolution process for tasks that have already been achieved. Hence, producing a system that could generate efficient controllers for the task of phototaxis using 10% or even 50% less computing time would not be considered a real advancement in the field. On the other hand, one particular ER effort might be considered an improvement over an earlier work if the later work required the use of much less *a priori* knowledge on the part of the researchers to evolve controllers for a similar task. In this case, the later system would have learned a greater portion of novel intelligent control, and would represent an improvement in methodology [106].

In general, the research papers reviewed in this survey report the successful evolution of controllers capable of performing the intended tasks. Moreover, most attempted research that failed to produce functional controllers will likely not have been published. Hence, the success of research is measured in the difficulty of tasks investigated, and the amount of *a priori* information needed to generate successful evolution of controllers capable of performing those tasks.

1.1. Prior work

The field of ER has been reviewed in various publications [1–8]. However, there is no current comprehensive review of the field that investigates the central issue of fitness selection methods in evolutionary robotics.

[1,3] both provide excellent reviews of the state of the field of ER in the mid-1990's. Robot controller hardware evolution is reviewed in [7] and an extensive review of the use of multi-objective optimization in evolutionary robotics is found in [8].

[5] explores issues related to training phase learning, lifetime learning and embodied learning in real robots, but that work differs considerably from our work both in focus and coverage. We focus on the issues of fitness determination and objective function formulation, and compare reported fitness functions using a common function nomenclature and classification system.

An important unanswered question within the field of ER is whether the methods used so far to obtain the moderately complex proof-of-concept results reported over the last decade and a half can be generalized to produce more sophisticated autonomous robot control systems.

1.2. Overview of robot controller evolution

In this paper, the term *controller* is used to describe the computational portion of an autonomous mobile robot system that receives information from the robot's sensors, processes this information, and produces actuator or motor commands that cause the robot to move or interact with its environment. The controller in this sense might be thought of as the brain of the robot, and some ER researchers use this terminology. In the broader field of autonomous robotics, control learning may focus on selected portions of a robot's control abilities, such as object recognition [94], path planning and localization [92,93], or error and fault accommodation. In contrast, ER research is typically directed toward learning (or evolving) the entire control system.

In ER, the process of controller evolution consists of repeating cycles of controller fitness evaluation and selection that are roughly analogous to generations in natural evolution. During each cycle, or generation, individual controllers taken from a large population of controllers attempt to perform a task or engage in some form of an evaluation period. This involves instantiating each controller into a robot (either real or simulated) and allowing the robot to interact with its environment (which may include other robots) for a period of time. In later discussions we will refer to this as an *evaluation, trial* or *test period*. Following this period, each robot controller's performance is evaluated based on a fitness function (also called an objective function). In the final step of every cycle, a genetic algorithm (GA) is applied [95]. The GA uses information generated by the fitness function to select and propagate the fittest individuals in the current population of controllers to the next generation population. During propagation, controllers are altered using stochastic genetic operators such as mutation and crossover to produce offspring that make up the next generation of controllers. Cycles are repeated for many generations to train populations of robot controllers to perform a given task,

Please cite this article in press as: A.L. Nelson, et al., Fitness functions in evolutionary robotics: A survey and analysis, Robotics and Autonomous Systems (2008), doi:10.1016/j.robot.2008.09.009

and evolution is terminated when suitably functional controllers arise in the population.

The success of the entire process depends on how effective the fitness function is at selecting the best controllers, and it is this feature of evolutionary robotics on which we focus our attention. In this paper we survey the current ER literature with an eye towards fitness functions and the relationship between fitness evaluation methods and complexity of behavior evolved. We present a taxonomic classification of fitness functions used in evolutionary robotics research (Section 2) and use this to organize the surveyed work (Section 3).

Many variations on standard GAs are used in ER, but the majority of the research uses the traditional set of process steps consisting of test, evaluate fitness, select, mutate/recombine and propagate during each generation. Other related population-based algorithms include particle swarms, ant optimization [101] and artificial immune optimization methods [102]. Such methods incorporate dynamics observed in nature into search algorithms based on the assumption that search-algorithm-like processes observed in nature represent efficient methods honed by evolution during the course of evolution of life on Earth. These methods, as well as single agent learning methods, are also fitness function driven.

The case can be made that most forms of learning of intelligent behavior based on interaction between agent and environment share similar underlying characteristics. The main motivation for using artificial evolution and GAs in learning robots is to accommodate the computationally intractable uncharacterized high-dimension real-valued search spaces encountered in intelligent control learning problems.

1.3. The fitness function

Successful evolution of intelligent autonomous robot controllers is ultimately dependent on the formulation of suitable fitness functions that are capable of selecting for successful behaviors without specifying the low-level implementation details of those behaviors.

The fitness function is at the heart of an evolutionary computing application. It is responsible for determining which solutions (controllers in the case of ER) within a population are better at solving the particular problem at hand. In work attempting to evolve autonomous robot controllers capable of performing complex tasks, the fitness function is often the limiting factor in achievable controller quality. This limit is usually manifested by a plateau in fitness evaluation in later generations, and indicates that the fitness function is no longer able to detect fitness differences between individuals in the evolving population.

Although developing an experimental research platform capable of supporting the evolutionary training of autonomous robots remains a non-trivial task, many of the initial concerns and criticisms regarding embodiment and transference from simulated to real robots have been addressed. There are sufficient examples of evolutionary robotics research platforms that have successfully demonstrated the production of working controllers in real robots [9–12]. Also, there have been numerous examples of successful evolution of controllers in simulation with transfer to real robots [13–19]. One of the major achievements of the field of ER as a whole is that it has demonstrated that sophisticated evolvable robot control structures (such as neural networks) can be trained to produce functional behaviors in real (embodied) autonomous robots. What has not been shown is that ER methods can be extended to generate robot controllers capable of complex autonomous behaviors. In particular, no ER work has yet shown that it is possible to evolve complex controllers in the general case or for generalized tasks.

Concerns related to fitness evaluation remain largely unresolved. Much of the ER research presented in the literature employs some form of hand-formulated, task-specific fitness function that more or less defines how to achieve the intended task or behavior. The most complex evolved behaviors to date consist of three or four coordinated fundamental sub-behaviors [14,20–22]. In [14], the fitness evaluation method used was relatively selective for an *a priori*, known or predefined solution. In [20–22] the fitness functions used for selection contained relatively little *a priori* knowledge, and allowed evolution to proceed in a relatively unbiased manner. This is an interesting contrast to much of the work aimed at evolving simple homing or object avoidance behaviors, many of which use complex fitness functions that heavily bias the evolved controllers toward an *a priori* known solution.

1.4. Robots

A wide variety of robots equipped with different kinds of sensor types are used in ER. Almost all of these are mobile robots of one form or another and include wheeled mobile robots, legged robots, and flying robots.

The most typical robots used in this field are small (between 5 and 20 cm in diameter) differential drive (skid steering) robots equipped with several IR proximity sensors, photodetectors, and tactical sensors. Some of these robots also use video. For most of the work discussed in this survey, robots operate in small arenas that contain obstacles and sometimes other robots. These arenas might be small enough to be placed on a desktop, or they might be constructed on a portion of floor space in a research lab or office.

There are several robot platforms that are commercially available. The Khepera robot platform is one of the most commonly used small differential drive robot systems in ER [97]. It is of modular design and can be equipped with IR, tactile and photosensors. A CCD camera unit and gripper unit are also available. The Khepera is 5 cm in diameter, has limited computational power and is often operated via an umbilical by a remote computer. The Koala is a larger differential drive robot (30 cm in length) also manufactured by the makers of the Khepera, and has been used in a few ER experiments. Commercially available LEGO Mindstorm-based robots have also been used for several ER experiments.

Many researchers use small custom robots of their own construction for ER work. For example, the EvBot [96] is a small differential drive robot that has been used by several research groups for a variety of ER experiments.

Larger lab robots such as the RWI B21 [50] and the Nomad [28] have been used in a few ER research efforts. Unlike the smaller robots, these robots are heavy, more powerful, and capable of damaging walls and other laboratory equipment. In addition, these robots can be quite expensive and difficult to maintain.

A smaller but considerable amount of work has been done using legged robots from bipeds to octopods. The majority of these robots are custom-built by the various researchers and labs using hobby servos. In addition to these, the commercially available Sony AIBO quadruped robot has been used in a number of gait learning and locomotion learning ER experiments. This is a small 18 degree-of-freedom (DOF) robot that uses video and IR sensors.

When discussing particular research examples in the survey portion of this paper we mention briefly the type of robot used and the sensor configuration, but do not go into detail unless the robot platform is significantly different from the common differential drive systems used by the majority of the researchers.

Please cite this article in press as: A.L. Nelson, et al., Fitness functions in evolutionary robotics: A survey and analysis, Robotics and Autonomous Systems (2008), doi:10.1016/j.robot.2008.09.009

1.5. Controller architectures

Learning control is common to all ER work. A variety of controller architectures are used in ER. These include neural networks, evolvable programs, various parameterized control structures, and evolvable hardware devices.

Neural networks are well suited for training with evolutionary computing-based methods because they can be represented by a concise set of tunable parameters. A wide variety of neural network structures have been used. The most common of these are layered feedforward or recurrent network architectures. A few of the papers cited here use Hebbian networks or other self-training networks, and these are pointed out. Neural networks are used in approximately 40% of ER work.

Evolvable programming structures are used in about 30% of the ER research. The process is referred to as genetic programming (GP). The work using evolvable hardware generally implemented some form of genetic programming or evolvable logic structure in hardware.

Much of the ER work that focused on evolution of gaits for legged robots simply evolved sets of gait control parameters. For instance, the Sony AIBO robot's gait is controlled by a set of timing and joint-position parameters, and in several of the works surveyed here, a subset of these were evolved directly. Evolving parameters of an otherwise specified gait control program differs from the majority of other ER work in that the full control system was not evolved. Most other ER work focuses on learning of monolithic control systems that act directly on sensor inputs and produce actuator commands.

1.6. Tasks and behaviors

In this subsection we will briefly discuss some of the most common tasks that robots have been evolved to perform in ER research. Some of these tasks have been studied by many different researchers over the last two decades.

Locomotion and object avoidance is one of the most frequently investigated robot tasks studied in ER. In this task robots must evolve to travel about an environment while avoiding stationary and sometimes mobile obstacles. This task might also be referred to as navigation, although technically, the term navigation generally involves traveling to and from specified locations, not just moving about without hitting anything.

Gait learning in legged robots is another commonly studied task. In the simplest form of gait evolution, functional forward locomotion is the only goal and no sensor inputs are used. Gait learning is a form of locomotion learning, but it might be considered a somewhat more difficult problem in legged robots than in wheeled robots. For locomotion to occur in wheeled robots, the wheel actuators must simply be turned on. For differential drive robots, this essentially consists of a 2-DOF system and evolving a controller to produce straight motion in an open environment would be considered trivial by modern standards. Locomotion in legged robots, on the other hand, is much less trivial. Most legged robots have between 12 and 20 DOF. Simply energizing the actuators is very unlikely to produce efficient locomotion. The leg actuators must be cycled on and off in a coordinated and controlled fashion.

Phototaxis is another frequently studied task. As mentioned earlier in the paper, robots must detect and home in on a light source in this task. Goal homing is a related task, but here, the goal location is not marked by a light source, and might not be marked at all. Environment complexity can play a significant role in the difficulty of the behavior to be learned in both goal homing and phototaxis tasks. Environments that contain objects that occlude the goal or light location from the sensors of the robots will require a more sophisticated strategy to negotiate than would be required in a simple environment.

Searching tasks are also commonly studied in ER. In searching tasks, robots travel about an environment searching for various objects. This might be considered a variation on goal homing, but the environment could contain many search objects.

Foraging is similar to searching, but the robots are also required to pick up or acquire the objects, and in some cases to then deposit the objects at a goal location. Foraging with object deposition (or object carrying) is on the complex end of the scale for tasks and behaviors studied in ER. Robots must find objects in an environment, then pick them up and carry them to another location and deposit them. These steps taken together contain an element of sequencing and cannot easily be performed by a purely reflexive system.

Predator and prey tasks involve one robot learning (or evolving) to capture another robot while the other learns to evade the first. There are several variations on this theme. Most common is a setup in which only one of the robots uses evolving controllers while the other uses a fixed hand-designed controller.

There are a few examples of other complicated tasks found in the literature. These include multiple goal homing, in which a robot must travel to two or more goal locations in a specified sequence. Another more complex task is represented by groups of robots competing against one another to find hidden objects.

1.7. Fitness landscapes

The analysis of fitness landscapes is usually considered to be an important issue in evolutionary computing applications. For a given search space, a given fitness function will define a fitness landscape or manifold. In evolutionary robotics, the search space is defined by the genome defining the controller representation (or controller and morphology representation, if body and mind are being co-evolved). Each evolvable parameter of the genome defines a dimension of the search space, and the fitness landscape is then given by the manifold defined by the fitness of each point in the search space.

In many areas of evolution computing, great effort is made to elucidate the properties of the search space and the topology of a given fitness landscape generated by application of a given fitness function. Certain more tractable fitness landscapes are amenable to specialized algorithms that may reduce computation effort, guarantee convergence or otherwise produce desirable features.

However, in ER, genome search spaces and fitness landscapes are often very difficult to characterize to the degree that significant benefit can be gained. The topologies of search spaces traversed by the evolving dynamic controller populations are generally rugged in the extreme, may have varying numbers of dimensions, and may potentially be non-static [98]. Because of these factors, search spaces and associated fitness landscapes in ER are often intractable in terms of full characterization. This state reflects the fact that the genomes are designed to be able to represent autonomous dynamic agents, at least in terms of control.

Currently, there is no adequate theory that can relate salient features of intelligent systems to representations. For example, it is difficult or impossible to distinguish between a well trained and a poorly trained neural network by any means other than direct testing. The intractable nature of fitness landscapes is one of the defining features of ER and any form of autonomous control learning based on interaction between agent and environment. Because of this underlying intractability, there is no great emphasis on fitness landscape analysis in ER. Further, and perhaps more importantly, attempts to make search spaces more tractable often impose biases into the evolving systems that reflect the designer's intuitive *a priori* knowledge of known solutions, thus reducing the system's ability to discover novel solutions.

Please cite this article in press as: A.L. Nelson, et al., Fitness functions in evolutionary robotics: A survey and analysis, Robotics and Autonomous Systems (2008), doi:10.1016/j.robot.2008.09.009

Table 1
Fitness function classes.

Fitness function class	*A Priori* knowledge incorporated
Training data fitness functions (for use with training data sets)	Very high
Behavioral fitness functions	High
Functional incremental fitness functions	Moderate-high
Tailored fitness functions	Moderate
Environmental incremental fitness functions	Moderate
Competitive and co-competitive selection	Very low-moderate
Aggregate fitness functions	Very low

2. Classification of fitness functions in evolutionary robotics

In this section, we present a classification system for fitness functions and review current methods used for controller fitness evaluation in evolutionary robotics. The classification hierarchy is based on the degree of *a priori* knowledge that is reflected in the fitness functions used to evolve behaviors or task performance abilities. The justification for using *a priori* knowledge as a basis for classification and organization of the research is that it reflects the level of truly novel learning that has been accomplished [106]. There are of course other means by which designers introduce their own *a priori* knowledge of task solutions into the design of experimental systems intended to study evolution (or learning) in autonomous robots. These include selection of appropriate sensors and actuators, design of training environments, and choice of initial conditions. Although these other forms of introduced *a prior* knowledge are also important (and perhaps worthy of a meta-study), it is the fitness function that contains the most explicit and varied forms of task solutions knowledge. Many of the research platforms have at least qualitative commonalities of sensor capabilities and actuator arrangements. For example in more than half of the literature survived in this review, wheeled robots that employed differential drive for steering were used.

We define seven broad classes of fitness functions. These are listed in Table 1. The characteristics of each class will be discussed in this section and a full survey of ER research in terms of particular fitness functions will follow in Section 3.

2.1. Training data fitness functions

The first class of fitness functions, those used with data sets, is not exclusive to evolutionary computing methods. Training data fitness functions are used in gradient descent training methods such as error back propagation for training neural networks, and various curve-fitting and numerical methods. Here, fitness is maximized when the system in question produces a minimum output error when presented with a given set of inputs with a known set of optimal associated outputs.

For a given problem, a training data set must include sufficient examples such that the learning system can extrapolate a valid generalizable control law. Thus, at least implicitly, an ideal training data set contains knowledge of *all* salient features of the control problem in question. For controllers that are intended to perform a complex behavior or task, sufficient training data sets are usually unavailable, and the knowledge needed to create such a data set could be used to formulate a more traditional controller. The main use of training data fitness functions in autonomous control learning is in the area of mimetic learning, where a robotic system learns to mimic behavior generated by a human or other trainer. In some sense, training data fitness functions require complete *a priori* knowledge of the task to be performed, at least insofar as it is possible to generate a suitable training data set. Robots trained with such data sets learn to duplicate an *a priori* known set of system inputs and outputs. Knowledge-based training and examples of the use of training data fitness functions in ER can be found in [23–25].

2.2. Behavioral fitness functions

Behavioral fitness functions are task-specific hand-formulated functions that measure various aspects of what a robot is doing and how it is doing it. These types of functions generally include several sub-functions or terms that are combined into a weighted sum or product. These sub-functions or terms are intended to measure simple action-response behaviors, low-level sensor-actuator mappings, or other events/features local to the robot. These will be referred to as *behavioral terms*, and measure some aspect of how a robot is acting (behaving), not what it has accomplished. In contrast, *aggregate terms* measure some aspect of what the robot has accomplished, without regard to how it was accomplished.

The quality that unifies functions in the class of behavioral fitness functions is that they are made up only of terms or components that select for behavioral features of a presupposed solution to a given task. For example, if one wished to evolve robots to move about an environment and avoid obstacles, one might include a term in the fitness selection function that is maximized if a robot turns when its forward sensors are stimulated at close range. In this case the system is set up such that robots will evolve to produce a certain actuator output in response to a given sensor input. Now, selection occurs for a behavior that the designer believes will produce the effect of obstacle avoidance, but the robots are not evolving to avoid objects *per se*, they are learning to turn when their forward sensors are stimulated. This is more specific than just selecting for robots that do not collide with objects.

Some terms in a behavioral fitness function are not selective for a precise sensor-to-actuator mapping, but rather for a desired control feature. For example, if one wished to evolve a robot controller that spent most of its time moving, one might include a term in the fitness function that is maximized when forward motion commands result in continued forward motion of the robot over time (if the front of a robot were in contact with an immobile object, it would not move forward regardless of its current actuator commands). This example term is not selective for an exact sensor-to-actuator mapping. There are other possible formulations that could also produce the desired control feature, such as a term that maximized the ratio of forward motion to forward sensor activity. Hence, this type of term does not require quite as much *a priori* knowledge of the exact details of the control law to be learned. Examples of the use of behavioral fitness functions can be found in [13,26,27].

2.3. Functional incremental fitness functions

Functional incremental fitness functions begin the evolutionary process by selecting for a simple ability upon which a more complex overall behavior can be built. Once the simple ability is evolved, the fitness function is altered or augmented to select for a more complex behavior. This sequence of evolution followed by fitness function augmentation continues until eventually the desired final behavior is achieved. The overall process can be considered one of explicit training for simple sub-behaviors followed by training for successively more complex behaviors. Often, an artificial evolution process that makes use of an incremental fitness function is referred to as *incremental evolution*.

Functional incremental fitness functions address a major difficulty in evolutionary robotics. For difficult tasks, it is possible that some or all of the controllers in a newly initialized population will possess no detectable level of ability to complete the task. Such a controller is referred to as being sub-minimally competent. If all the controllers in an initial population of controllers are sub-minimally competent for a particular fitness function, then the fitness function can generate no selective pressure and the population will fail to evolve. Functional incremental fitness functions overcome the problem of sub-minimally competent controller populations by augmenting the difficulty of the task for which the controllers are being evolved during the course of evolution.

One main criticism of using functional incremental fitness functions is that they may restrict the course of evolution to the degree that resulting controllers cannot be considered to have evolved truly novel behaviors. The designer is not only responsible for including features of a desired solution (as is the case for tailored fitness functions, discussed in the next subsection), but must also structure the search path through the controller's configuration space (search space). For non-trivial robot control tasks, there is no guarantee that this design problem is tractable. Examples of ER research making use of functional incremental fitness functions are found in [9,33,34].

2.4. Tailored fitness functions

In addition to behavior-measuring terms, tailored fitness functions contain aggregate terms that measure some degree or aspect of task completion that is divorced from any particular behavior or method. Hence, tailored fitness functions combine elements from behavioral fitness functions and aggregate fitness functions (discussed in Section 2.7 of this section). As an example, suppose a phototaxis behavior is to be evolved. A possible fitness function might contain a term that rewards a controller that arrives at the light source by any means, regardless of the specific sensor-actuator behaviors used to perform the task. This term would be considered an aggregate term. If it were the only term in the fitness function, then the whole function would be considered aggregate. If the function also contained a second behavioral term, for example, one that maximized the amount of time the robot spent pointing toward the light source, then the two terms together would constitute an example of a tailored fitness function. Note that this second term, selecting for pointing toward the light source, does represent implicit assumptions about the structure of the environment and may not be the best way to approach the light source in some complex environments.

Unlike true aggregate fitness functions, aggregate terms in tailored fitness functions may measure a degree of partial task completion in a way that injects some level of *a priori* information into the evolving controller. For example, in the phototaxis task, a tailored fitness function might contain a term that provides a scaled value depending on how close the robot came to the light source. This may seem at first glance to be free of *a priori* task solution knowledge, but it contains the information that being closer to the goal is inherently better. In an environment composed of many walls and corridors, linear distance might not be a good measure of fitness of a given robot controller. We use the term "tailored" to emphasize that these types of fitness functions are task-specific hand-formulated functions that contain various types of selection metrics, fitted or *tailored* by the designer to accommodate the given problem, and often contain solution information implicitly or explicitly. Examples of work using tailored fitness functions can be found in [28–30].

Together, tailored fitness functions and behavioral fitness functions make up by far the largest group of fitness functions used in current and past evolutionary robotics research. These types of fitness functions are formulated by trial and error based on the human designer's expertise.

2.5. Environmental incremental fitness functions

Rather than simply increasing the complexity of the fitness selection function, one form of incremental evolution involves augmenting the difficulty of the environment in which the robots must operate. This is referred to as *Environmental incremental* evolution. Environmental incremental evolution may not constrain the controller's search space to the degree that evolution must converge on a particular predefined solution. Relatively little work has been done using environmental incremental evolution. In [35] the authors used Environmental incremental selection to evolve controllers for a fairly complex peg collection task. That research showed that Environmental incremental evolution can produce robot controllers capable of expressing complex behaviors. However, it is not clear to what degree the selection and augmentation of training environments shaped the final evolved controllers. Other examples include [36,37,83].

2.6. Competitive and co-competitive fitness selection

Competitive fitness selection utilizes direct competition between members of an evolving population. Controllers in almost all ER research compete in the sense that their calculated fitness levels are compared during selection and propagation. However, in competitive evolution robot controllers compete against one another within the same environment so that the behavior of one robot directly influences the behavior, and therefore fitness evaluation, of another. For example, in a competitive goal-seeking task, one robot might keep another from performing its task by pushing it away from the goal. Here, the second robot might have received a higher fitness rating if it had not been obstructed by the first robot. Examples of the use of intra-population competition, in which the fitness of individual robots were directly affected by interaction with other robots using controllers from the same population, has been investigated in [21].

In co-competitive evolution two separate populations (performing distinct tasks) compete against each other within the same environment. Examples of co-competitive evolution involving populations of predator and prey robots exist in the literature [10,38,39,84]. Two co-evolving populations, if initialized simultaneously, stand a good chance of promoting the evolution of more complex behaviors in one another. As one population evolves greater skills, the other responds by evolving reciprocally more competent behaviors. [84] discusses this putative explanation for the selective power of competitive selection, termed the Red Queen Effect. The changing behavior of the evolving competing agents alters the fitness landscape, essentially generating a more and more arduous selection criterion but without changing the fitness function explicitly. The research presented in [10,38,39] shows this effect in evolving robot controller populations to a degree, but results from other areas of evolutionary computing suggest that given the correct evolutionary conditions, aggregate selection combined with intra-population competition within a single population performing a directly competitive task can produce very competent systems [40,41].

2.7. Aggregate fitness functions

Aggregate fitness functions select only for high-level success or failure to complete a task without regard to how the task was completed. This type of selection reduces injection of human bias into the evolving system by aggregating the evaluation of benefit (or deficit) of all of the robot's behaviors into a single success/failure term. This is sometimes called *all-in-one* evaluation. Examples of aggregate fitness selection are found in [17,16,32].

Please cite this article in press as: A.L. Nelson, et al., Fitness functions in evolutionary robotics: A survey and analysis, Robotics and Autonomous Systems (2008), doi:10.1016/j.robot.2008.09.009

Consider the following foraging task: a robot is to locate and collect objects and then deposit them at a particular location (or a "nest"). An aggregate fitness function would contain information only related to task completion. Suppose the task is considered to be complete when an object is deposited at the nest. An example of an aggregate fitness function for this task would be one that counted the number of objects at the nest after the end of a trial period.

Until recently, aggregate fitness selection was largely dismissed by the ER community. This is because initial populations of controllers generally have no detectable level of overall competence to perform non-trivial tasks (i.e. they are *sub-minimally competent*). In the example above, if the objects were sparsely distributed in a complex environment, and the controllers in the initial un-evolved population were truly randomly configured without any navigation, object recognition or homing abilities, the chances of one of them completing the task by chance are diminishingly small. This situation is often referred to as the *bootstrap problem* [31]. Completely aggregate selection produces no selective pressure in sub-minimally competent populations at the beginning of evolution and hence the process cannot get started.

Even so, aggregate fitness selection in one form or another appears to be necessary in order to generate complex controllers in the general case if one is to avoid injecting restrictive levels of human or designer bias into the resulting evolved controllers. For the evolution of truly complex behaviors, selection using behavioral fitness functions and incremental fitness functions results mainly in the optimization of human-designed controller strategies, as opposed to the evolution or learning of novel intelligent behavior.

It is possible to overcome some of the problems associated with aggregate selection. One approach is to use a tailored fitness function to train robots to the point at which they have at least the possibility of achieving a given complex task at some poor but detectable level, and then to apply aggregate success/failure selection in conjunction with intra-population competition to drive the evolutionary process to develop competent controllers. Intra-population competition presents a continually increasing task difficulty to an evolving population of controllers and may be able to generate controllers that have not been envisioned by human designers.

The chart in Fig. 1 relates the classes of fitness functions to degrees of *a priori* knowledge incorporated into evolving populations of robot controllers. The chart is qualitative and reflects general associations. Some of the fitness function classes discussed previously can be formulated to incorporate varying degrees of *a priori* knowledge into evolving populations and are depicted spanning several levels on the horizontal axis.

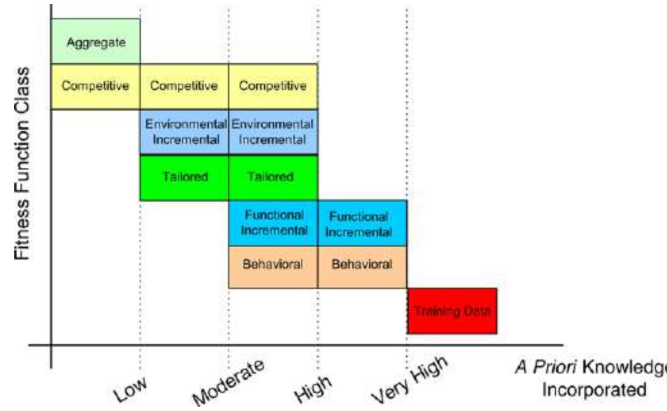


Fig. 1. Chart relating classes of fitness functions to levels of incorporated *a priori* knowledge.

3. A survey of fitness evaluation functions in ER

In this section we survey current and past ER research and organize the work using the classification system presented in Section 2. The surveyed research is listed by fitness function class and by date in Tables 3–8. A distinction is made between work that involved real robots and that in which only simulated robots were used. We have endeavored to reference most of the major research efforts that involved real robots at some level or another. Some work that was conducted only in simulation but never tested on real robots is also discussed at the end of this section. Here, though, we do not attempt a comprehensive summary of the purely simulated work.

Before we continue into our main survey and discussion of fitness functions used in ER, we will lay out some general bounds, define conventions used for symbolic representation of fitness functions, and define features or elements that are common to most of the reviewed research.

Almost all of the work considered in this survey employed some form of population-based artificial evolution in which the candidate solutions being evolved are autonomous robot controllers. Although the evolutionary algorithms vary to a degree from work to work, most of them fall within a general class of stochastic hill-climbing learning algorithms. Unless otherwise stated, the research papers reviewed here may be assumed to use an evolutionary method roughly equivalent to that which was outlined in the introduction to this paper. Population sizes vary widely. Much of the research used population sizes in the range of 20–100. Twenty to 300 generations are generally reported to be required to evolve suitable controllers for the majority of tasks investigated, but generations might range into the thousands in some cases. In a few cases the evolutionary algorithms differ significantly from standard forms and in these cases a short description of the artificial evolution methods used will be included.

As noted in the introduction, efficacy of methods beyond that of obtaining reasonably functional controllers is not a primary focus of evolutionary robotics in its current state of development. The focus, rather, is upon designing evolutionary systems able to evolve controllers capable of performing new tasks of greater complexity. It is true that some methods or robot platforms may show a 2-fold (or even 10-fold) increase in training efficiency over others, but it is the fitness function that finally determines the achievable performance.

We have endeavored to translate the diverse formulations of the fitness functions into a consistent summary representation. It is important to note that in some cases details of the original fitness functions are abstracted so that the underlying forms can be presented and compared in a standardized way. In some cases, fitness functions have been generated from a text description. This is done so that the underlying natures of functions can be compared more directly.

Where possible, f will be used to indicate instantaneous fitness. In general, fitness is maximized during an artificial evolution process, and unless otherwise stated, it will be assumed that a given fitness function is optimized when it is maximized. Those functions that are minimized are denoted with a minus sign subscript ($f_{(-)}$).

In the process of evaluating the fitness of a given robot controller, fitness functions are commonly integrated or averaged with respect to sensory-motor cycles or time steps over the course of a fitness evaluation trial period. In many cases, researchers report fitness functions that explicitly include this integration process. To facilitate comparison, and to define a simple unified

Table 2
List of common symbols used in fitness function representation.

Symbol	Meaning
F	Explicit fitness function
f	Fitness function integrand or summand
φ	Non-standard integrand or summand
f_1, f_2, f_3	Incremental fitness function integrand
d	Distance traveled
v	Velocity (drive motor)
s	Sensor activation level
B	Boolean sub-function
c	Constant coefficient

format as much as possible, the standardized representation used in this survey presents fitness functions in the form of an integrand only. Integrals or averaging summations are not explicitly symbolized in the standardized representation of these fitness functions. Integration or summing is assumed to be part of the evaluation process and is common to almost all the work surveyed. This means that for a particular fitness function integrand reported as $f(\cdot)$ in this survey, the actual fitness calculation for an evaluation period of time length N (or of N time steps) would be calculated by

$$F(t) = \frac{1}{N} \int_{t_0}^{t_N} f(t, \mathbf{q}(t)) \mathrm{d}t \tag{1}$$

or

$$F(t) = \frac{1}{N} \sum_{t_0}^{t_N} f(t, \mathbf{q}(t)) \tag{2}$$

where t, t_0, t_N, and $\mathbf{q}$ represent time, initial time, final time and a vector of other (possibly implicit) functions of time respectively. Note that in many cases t does not relate directly to physical time, but rather measures time steps in a simulation environment or measures a quantized form of time. Among others, the symbol k is used in some of the referenced works to indicate discrete time, but we use the symbol t in all formulas to facilitate comparison of general forms.

Fitness functions that cannot be reduced to an average, sum or integral are stated explicitly. In these cases an uppercase F is used and represents the entire cumulative fitness of a given individual measured over a given evaluation period. Aggregate fitness functions, for example, usually only report success or failure after a given trial evaluation period and do not represent a continuous integration or summing process.

Fitness functions whose values depend on specific events, or that use secondary summations, integrals, or other terms that are not integrated with respect to time are also reported in full form. Occasionally researchers employ fitness metrics that update fitness at specific trigger points during an evaluation period rather than at each time point. These functions will be given using a lower-case phi (φ).

Common terms and factors appear in many fitness functions and where possible we will use consistent notational conventions. These include distance traveled, speed, and sensor activation levels, and these will be represented by d, v, and s respectively. Boolean functions will be represented with an uppercase B. Constant coefficients will be represented by c. In the case of incremental fitness functions, f_1, f_2, f_3 and so on will be used to indicate the functions and their order of application. Table 2 provides a list of common symbols used in this paper in the representation of fitness functions.

Tables 3–8 list the main body of work cited in this survey, and include the fitness function class used, the author(s) and year of publication of the citation, the task or behavior investigated, the environment in which the evolution was performed, the type of robot used, and the controller architecture or learning representation used.

It is our view that evolutionary robotics work should be verified in real robots. Physical verification in real robots forces researchers to use simulations that are analogous to the physical world. Some subtleties of control are contained within the robot–world interface and are easily overlooked. In particular, adequate simulators must maintain a suitable representation of the sensory-motor-world feedback loop in which robots alter their relationship to the world by moving, and thus alter their own sensory view of the world. Robotics work involving only simulation, without physical verification, should not be considered fully validated. Much of the pure-simulation work falls into the category of artificial life (AL), and many of these simulation environments include unrealistic representation or rely on sensors that report unobtainable data or that report conceptual data. That said, learning in simulation with transfer to real robots has been repeatedly demonstrated to be viable over the last decade. Much of this work has involved new physics- and sensor-based simulators. Introducing noise into the simulation environment has been shown to aid in transference of evolved controllers from simulated to real robots [27] and continues to be studied [105]. The verification of evolved controllers in real robots allows a clear distinction to be made between the large amount of work done in the field of artificial life, and the similar, but physically grounded work pursued in evolutionary robotics.

We focus on providing a comprehensive survey of evolutionary robotics in terms of fitness functions used during training of controllers; later in this section we list these fitness functions explicitly by class. In order to give some context, we also include some details of the tasks learned by the robots, controller representations and other experimental details, but the review is not meant to comprehensively describe all aspects of the experimental procedures used by the researchers.

If included in the individual papers surveyed, we also report the population size and number of generations required for successful evolution of competent controllers. The reader may note that there is considerable variation in the number of generations required to evolve controllers for a given task. These differences reflect various aspects of the individual algorithms used, the evolvable controller structures used, and the physical robots involved. Beyond the fitness function we do not attempt to delve into the specifics of the genetic algorithms used unless the details are salient or differ significantly from the norm.

The relative efficiency of controllers evolved in different research efforts is not the central focus of comparison in this survey, or in the field of evolutionary robotics as a whole. Most of the cited research efforts did produce functional controllers. There are however a few works that did not produce robot controllers capable of achieving the particular tasks intended. These are pointed out when discussed.

The works surveyed are in some ways disparate, and an absolute direct quantitative comparison of the various results is not suggested. Complexity of tasks learned, as well as the amount of *a priori* information needed to select for a given task using a given robot system are central issues in ER. These two factors, task complexity and *a priori* information used during evolution, can form the basis of a qualitative comparison of various methods use to evolve controllers. The most advanced work is that which evolves controllers for the most complex tasks while minimizing the amount of *a priori* information contained in the fitness functions used to drive the evolutionary process. For the complexity of the simple tasks studied in current research, and common fitness function formulations, this is a viable general approach. However, as the state of the art of ER becomes more advanced, even qualitative comparison of different ER research efforts will require a greater emphasis on formal comparison methods based on task difficulty metrics and machine intelligence quotient definitions.

Please cite this article in press as: A.L. Nelson, et al., Fitness functions in evolutionary robotics: A survey and analysis, Robotics and Autonomous Systems (2008), doi:10.1016/j.robot.2008.09.009

Table 3
Summary of ER research using behavioral fitness functions.

Citation	Author(s), Year of publication	Task evolved/Learned	Embodied/Real/Simulated	Robot platform	Evolved controller type/Algorithm
[11]	Floreano and Mondada, 1996	(1) Locomotion with object avoidance (2) Locomotion with periodic goal homing	Embodied	Khepera	Neural network
[13]	Lund and Miglino, 1996	Locomotion with object avoidance	Simulated, transferred to real	Khepera	Neural network
[26]	Banzhaf et al., 1997	(1) Locomotion with object avoidance (2) object following (3) wall following (4) light avoidance	Embodied	Khepera with IR sensors	Evolvable program (GP)
[27]	Jakobi, 1998	Locomotion with object avoidance	Simulated, transferred to real	Octopod robot	Neural network
[42]	Gomi and Ide, 1998	Gait evolution	Embodied	OCT-lb, Octopod robot	Set of gait control parameters
[43]	Matellán et al., 1998	Locomotion with object avoidance	Embodied	Khepera	Fuzzy logic controller
[15]	Nordin et al., 1998	Locomotion with object avoidance	Simulated, transferred to real	Khepera	Evolvable program (GP)
[44]	Liu et al., 1999	Object pushing	Embodied	Custom built robot (JUNIOR)	Evolvable program (GP)
[45]	Seok et al., 2000	Phototaxis with obstacle avoidance	Embodied	Custom built robot	Evolvable hardware (FPGA)
[46]	Ziegler and Banzhaf, 2001	Locomotion with object avoidance	Simulated, transferred to real	Khepera	Directed graph

Table 4
Summary of ER research using functional incremental fitness functions.

Citation	Author(s)/Year of publication	Task evolved/Learned	Embodied/Real/ Simulated	Robot platform	Evolved controller type/Algorithm
[9]	Harvey et al., 1994	Differential goal homing	Embodied	Gantry robot	Neural network
[33]	Lee et al., 1997	Object pushing with goal homing	Simulated, partial transfer to real	Khepera	Evolvable program (GP)
[71]	Filliat et al., 1999	Locomotion with object avoidance	Simulated, transferred to real	SECT Hexapod robot	Neural network
[36]	Pasemann et al., 2001	Goal homing with object avoidance	Simulated, transferred to real	Khepera	Neural network
[34]	Barlow et al., 2005	Goal homing and circling	Simulated, transferred to real	EvBot	Evolvable program (GP)

3.1. Training data fitness functions

Training data fitness functions such as those used in back propagation training of neural networks require full knowledge of the solution sought in the form of a training data set. As such, these functions represent a form of solution optimization and/or *n*-dimensional surface fitting. We mention these here for completeness and context, but these methods fall outside the focus of this review and of ER, and they cannot be used to discover intelligent control solutions whose features are not captured in a training data set (and therefore known *a priori* at some level).

In mimetic methods, a training data set is generated by recording the sensor inputs and motor outputs of a system while it is performing a particular task. Such data sets are often derived from a teleoperated system controlled by a human, and the resulting trained systems in effect learn to mimic a particular example of a human performing the task.

Also along these lines, *breeder* or *clicker* training does not use a specific training data set, but requires a human trainer to provide fitness feedback during training [99,100]. In essence, a new set of training examples is created and coded (evaluated as positive or negative) during each training session. In breeder training, the trainer need not be able to define an explicit fitness function, but he or she must still rely on his or her own *a priori* knowledge of how to perform the task which the agent is being trained to perform.

3.2. Behavioral fitness functions

Behavioral fitness functions measure fitness by measuring qualities or features of how a robot behaves while that robot is attempting to perform a task. Behavioral fitness functions do not directly measure how well the robot has accomplished its overall task *per se*. Task completion is measured implicitly by the terms that measure various aspects of the robot's behavior (see Section 2 for an illustrative example). Research that employed behavioral fitness functions is summarized in Table 3.

In [11], an experiment is discussed in which neural network-based controllers for a Khepera robot were evolved to perform a navigation and obstacle avoidance task. During the experiment, the robot (or more precisely, a population of neural networks) learned to navigate around a maze-like environment with a single closed loop, and to avoid bumping into walls while doing so. The robot was equipped with IR sensors for detection of its environment. The fitness function integrand used to select the fittest controllers during evolution was

$$f = \text{mean}(v_l, v_r)(1 - \sqrt{|v_l - v_r|})(1 - s_{ir}) \quad (3)$$

where v_l and v_r are left and right drive motor speeds, and s_{ir} is the greatest current activation level of the IR sensors. This is considered a behavioral fitness function because it bases fitness on local motor behaviors and sensor responses and does not directly

Please cite this article in press as: A.L. Nelson, et al., Fitness functions in evolutionary robotics: A survey and analysis, Robotics and Autonomous Systems (2008), doi:10.1016/j.robot.2008.09.009

Table 5
Summary of ER research using tailored fitness functions.

Citation	Author(s), Year of publication	Task evolved/Learned	Embodied/Real/ Simulated	Robot platform	Evolved controller type/Algorithm
[47]	Hoffmann and Pfister, 1996	Goal homing with object avoidance	Simulated, transferred to real	Custom lab robot	Fuzzy logic controller
[48]	Thompson, 1996	Locomotion with object avoidance	Simulated, transferred to real	Sussex Mr. Chips robot	Evolvable hardware (FPGA)
[28]	Schultz et al., 1996	Agent herding	Simulated, transferred to real	Nomad 200	Evolvable rule set
[14]	Nolfi, 1997	Foraging with object deposition	Simulated, transferred to real	Khepera with gripper	Neural network
[29]	Keymeulen et al., 1998	Target homing with obstacle avoidance	Simulated, transferred to real	Custom lab robot	Evolvable hardware (FPGA)
[49]	Ishiguro et al., 1999	Object pushing with goal homing	Simulated, transferred to real	Khepera	Neural network
[50]	Ebner and Zell, 1999	Locomotion with object avoidance	Simulated, transferred to real	RWI B21	Evolvable program (GP)
[20]	Floreano and Urzelai, 2000	Sequential goal homing	Embodied	Khepera, Koala	Neural network
[52]	Sprinkhuizen-Kuyper et al., 2000	Object pushing	Simulated, transferred to real	Khepera	Neural network
[53]	Wolff and Nordin, 2001	Gait optimization	Embodied	ElVINA (biped)	Gait parameter set
[54]	Nehmzow, 2002	(1) photo-orientation (2) object avoidance (3) robot seeking	Embodied	Custom LEGO robots	Evolvable program (GP)
[12]	Watson et al., 2002	Phototaxis	Embodied	Custom robot	Neural network
[56]	Marocco and Floreano, 2002	Locomotion with wall avoidance	Embodied	Koala	Neural network
[57]	Okura et al., 2003	Locomotion with object avoidance	Embodied	Khepera	Evolvable hardware (FPGA)
[30]	Quinn et al., 2002	Coordinated movement	Simulated, transferred to real	Custom robots	Neural network
[58]	Gu et al., 2003	Object (ball) homing	Embodied	Sony AIBO	Evolvable fuzzy logic controller
[55]	Simões and Barone, 2004	Locomotion with object avoidance	Embodied	Custom robots	Neural network
[59]	Nelson et al., 2004	Locomotion with object avoidance	Simulated, transferred to real	EvBot	Neural network
[60]	Boeing et al., 2004	Gait evolution	Simulated, transferred to real	Andy Droid robot	Spline controller
[61]	Hornby et al., 2005	Gait evolution	Embodied	Sony AIBO	Gait parameter set
[62]	Kamio and Iba, 2005	Object pushing with goal homing	Simulated, transferred to real	Sony AIBO, HOAP-1	Evolvable program (GP)
[22]	Capi and Doya, 2005	Triple sequential goal homing	Simulated, transferred to real	Cyber Rodent	Neural network
[63]	Parker and Georgescu, 2005	Phototaxis with obstacle avoidance	Simulated, transferred to real	LEGO Mindstorm	Evolvable program (GP)
[110]	Trianni and Dorigo, 2006	Coordinated locomotion with hole avoidance	Simulated, transferred to real	Swarm-bot	Neural network

Table 6
Summary of ER research using environmental incremental fitness functions.

Citation	Author(s)/Year of publication	Task evolved/Learned	Embodied/Real/Simulated	Robot platform	Evolved controller type/Algorithm
[37]	Miglino et al., 1998	Goal homing with object avoidance	Simulated, transferred to real	Khepera	Neural network
[35]	Nakamura, 2000	Foraging with object carrying	Simulation only	Simulated Khepera	Neural network

Table 7
Summary of ER research using competitive fitness functions.

Citation	Author(s)/Year of publication	Task evolved/Learned	Embodied/Real/ Simulated	Robot platform	Evolved controller type/Algorithm
[10]	Nolfi and Floreano, 1998	Pursuit and evasion	Embodied	Khepera	Neural network
[21]	Nelson and Grant, 2006	Competitive goal homing with object avoidance	Simulated, transferred to real	EvBot	Neural network

Table 8
Summary of ER research using aggregate fitness functions.

Citation	Author(s), Year of publication	Task evolved/Learned	Embodied/Real/Simulated	Robot platform	Evolved controller type/Algorithm
[17]	Hornby et al., 2000	Object pushing	Simulated, transferred to real	Sony AIBO	Neural network
[64]	Earon et al., 2000	Gait evolution	Embodied	Hexapod robot Kafka	Evolvable state lookup tables
[16]	Lipson and Pollack, 2000	Locomotion (co-evolution of body)	Simulated, transferred to real	Auto-fabricated modular robots	Neural network
[65]	Hornby et al., 2001	Locomotion (co-evolution of body)	Simulated, transferred to real	TinkerBot modular robots	Actuator control parameter set
[32]	Hoffmann and Montealegre, 2001	Locomotion with object avoidance	Embodied	LEGO Mindstorm	Evolvable sensor-to-motor excitation mapping
[66]	Augustsson et al., 2002	Flying lift generation	Embodied	Winged robot	Genetic programming
[67]	Zufferey et al., 2002	Locomotion with wall avoidance	Embodied	Robotic blimp	Neural network
[68]	Macinnes and Paolo, 2004	Locomotion (co-evolution of body)	Simulated, transferred to real	LEGO-servo modular robots	Neural network
[69]	Zykov et al., 2004	Gait evolution	Embodied	Pneumatic Hexapod robot	Gait parameter set
[70]	Chernova and Veloso, 2004	Gait evolution	Embodied	Sony AIBO	Gait parameter set

measure partial or overall task completion. At each generation during the evolutionary process every network in the controller population was tested on a real robot in a real environment. Evolution performed without the use of a simulator, as in this case, is referred to as embodied evolution. The researchers reported that after the 50th generation, the fittest evolved neural network-based controller performed the task at near optimum levels and was able to travel around its environment indefinitely without colliding with walls or getting stuck in corners.

A further experiment using the same platform was also discussed in [11]. Neural networks were again evolved to control a Khepera robot. The task for the robot was a periodic goal homing behavior in which the robot was to travel about an arena for a period of time and then move to a goal location and remain there for a short time. The goal location was marked by a light source and the robot was equipped with photosensors in addition to its IR sensors. The motivation for the experiment was to evolve a behavior that could allow a robot to return to a battery recharging station, hence the robot was given a simulated energy level that would fall to zero after a period of time. The fitness function integrand used was

$$f = \text{mean}(v_1, v_2)(1 - s_{ir}). \tag{4}$$

Note that a robot that recharges its virtual energy level will achieve a greater mean velocity over a long evaluation period than one that runs out of energy too far away from the recharging station. In addition, the recharge station was placed next to a wall so that robots must spend time away from it to maximize the $(1 - s_{ir})$ factor of f. As in the first experiment in [11], embodied evolution was employed. Evolution of successful controllers took 10 h of testing time with the real robot and represented 240 generations with a population of 100 controllers.

In [13], experiments on a locomotion and object avoidance task similar to the work presented in [11] were reported. Simple neural networks with no hidden layers were evolved to perform the task. The work also used the Khepera robot platform and evolution was conducted using a behavioral fitness function integrand similar to that used in [11]:

$$f = \text{mean}(v_l, v_r)(1 - (v_l - v_r)^2)(1 - s_{ir}) \tag{5}$$

where v_l and v_r are left and right drive motor speeds, and s_{ir} is the greatest current activation level of the IR sensors. Using a population of 100 neural network controllers, evolution was initially performed in a simulation environment for 200 generations, and then optimized in a real robot for an additional 20 generations. Robots with the fittest evolved neural controllers were reported to be able to perform their intended task reliably in a real environment using the real robot.

[26] evolved four separate behaviors using embodied evolution and genetic programming (GP). A Khepera equipped with 8 IR proximity sensors was used. All of the fitness functions used were behavioral and couched in terms of function regression mapping sensor inputs to actuator outputs. The fitness functions used did not use measurements of direct task completion for fitness evaluation, but rather they selected for sensory-motor behaviors that the researchers deemed would produce the ability to perform the tasks. The behaviors evolved were forward motion with object avoidance, object homing/following, wall following, and hiding in the dark (light avoiding). The four fitness function integrands used for the four behaviors follow respectively:

$$f_{(-)} = s_{ir} - (v_l + v_r - |v_l - v_r|) \tag{6}$$

where s_{ir} is the sum of the activations of all of the IR sensors and v_l, v_r are the left- and right-hand motor velocities;

$$f_{(-)} = (s_{ir1} + s_{ir2} + s_{ir3} + s_{ir4} - c)^2 \tag{7}$$

where c is a constant picked so that the robot will learn to follow a distance behind the object such that the sum of the four forward facing sensors is near c;

$$f_{(-)} = (s_{ir1} - c_1)^2 + (s_{ir2} - c_2)^2 + {s_{ir3}}^2 - (v_l + v_r)^2 \tag{8}$$

where s_{ir1} and s_{ir2} are sensor activations on the wall-side of the robot, s_{ir3} is a sensor on the outward facing side, and c_1 and c_2 are constants; and

$$f_{(-)} = s_{\text{photo}} - (v_l + v_r - |v_l - v_r|) \tag{9}$$

where s_{photo} is the activation of a photosensor. The authors report successful evolution of these four behaviors, using purely reactive and memory-based machine language GP formulations, but no specific training data were presented.

For very simple tasks, one might define the task to be exactly that which is accomplished by producing a particular sensory-motor behavior. In cases where there is no distinction between an overall task description and the low-level sensory-motor behavior, the task will be classified as behavioral.

In [27], locomotion with obstacle avoidance in legged robots was evolved. Robot controllers were evolved in a minimal simulation environment and then transferred to a real robot for verification. The robot had IR sensors on the right- and left-hand sides and a tactile bumper on the front. A behavioral fitness function is used in this work and has four cases, each designed to calculate fitness to perform a desired aspect of the task. The author defines this in terms of an extended case statement. This can be represented as an integrand of four terms with mutually exclusive Boolean coefficients as follows:

$$f = B_1(v_l + v_r) + B_2(v_l - v_r) + B_3(-v_l + v_r) + B_4(-v_l - v_r) \tag{10}$$

where v_l and v_r are the left- and right-hand-side velocities of the robot, and B_1–B_4 are mutually exclusive Boolean coefficients that are non-zero under the following conditions: B_1 is non-zero when no obstacles are in range of the IR sensors and the bump sensor is not engaged, B_2 is non-zero when there is an object in range of the right-hand IR sensor, B_3 is non-zero when there is an object in range of the left-hand IR sensor, and B_4 is non-zero when the bump sensor is engaged. The target behavior is defined at the level of sensor readings and robot body responses in each of the cases and the fitness is formulated to select for these only. Further, evolution took place in a carefully structured minimal simulation environment in which only dynamics that the designers believed would be relevant to an optimal solution were reproduced. All other dynamics were simply structured to produce a very low fitness in the simulated robot. As with a few of the other evolutionary robotics experiments that used very solution-specific fitness functions, in this work, a novel solution cannot be considered to have been truly learned by the system. Rather, the system has been programmed in a roundabout way to reproduce a particular *a priori* known solution. Another unusual feature of this work is that 3500 generations were used to develop controllers that produced effective locomotion in the real robot. This is between 10 and 100 times the number required for most other similar reported experiments. However, it should be noted that this is one of only a handful of research efforts to investigate intelligent control learning in an octopod.

In [42], hexapod gaits were evolved using a real hexapod robot. The behavioral fitness function used was one of the most complicated found in the literature for the task of legged-robot locomotion. We only summarize it here:

$$F = (strides)(1 - overcurrent)(balanced)(1 - bellytouch) \tag{11}$$

where each of the terms is a function based on a combination of the robot's behavior and sensor inputs. The function *strides* counts leg cycle movements, *overcurrent* measures actuator commands that exceed the current capacity of the leg motors, *balanced* measures the degree of tilt of the robot body, and *bellytouch* counts the number of times the robot's body falls low enough to scrape on the floor. The hexapod was able to evolve efficient gaits within fifty generations. As is the case with several of the other gait-learning research examples, this robot had no sensors and thus did not learn to react intelligently or dynamically to the environment *per se*. In contrast to this very complex fitness function, similar examples of gait learning have been achieved using aggregate fitness functions (see Section 3.8).

A population of fuzzy logic rule-based controllers was evolved in [43]. The robot task was locomotion and object avoidance. The fitness function integrand includes a parsimony term to reduce the number of rules in the controller fuzzy rule set:

$$f = \frac{\text{mean}(v_l, v_r)(1 - |v_l - v_r|)(1 - s_{ir})}{|rules|} \tag{12}$$

where v_l and v_r are left and right drive motor speeds, s_{ir} is the greatest current activation level of the IR sensors, and |rules| counts the number of rules in the controller's fuzzy logic rule set. A population of 100 individuals was evolved for 100 generations in a real Khepera robot, and the authors report that successful controllers able to travel around their environment without colliding with obstacles are developed by the 60th generation.

In [15] genetic programming was used to evolve locomotion and object avoidance in Khepera robots. A behavioral fitness function was used and is given by

$$f = c(|v_l - v_r| + |v_r| + |v_l| - (v_l - v_r)) + \sum s_{ir} \tag{13}$$

where v_l and v_r are the left and right wheel motor speeds, and $\sum s_{ir}$ represents the sum of the activations of the proximity sensors. The authors used an extremely large population size of 10 000 and ran evolutions for 250 generations. They repeated evolution runs 100 times in simulation starting with different seed populations and reported that 82 out of the 100 runs produced useful controllers able to perform the task of obstacle avoidance while traveling about a small environment with a single circular path.

Controllers for a wall-following behavior were also evolved in [15]. A very complex conditional fitness selection method that specifies desired responses to possible sensor activation patterns was used. This fitness selection method essentially specified the solution to be evolved and injected a very high level of *a priori* knowledge into the evolved controllers.

The authors of [44] describe an object-pushing task in terms of a sumo-robot behavior. Controllers were evolved to push objects out of an arena. GP was used to evolve controllers composed of behavioral primitives such as "more forward" and "left-hand turn". The fitness function integrand used here counts the number of active sensors, the number of arms in contact with the object, and the number of arms holding the object:

$$f = \sum s_{\text{active}} + \sum arms_{\text{holding}} + \sum arms_{\text{touching}} \tag{14}$$

where $\sum s_{\text{active}}$ is the number of active proximity sensors, $\sum arms_{\text{holding}}$ is the number of arms applying side pressure to the object, and $\sum arms_{\text{touching}}$ is the number of arms in contact with the object. The researchers reported that the robot was able to learn to push objects using the fitness function in (14).

[45] presents the evolution of a phototaxis and object avoidance behavior in a robot equipped with sonar and photosensors. A genetic programming structure implemented on an FPGA was used for the controller architecture. The behavioral fitness function used here is unusual in that it includes fitness values measured at previous time steps. The function is summarized as:

$$\varphi(t+1) = c_1\left(\varphi(t) + c_2(s_{\text{photo\_max}} - s_{\text{photo}}) + c_3\frac{\sum s_{\text{sonar}}}{s_{\text{sonar\_max}}} + c_4\right) \tag{15}$$

where s_{photo}, $s_{\text{photo\_max}}$, $\sum s_{\text{sonar}}$, $s_{\text{sonar\_max}}$ are the forward photo-detector excitation, the maximum photo-detector excitation, the sum of the sonar sensor excitations, and the maximum sonar excitations respectively. Notation note: the function is not in the form of an integrand (f) or an overall function (F), rather, it is presented in the original work as a recursive function and is here denoted by φ. Learning required 300 generations in addition to a 35-generation sensor tuning phase to develop functional controllers for the task.

[46] also evolved controllers for locomotion and object avoidance. The evolvable controller architecture was described in terms of artificial chemistries, a form of optimization algorithm based on the concepts of binding and reaction of compounds in chemistry. A Khepera with IR sensors was used. The evolvable controller architecture was a form of evolvable directed graph similar to a finite state machine. A behavioral fitness function with one term was used. The fitness function integrand minimizes the sum of differences between wheel speeds:

$$f_{(-)} = |v_l - v_r| \tag{16}$$

where v_l and v_r denote the right and left wheel motor speeds. Note that unlike most of the other locomotion and object avoidance experiments, no sensor activation term was used here, but the controllers still evolved successfully over the course of 160 generations. Controllers were evolved in simulation and the best resulting controllers were transferred to a real robot and tested in a small maze environment.

3.3. Functional incremental fitness functions

In this section we present ER research that used incremental fitness functions (summarized in Table 4). Recall that incremental fitness functions begin the evolutionary process in a form that selects for a simpler behavior than the final desired behavior, and then change their form to select for complex abilities. The function may change forms several times before the final task or behavior is achieved. To begin with, we will discuss early research from the mid-1990's in some detail (in particular the research performed at the University of Sussex [9]).

In [9] a differential goal homing task was investigated in which a robot must move toward a triangular target placed on a white wall while avoiding a rectangular target. The robot used in this work consisted of a camera mounted on an X–Y positioning gantry system. The gantry was placed over an arena, and evolution was performed with trials evaluated using the physical system. The work used a three phase functional incremental fitness function.

The first sub-function maximized the distance from the arena wall opposite the target by summing the robot's current distance from the wall (d_{wall}) at 20 time points over the course of a trial:

$$F_1 = \sum_{i=1}^{20} d_{\text{wall}}. \tag{17}$$

Here, we explicitly include the summation over 20 steps since it does not represent a true averaging of fitness and is trial-time dependent. F_1 might be considered a behavioral function. After fitness converged using F_1, the fitness function was replaced with F_2 and evolution was continued with a new population derived from the best-performing member of the first evolved population:

$$F_2 = \sum_{i=1}^{20} (-d_{\text{target}}). \tag{18}$$

F_2 is maximized when the distances d_{target} to the target (measured over the course of a trial) are minimized. F_2 might also be considered a behavioral function because it does not explicitly measure task completion. Note that the final form of an incremental fitness function can be classified as one of the non-incremental forms from the classification system of Section 2, but the intermediate forms are not necessarily classifiable unless they are intended to generate a specific behavior or task.

Likewise, a third fitness function was applied to a population derived from the best performing member of the previous population. Here, the single target was replaced with two targets, one triangular and one rectangular. F_3 is maximized when the distance from the triangular target is minimized and the distance from rectangle target is maximized (measured at 20 time points over the course of each trial):

$$F_3 = \sum_{i=1}^{20} \left(c_1(D_1 - d_{1i}) - c_2(D_2 - d_{2i})\right). \tag{19}$$

Here c_1 and c_2 are experimentally derived coefficients, D_1 and D_2 are the initial distances from the triangular and rectangular targets respectively, and d_{1i} and d_{2i} are the test point distances measured over the i time steps of each trial. Dynamic recurrent neural networks were evolved and were provided only two inputs from the camera. The areas within the camera's receptive field that led to activation of the two network inputs were also modified by the evolutionary process. Successful evolution of controllers capable of identifying and homing in on the correct target was reported after a total of 60 generations (20 generations using each of the three fitness functions). The three functions used might be considered together to be behavioral fitness functions because they do not measure task completion directly.

In [33] functional incremental evolution was applied to evolve a box-pushing and goal homing behavior for a Khepera robot. Genetic programming was used and controllers were encoded by tree representations that generated purely reactive controllers. A series of three incremental functions were applied to evolve the final behavior. Note that the authors minimized these functions during evolution rather than maximizing them. The fitness function integrands $f_{1(-)}$ and $f_{2(-)}$ were used to evolve the separate primitive behavior controllers of pushing a box in a straight line and box circumnavigation, respectively. $f_{1(-)}$ is given by

$$f_{1(-)} = c_1(1 - \text{mean}(v_l, v_r)) + c_2(|v_l - v_r|) + c_3(1 - s_{f\_ir}) \tag{20}$$

where v_l, v_r, c_1, c_2, and c_3 are used as in the previously presented standardized fitness function forms, and s_{f_ir} is the average current activation level of the two forward-most IR sensors on the robot. $f_{2(-)}$ is given by

$$f_{2(-)} = c_1(1 - \text{mean}(v_l, v_r)) + c_2(|s_{s\_ir} - c_3|) \tag{21}$$

where s_{s_ir} is the activation of a particular one of the IR sensors on one side of the robot that the designers chose to act as a distance regulator between the box and the robot. The constant values c_1, c_2, and c_3 selected in f_1 are different than those selected in f_2. Both functions can be classified as behavioral. A third controller evolution was performed to generate an arbitrator controller that was responsible for turning the primitive behaviors on and off, to produce the final goal homing behavior. The fitness function used was

$$F_3 = d_{\text{box,goal}} \tag{22}$$

where $d_{\text{box,goal}}$ measures the distance between the box and the goal location (indicated by a light source) at each time point during an evaluation trial. The final form of the function could be considered aggregate if the task were defined as getting the box as close to the goal as possible. It should be noted that only the primitive behaviors were tested in real robots, so it is not clear that the final evolutionary step was entirely successful.

The research reported in [71] used a two stage functional incremental fitness function to evolve locomotion and object avoidance abilities in a hexapod robot constructed using hobby servos. The robot was equipped with IR and photosensors. The overall behavior was evolved in two steps. First, legged locomotion was evolved using a fitness function of the form

$$F_1 = c_1 d + c_2 L \tag{23}$$

where d is the maximum distance achieved by the robot and L is a measure of activation to the leg actuators (the exact forms of the fitness functions used are not explicitly stated by the authors and the functions presented here were extrapolated from the text descriptions). The second stage of evolution generated object avoidance and made use of a simple single-term fitness function, F_2, for selection:

$$F_2 = d_{\text{collision}} \tag{24}$$

where $d_{\text{collision}}$ is the distance covered by the robot before it hits an obstacle during a given evaluation trial. Note that this is very close to being an aggregate fitness function but it does not measure full task completion directly. It makes the implicit assumption that avoiding collisions will result in the best locomotion. Controllers

Please cite this article in press as: A.L. Nelson, et al., Fitness functions in evolutionary robotics: A survey and analysis, Robotics and Autonomous Systems (2008), doi:10.1016/j.robot.2008.09.009

were evolved in simulation and transferred to a real hexapod robot for testing. A neural network for locomotion was generated in the first phase of evolution, and then a second network was piggybacked on the first and evolved to generate the final collision avoidance behavior. It is not clear from this particular work how many generations were needed for each step of evolution.

In [36] populations of neural network-based controllers for Khepera robots were evolved to perform goal homing (phototaxis) and obstacle avoidance using incremental evolution in two stages. For this experiment, the robots used photosensors and IR sensors, but neural connections for the photosensors were not introduced until the second stage of evolution. Initially, controllers were evolved for straight-line motion with obstacle avoidance using the following fitness function integrand:

$$f_1 = c_1(v_l + v_r) - c_2(|v_l - v_r|). \tag{25}$$

The best controllers resulting from f_1 were then used as a seed population and were evolved further with f_2:

$$f_2 = c_1 s_{\text{front}} + c_2\left(|s_{\text{front}} - s_{\text{back}}|\right) \tag{26}$$

where s_{front} and s_{back} are the activation levels of the forward and backward photosensor arrays respectively. A population of 30 network controllers was evolved first for 100 generations with f_1 and then for an additional 100 generations with f_2. Both of these fitness functions are classified as behavioral.

The experiments discussed in [36] also contained an aspect of environmental incremental evolution. The experimenters placed additional obstacles in the environment and reduced the number of goal light sources over the course of evolution. This provided an environment of incrementally increasing difficulty. The best resulting controllers were tested in real robots and were demonstrated to be able to home in on a light source in environments containing obstacles (walls) arranged to force the robot to backtrack, and at times to explore dead ends in order to find the light source goal. [34] presents the evolution of a flight controller for beacon homing and circling. The controllers were evolved in simulation and then tested in a real ground robot that homed in on and circled a sonic beacon (EvBot II equipped with directional sound sensors). A combination of incremental and multi-objective selection was employed, using the following fitness function integrands:

$$f_{1(-)} = \frac{d}{d_0} \tag{27}$$

$$f_{2(-)} = B_{\text{inrange}} d^2 \tag{28}$$

$$f_3 = (1 - B_{\text{inrange}}) B_{\text{level}} \tag{29}$$

$$f_{4(-)} = B_{10^\circ\text{turn}} \left|\theta(t) - \theta(t-1)\right| \tag{30}$$

where d_0 is the initial position of the robot, d is the current position of the robot, and $\theta(t)$ gives the roll angle of the robot at time t. B_{inrange}, B_{level}, and $B_{10^\circ\text{turn}}$ are Boolean functions that are true when the robot is in range of the beacon, when the robot is level, and when the turn angle is greater than 10 deg respectively. Note that f_1, f_2, and f_4 are minimized while f_3 is maximized. Initially, f_1 was used to bootstrap evolution for 200 generations. After a level of homing confidence was gained, multi-objective optimization used all four fitness functions for 400 generations. In some ways, multi-objective optimization raises questions about the relationship between task definition and the definition of aggregate selection. If the task is explicitly defined in terms of solution features and the designer formulates these features into a set of multiple objectives, then this set of multiple objectives can be considered an aggregate selection mechanism when viewed as a whole. However, the choices made about which features of the solution should be jointly optimized, and the formulations of the functions for the objectives, requires a considerable amount of *a priori* task solution knowledge.

3.4. Tailored fitness functions

Tailored fitness functions contain aggregate terms that measure some level of task completion, but they may also contain behavioral terms. Aggregate functions that measure nothing but final task completion are classified separately and are presented in Section 3.8 of this section. Research using tailored fitness functions is summarized in Table 5.

In [47], Fuzzy logic rule sets were evolved to control a robot performing a goal homing and object avoidance task. A tailored fitness function with no behavioral terms was used.

The overall fitness function was stated in terms of two mutually exclusive cases, each with its own sub-function:

Case (1) A collision occurs:

$$F_{\text{collision}} = \frac{t_{\text{collision}}}{t_{\max}} \tag{31}$$

where $t_{\text{collision}}$ is the time at which the collision occurred and $t_{\max}$ is the maximum allowable number of time steps.

Case (2) No collision occurs:

$$F_{\text{free}} = \left(1 - \frac{d}{d_0}\right) \tag{32}$$

where d is the distance remaining between the goal and the robot, and d_0 is the initial distance between the robot and the goal at the beginning of the trial. A trial ended if a collision occurred, if the robot arrived at the goal location, or at $t_{\max}$.

After 30 generations controllers were evolved that were capable of reaching the goal location without collision in approximately 50% of the trials. This level of success is lower than that reported in other similar work [36,37], but at the same time, the evolutionary process was only allowed to continue for 30 generations. It is possible that controller populations had not yet converged.

In [48] the author evolves a simple locomotion with wall avoidance behavior in a robot using an evolvable hardware control system. The following tailored fitness function integrand was used:

$$f = \mathrm{e}^{-c_1(d_x)^2} + \mathrm{e}^{-c_2(d_y)^2} + B \tag{33}$$

where d_x and d_y are the distances from the center of the arena to the robot's current position, and B is a Boolean whose value is 1 if the robot is moving and 0 otherwise. The function is intended to keep the robot away from its starting position (the middle of an arena) and also keep it moving. This work was performed in a laboratory robot using sonar sensors. An unusual real robot and simulator evolutionary setup was used here. A real hardware controller using an FPGA was attached to a real robot during controller evaluation, but the robot was placed on a platform that did not allow its wheels to touch the floor. Wheel movements were measured and then fed into a simulator for final fitness evaluation. The motivation for doing this was to avoid simulating the entire dynamic controller-robot hardware system, but still allow the evolutionary process to proceed in a fully automated way. Functional controllers were evolved in 35 generations and tested in the real robot operating in a real environment.

In [28] an agent-herding task was investigated. A shepherd robot was evolved to herd a single agent with a fixed control strategy into a goal area. This work used two Nomad 200 robots, and along with [50] is one of the few evolutionary robotics experiments to use full-sized laboratory robots. The evolved control strategies consisted of a set of sensory stimulus-motor response rule sets. Controller rule sets were evolved in simulation and tested in the real robots. Fitness was measured in terms of percent, denoting partial or full success: 100% was given if the robot herded the sheep agent into the goal area before t reached $t_{\max}$. A constant c% was given if the robot herded the sheep agent to within a predefined distance of the goal area when t reached t_{max} (with c selected to be less than 100 but greater than 0), and 0% was given in all other cases. This is essentially a tailored fitness function with two aggregate terms. In tests of the best evolved controller in the real robots, the herding robot was able to herd its companion robot into the goal area in 67% of test cases.

[14] reports on the evolution of a task in which a robot must pick up pegs in an arena and deposit them outside the arena. A Khepera robot with an attached gripper unit was used to test controllers evolved in a simulation environment. Sub-behaviors were identified and evolved (and one was hand-coded). A master coordination module was evolved to generate the final overall behavior. The method of fitness evaluation used here was extremely complex and must be considered more of a tailored and behavioral algorithm than a single function. Although in theory there could be many methods of performing the target task, the fitness selection algorithm allowed only one overall solution to evolve: the robot wanders through the arena environment until it detects a peg, it picks up the peg, it moves to the edge of the arena, and it drops the peg. The authors performed 10 separate evolutions of 1000 generations, using a population size of 100 individuals. They report the emergence of individuals capable of completing the task reliably (in over 90% of attempts) in simulation and in testing on the real robot. The peg collection and deposition task evolved here is among the most complex achieved to date. [35] investigated an almost identical task in simulation only, but used an aggregate success/failure fitness function in conjunction with environmental incremental evolution (discussed in Section 3.6 of this section).

[29] presents the evolution and testing of FPGA hardware controllers for a small robot with two cameras. The robot's task was to home in on a target object in an environment containing additional obstacles. The tailored fitness function required 64 full evaluation periods to be completed before application, and included an aggregate term that counted the number of outright completions of the task:

$$F_{64\_trials} = \sum_{1}^{64} \left(B_{goal\_found} + c_1 \left(1 - \frac{d_{goal}}{d_{max}} \right) + c_2 \left(1 - \frac{t_{goal}}{t_{max}} \right) \right) \quad (34)$$

where B_{goal_found} is a Boolean that is true if the robot found the target object during that trial, d_{goal} is the remaining distance between the robot and the goal at the end of the trial, t_{goal} is the number of time steps required to find the goal and d_{max} is the greatest travel distance (linear offset) that can be achieved in the training environment. Several variations on evolutionary conditions and methods were investigated, and the reader is referred to the paper [29] for a description of these. Using the most successful evolution strategy, the authors report that the best controllers from a population of 20 evolved for 650 generations and tested on a real robot were able to complete the task of finding the target object in a fairly complex environment containing many walls and starting from 64 separate initial positions within the environment.

In [49], controllers for Khepera robots were evolved for an object pushing and phototaxis task. In this task, robots locate pegs (small cylinders) within an environment and then push them to a goal location marked by a light source. For this experiment, the controllers were evolved in simulation and then transferred to real robots. The authors used the following fitness function:

$$F = c_1 \left(1 - \frac{d_f(\text{peg, goal})}{d_0(\text{peg, goal})} \right)^2 \quad (35)$$

where d_0 and d_f measure the distance between the peg and the goal light source at the beginning and the end of an evaluation trial. F falls into the class of tailored fitness functions because it measures some degree of success. The function does not inject a high degree of *a priori* knowledge into the controller strategies. In addition, as denoted by the upper case F, this is the full fitness function, not an integrand. In [49], populations of 100 standard feedforward networks and 100 partially Hebbian neural controller architectures were evolved in separate experiments for 200 generations. Following evolution, the qualities of the resulting controllers were compared. The best-evolved Hebbian controllers, which allow for some modulation of behavior during controller evaluation (often referred to as lifetime learning [87]), were more robust to noise in the robot's actuator system. The best Hebbian controllers were able to complete the task in 92% of trials (with actuator system noise) while the standard feedforward networks were only able to complete the task 75% of the time.

A controller for locomotion with object avoidance was evolved in simulation and transferred to an RWI B21 service robot equipped with sonar [50]. This robot is larger than most robots used in ER work (over a meter tall) and was tested in a building hallway, rather than in small specially constructed arenas used in other work. GP was used to evolve populations of controller programs and the authors formulated a complex fitness function for the relatively simple task. The tailored fitness function used here maximized time of motion while minimizing overall rotation of the robot over the course of a trial evaluation period. This can be written as:

$$F = 1 - \frac{t}{t_{max}} \sqrt{\frac{\left|\sum r\right|}{\omega_{net} t_{max}}} \quad (36)$$

where t is the time of the first collision, t_{max} is the maximum time allotted for an evaluation period, $\sum r$ is the sum of all discrete rotations made by the robot, and ω_{net} is the average angular velocity over the course of an evaluation period. One of the controller evolutions was performed on the real RWI B21 robot and required approximately 200 h of evolution time to perform 50 generations with a population size of 75. This highlights the need for high quality simulators if ER is to be used to generate controllers for robust robots operating in non-laboratory environments.

In [20] the authors report on the embodied evolution of a robot behavior in which a Khepera robot moves to an intermediate goal zone and then to a final home goal position. When the robot arrives at the intermediate goal position a light source is triggered. The authors used a simple tailored fitness function that injected limited amounts of *a priori* knowledge into the evolved solutions:

$$F = \frac{t_{goal}}{t_{max}} \quad (37)$$

where t_{goal} is the number of time steps spent in the home goal position after the light has been triggered, and t_{max} is the total number of time steps per trial. As in [49], Hebbian neural networks were evolved for the task, and the learning rates of the networks were evolved, rather than the connection weights. Hence, the controllers were evolved to *learn* how to perform their task while in operation. Using a population size of 100 individuals, 500 generations were required to evolve competent networks. After evolution using Khepera robots, the fittest evolved networks were tested in Koala robots with similar sensor and actuator architectures and were found, as in [49], to be robust against changes in actuator response.

[52] describes experiments that evolve neural controllers for a box-pushing task. Here, the robot (a Khepera with IR sensors) must push an object toward a light source. The authors evolved separate populations of controllers using four different fitness

Please cite this article in press as: A.L. Nelson, et al., Fitness functions in evolutionary robotics: A survey and analysis, Robotics and Autonomous Systems (2008), doi:10.1016/j.robot.2008.09.009

function integrands and then compared the quality of the evolved controllers. The four fitness functions are given by:

$$F = d_{\text{box}} - \frac{1}{2} d_{\text{box,robot}} \tag{38}$$

where d_{box} is the total distance that the box moved, and $d_{\text{box,robot}}$ is the final distance between the box and the robot;

$$f = \Delta d_{\text{box}} - \frac{1}{2} \left(\Delta d_{\text{box,robot}}\right) \tag{39}$$

where Δ (delta) indicates change over the current time step;

$$f = c_1(s_{ir2} + s_{ir3}) + \left(1 - c_2 \sum s_{\text{photo}}\right) \tag{40}$$

where s_{ir2} and s_{ir3} are the forward facing IR proximity sensors and $\sum s_{\text{photo}}$ is the combined activation of 4 photosensors;

$$f = c_1(s_{ir2} + s_{ir3}) + c_2 \left|v_l + v_r\right| - c_3 \left|v_l - v_r\right| \tag{41}$$

where v_l and v_r are the left and right wheel motor speeds.

All of these fitness functions converged upon a solution within 250 generations, using a controller population size of 30. The first and simplest function was reported as generating the best solutions. Only the controllers evolved with the first function were demonstrated in the real robot. Also, of the four fitness functions, the first contained the least *a priori* information about the task. This is interesting, and indicates that assumptions the researchers made about what features a good solution should have may not have actually been helpful in generating better solutions.

In [53], gaits were developed for a biped robot built from hobby servos. Embodied evolution was used to optimize a set of 12 gait parameters. The robot's vision and IR sensors generated information needed for fitness evaluation:

$$F = vD(\theta). \tag{42}$$

Here v is the average velocity of the robot over the trial period and $D(\theta)$ is a function that measures angular change in the robot's heading. D has a somewhat complex formulation but essentially rewards controllers that produce less rotation. F is used here to indicate that this is the complete fitness evaluation function, not an instantaneous integrand that is averaged or integrated over a trial period. This work started evolution with a working hand-formulated controller, and hence represents optimization rather than primary synthesis.

[54] investigated the evolution of three basic behaviors in the context of eventually learning more advanced behavioral coordination mechanisms. The behaviors were photo-orientation, object avoidance, and robot homing (come into proximity of another robot in the environment). Robots built mainly from LEGOs and equipped with IR, tactile and photosensors were used to implement a fully embodied evolution scheme. For each task, a simple tailored fitness function was used during evolution. The fitness functions are listed below:

$$F_{\text{phototaxis}} = \frac{\sum t_{\text{light}}}{t_{\max}} \tag{43}$$

where $\sum t_{\text{light}}$ is the number of time steps in which the robot was facing the light source, and $t_{\max}$ is the total number of time steps in the trial;

$$F_{\text{avoidance}} = \frac{\sum t_s}{t_{\max}} \tag{44}$$

where $\sum t_s$ is the number of time steps in which none of the IR sensors report activation; and

$$F_{\text{homing}} = t_{\max} - t_{\text{complete}} \tag{45}$$

where t_{complete} is the amount of time required to position the robot in proximity to another robot in the environment. The authors report successful evolution of behaviors within 30 generations for each of the three tasks. Unlike most of the other work reviewed, the genetic programming and controller structures used here contained some very high-level primitives, such as obstacle avoidance. It is unclear how much of the resulting control strategies were truly learned, and how much was encoded into the GP structure and learning environment.

In [12] a phototaxis task in which robots learn to home in on a light source was investigated. Evolution was performed in a population of eight real robots using an asynchronous algorithm and a tailored fitness function. By asynchronous we mean that the evolutionary algorithm did not employ any specific generation or epoch period. Rather, a controller might propagate as soon as it had achieved a high enough fitness level, regardless of what the other controllers were doing. During evolution, fitness was considered to be the current *energy level* of each robot and increased each time the robot reached the light source, and decreased during reproduction when a robot would broadcast its genes to other population members. The fitness function φ calculates the energy level and is updated at each time step. It can be summarized as follows:

$$\varphi(t) = \varphi(t-1) + c_1 B_{\text{reward}} - c_2 B_{\text{penalty}} \tag{46}$$

where t is time, B_{reward} is a Boolean that is true in any time step when the robot reaches the light source, and B_{penalty} is a Boolean that is true in any time step where the robot broadcasts its genes. The fitness is limited to a maximum value. The algorithm is fully asynchronous and there is no population-wide trigger for reproduction. Reproduction (broadcasting and receiving of genes) is governed probabilistically based on robot energy (fitness) levels. After about 100 min of evolution, fitness improvement in the population leveled off. Controllers capable of approaching the light source using a variety of strategies were reported.

In [56] embodied evolution was used to generate neural network controllers for locomotion and wall avoidance in a Koala robot equipped with a vision system. The authors used a simple fitness function integrand that maximized the forward velocities of the wheel motors:

$$f = (v_l + v_r) - (\left|v_l - v_r\right|) \tag{47}$$

where again v_l and v_r are left and right drive motor speeds. Fitness was integrated over the time steps of a trial, and then integrated again over a given number of trial periods. Note that unlike the works of [11,13], no explicit sensor activation term was used. This allowed the evolutionary process more freedom to evolve novel solutions. The underlying task investigated here is quite simple, and has been studied in many previous research efforts. In most previous work, though, IR and photosensors were used. Here, a grey-scale image from the vision system was partitioned into a 5 by 5 grid, and the average light level of each grid cell was used as a network input. Using a population of 40 neural controllers, fit controllers able to navigate around the simple environment were evolved within the physical environment in only 15 generations.

[30] discusses the evolution of a coordinated movement task in which three robots must move together in formation. The robots used a typical differential drive system and each had four IR sensors for detection of the environment. The authors use a relatively complex tailored fitness function given by

$$F = P \sum_{1}^{T\max} \left(D_{\text{gain}}(d, d_{\text{best}}) \left(1 + \tanh\left(\frac{S}{20}\right)\right)\right) \tag{48}$$

where P is a collision penalty term that decreases toward 0 as the number of collisions increases, D_{gain} is a function of present

Please cite this article in press as: A.L. Nelson, et al., Fitness functions in evolutionary robotics: A survey and analysis, Robotics and Autonomous Systems (2008), doi:10.1016/j.robot.2008.09.009

distance d and trial-best distance d_{best}, and S is a measure of team dispersion. In this case we included the summation term explicitly in the fitness function representation because the collision penalty factor P is applied after the integration over the evaluation time period of the other elements of fitness evaluation. The function is represented by an uppercase F, in accordance with the nomenclature used in this paper. The function is relatively selective for a class of *a priori* known solutions, but is not explicitly selective for an exact known solution. In [30] the authors used a population of 50 controllers and ran 100 separate evolutions. They report that in every evolutionary run, a fit controller eventually arose that was capable of achieving the group locomotion task in simulation, and also when tested in the three real robots.

[57] studied the evolution of locomotion and object avoidance behaviors using evolvable hardware controllers (FPGA). Embodied evolution was used and fitness evaluation was performed in a physical Khepera robot with FPGA turret. The following fitness function was used:

$$F = c\frac{d(1-\sum s)}{\sum rev} \tag{49}$$

where d is the distance traveled by the robot, $\sum s$ is the sum of all sensor activations, and $\sum rev$ counts the number of discrete motor direction reversals. This fitness function is related to the behavioral functions used for locomotion and object avoidance in earlier works [11,13,56], but here, distance traveled d (an aggregate term) is used, rather than wheel motor speeds. The authors also include the unusual behavioral term $\sum rev$ counting the number of motor reversals over the course of a trial period. Successful controllers were evolved within about 20 generations.

As part of a larger layered control architecture, in [58] the authors evolved fuzzy logic controller modules for object (ball) homing behaviors in a Sony AIBO. Here an evolvable fuzzy logic controller architecture was evolved for control. A tailored fitness function with three aggregate terms of the following form was used:

$$F = (1 - c_1 d)(1 - c_2 a_{\text{goal}})(1 - c_3 t_{\text{goal}}) \tag{50}$$

where d is the final distance between the target object (a ball) position, a_{goal} is the final angle between the target object position and the robot's head, and t_{goal} is the amount of time elapsed during the trial. c_1, c_2, and c_3 are chosen to normalize the various factors. Note that this fitness function implicitly includes a stopping condition when the target is found. Unlike many other ER experiments, this particular work included a fair amount of hand-coded control elements, including sensor fusion and object identification, and predefined gaits. In the fuzzy logic controllers, the antecedents of the rules were predefined, while the consequences of the rules were evolved, and this may have introduced a high level of additional *a priori* task knowledge into the resulting evolved controllers. The task of object homing was evolved in the real robot in 20 generations using a population of only 10 individuals.

Neural network controllers for object avoidance and locomotion were evolved using embodied evolution in a system of six real robots and also in a simulated version of the system [55]. The robots were custom-built differential drive systems with IR and tactile sensors. Fitness over a given trial period was calculated using the following set of rules: (1) Start with 4096 points; (2) Receive 10 points for each second of forward movement; (3) Lose 30 points for any occurrence of a forward motion command that is shorter than 15 s in duration; (4) Lose 10 points for each collision that occurs in conjunction with a forward motion command. In terms of selection, this set of rules produces an effect similar to some of the other behavioral and tailored fitness functions but it is unusual in that it includes actual motor commands explicitly. Element 2 makes the function tailored since it measures a degree of task completion. Elements 3 and 4 are behavioral terms. As in the case of [12] the evolutionary algorithm studied in [55] was intended to operate within the six physical robots. During each generation, the fittest robot controller was transferred to the other five robots where it was combined with the local controller using crossover. The authors explored various mutation rates, including a form of periodic macro-mutation couched in terms of predation. Robot controllers able to maximize the fitness function arose during the course of 200 generations in experiments run in the real robots.

[59] discusses the evolution of locomotion and object avoidance behaviors using an EvBot robot equipped with 5 binary tactile sensors. The EvBot robots [96] are small cylindrical robots between 15 and 25 cm in diameter, and can be equipped with a variety of sensors. These robots have more computing power than typical robots of this size and generally run full PC operating systems as well as high-level computing packages on board. Neural controllers were evolved to perform the navigation task, and a tailored fitness function was used:

$$F = c_1 d_{\text{net}} + c_2 d_{\max} + c_3 d_{\text{arc\_length}} - c_4 B_{\text{stuck}} \tag{51}$$

where d_{net} measures the offset distance between the robot's starting and its final position, $d_{\max}$ measures the greatest distance achieved by the robot at any time during the trial, $d_{\text{arc\_length}}$ is the line integral arc length of the robot's path over the course of a trial and B_{stuck} is a Boolean that is true if the robot becomes permanently immobilized during the trial. Because only five binary sensors were used to detect the environment, the robot's perceptual space contained only 32 distinct states, hence a simple reactive controller would be sub-optimal. To compensate for this, recurrent neural networks with several hidden layers and capable of temporal signal processing were used. The authors report that effective controllers were evolved in simulation and tested on real robots using a population size of 20 controllers and required on the order of 3000 generations. This number of generations is quite high compared to much of the other ER work surveyed, but at the same time, most other work resulted in the evolution of simple controllers that were purely reactive. The controllers in [59] evolved time-memory control solutions that compensated for the extreme temporal aliasing introduced by the binary tactical sensor system used.

Gaits for a biped servo-bot were evolved in simulation and then demonstrated on a real biped robot in [60]. Evolvable spline controllers were used in this work. Splines controlling each joint actuator were coordinated based on gait cycle time, and the evolutionary process altered each joint actuator spline's defining control point parameters. During evolution, a tailored fitness function that measured how far the robot moved was used:

$$F = c_1 d - c_2 v_{\text{body\_lowering}} \tag{52}$$

where d is the distance traveled by the robot and $v_{\text{body\_lowering}}$ is the average downward motion of the robot's torso. This term is included to reduce the amount of body movement of the robot to create a smoother gait. This fitness function is classified as tailored because it measures a degree of success of task completion, i.e. how far the robot moved, regardless of how that movement was achieved. The second term of the function is a behavioral term that selects for gaits with minimal movement in the torso. As with some, but not all, of the research into gait learning, in this work no sensors were used and the robots did not learn to dynamically interact with their environment. The evolved controllers were able to generate balancing and walking both in simulated and real biped robots.

Gaits for a Sony AIBO robot were evolved using embodied evolution in [61]. A gait parameter set was evolved. The fitness function used is very similar to the one used in [53] and derived all required information from images received from the robot's head-mounted camera:

$$F = vD(\theta). \tag{53}$$

Here v is the velocity of the robot over the trial period and $D\theta)$ is a function that measures change in the robot's heading. D rewards controllers that produce less rotation. Robots were reported to travel at speeds of 1 m/min using the best evolved set of gait parameters. Although images from the robot's camera were used during evolution to determine fitness, the robot did not learn to react to an object or other elements of its environment based on sensor information. As in many of the other gait-learning experiments, the learned control was not dynamic with respect to the environment. The evolutions required between 300 and 600 generations using a population of size 30. This number of generations is quite high compared to other embodied work.

In [62], controllers for locating and pushing objects toward a goal region were evolved in simulation and then optimized in real robots. A form of Q-learning was used, in addition to a genetic programming phase, for a hybrid system that cannot be purely labeled as evolutionary robotics. In order to accommodate the Q-learning phase of control learning, a state space was formulated based on a classification of images from the robot's camera. Further, the robot's possible actuator command set was similarly formulated into a finite set of states. A complicated tailored fitness function was used and is summarized by:

$$\begin{aligned} F = {} & c_1 B_{\text{goal}} + c_2\left(1 - \frac{moves}{moves_{\max}}\right) + c_3\left(1 - \frac{turns}{turns_{\max}}\right) \\ & + B_{\text{box\_moved}} + B_{\text{goal\_seen}} - \frac{goal\_lost}{t_{\max}} \end{aligned} \tag{54}$$

where B_{goal} is a Boolean function that is true if the robot moves the object to the goal area before the trial period is over, *moves* is the number of linear moves made by the robot, *turns* is the number of turning moves made by the robot, $B_{\text{box\_moved}}$ is a Boolean function that is true if the robot manages to move the object at all, $B_{\text{goal\_seen}}$ is true if the robot positively detects the goal region at any time during the trial, and *goal_lost* counts the number of times the robot turns away from the goal region after it has first detected it. The terms $\text{moves}_{\max}$, $\text{turns}_{\max}$ and $t_{\max}$ refer to maximum allowable numbers of moves, turns and time steps per trial period respectively. These three terms were set by the researchers based on their expertise and understanding of the experimental setup. In [62], the authors reported that successful controllers were generated in simulation and then tested and automatically optimized in real robots.

A relatively difficult sequential task in which a robot must visit three goal locations in a specific order is investigated in [22]. Populations of small recurrent neural networks (6–10 neurons) were evolved with an extended multi-population genetic algorithm. Most other ER researchers use single population genetic algorithms, and the work in [22] compared the two forms of evolution and reported superior results using multi-population evolution. During a given trial period, fitness is updated by:

$$\varphi_{\text{goal\_position}} = \begin{cases} 1 & \text{if in sequence} \\ -1 & \text{otherwise.} \end{cases} \tag{55}$$

Here fitness is updated only when the robot reaches any one of the goal locations. An additional single overriding behavioral condition is also included: a robot that produced only spinning-in-place behaviors is given -30 points. The main fitness function contains relatively little *a priori* task solution knowledge and uses only information related to the completion of the cycle of goal visitations. Successful controllers were evolved after 50 generations using populations of on the order of 200 individuals. This is well within the range of generations required for evolution reported by other researchers for other less complex tasks.

Controllers capable of performing a phototaxis task in an environment with many occluding obstacles were evolved in [63]. A robot constructed from LEGO Mindstorm kits and equipped with two photosensors and a single forward-mounted tactile sensor was used. A simple tailored fitness function was employed, and is given by:

$$F = d_{\max}^2 - d^2 \tag{56}$$

where $d_{\max}$ is the largest dimension of the training environment, and d is the final distance between the robot and the light source after an evaluation trial period. The authors report successful evolution of controller programs after 350 generations and using a population size of 64. Control programs were evolved in simulation and then tested in the real robot. The best evolved control program was reported to be able to complete its task (finding a light source) in all 15 trials in the real robot.

In recent work [110] neural network-based controllers were evolved to perform a coordinated group movement task in which interconnected robots travel together in an environment containing holes. An aspect of robot–robot communication was involved in this work, and robots were given the ability to produce and receive signals in the form of a tone. To perform the task, the robots (arranged in a square) must move in as straight a line as possible while maintaining formation and avoiding holes. Evolution of controllers was performed in simulation and then evolved controllers were tested in physical robots. A complex tailored fitness function that combined individual robot fitnesses and success at hole avoidance was used. The individual robot fitness was measured using a function similar to Eq. (3) but with additional tailored factors. This function included elements that selected for rapid movement using a normalized version of the first factor of Eq. (3), straight movement using the second factor of Eq. (3) with a zero-floor condition, as well as a factor that measured the degree of coordinated traction force produced by the robots. The function contained additional factors that measured the degree of ground sensor activation (used to detect the holes) and a term intended to minimize robot–robot communication. The fitness of the robot group was 0 if any member of the group fell into a hole, and a function of the lowest individual robot fitness otherwise. In addition to the explicit fitness function used in this work, three separate environments, two of which did not contain holes were used during evolution. Although each fitness evaluation trial combined results from all three environments, in some respects, this represents a form of environmental incremental evolution (discussed in Section 3.5) since robots could realize fitness gains early in evolution by improving simple locomotion abilities in the environments without holes.

3.5. Environmental incremental fitness functions

Environmental incremental evolution (research listed in Table 6) differs from functional incremental evolution in that the difficulty of the environment is augmented, and not the fitness function. Potentially, this can produce controllers evolved with aggregate selection and less explicit human bias. Some degree of human bias is still injected into the evolving controllers due to the selection of the various incrementally more challenging environments.

In [37] a goal homing and object avoidance behavior was evolved using a robot equipped with IR sensors and binary photosensors. Neural network controllers for the robot were evolved in two sequentially more difficult environments. The first environment included a single simple wall-like obstacle, while the second contained a concave cul-de-sac obstacle that required the

Please cite this article in press as: A.L. Nelson, et al., Fitness functions in evolutionary robotics: A survey and analysis, Robotics and Autonomous Systems (2008), doi:10.1016/j.robot.2008.09.009

robot to learn to initially backtrack away from the goal in order to eventually approach it. The fitness function integrand used in both environments takes the following form:

$$f = (D_{\text{goal}}) \vee_{\text{xor}} \left(|\text{mean}(v_l, v_r)| \, (1 - \sqrt{|v_l - v_r|})(1 - s_{ir}) \right) \quad (57)$$

where v_l and v_r are left and right drive motor speeds, and s_{ir} is the greatest current activation level of the IR sensors. D_{goal} is a measure of proximity to the goal. Note that the two parts of the fitness function are mutually exclusive; only one can apply at a time. 200 generations were performed in the simpler environments, followed by 200 additional generations in the more complex environment. Populations of 100 network-based controllers were used in this work, and the authors compared recurrent and non-recurrent neural network architectures in separate experiments. The recurrent networks were able to achieve higher levels of performance (they reached the goals more quickly), but the non-recurrent networks were still able to learn (i.e. evolve) to perform the task.

Environmental incremental evolution is also studied in [35]. The authors investigated a fairly complex object acquisition and deposition task in which a simulated robot equipped with a gripper must find and pick up a peg in an arena and then deposit it outside the border of the arena. A single fitness function was used while the conditions of the environment were changed incrementally to produce an increasingly difficult task. The fitness function used is given by

$$F = t_{\max} - t_{\text{finish}} \quad (58)$$

where $t_{\max}$ is the maximum allowable time per evaluation trial, and t_{finish} is the time at which the task is completed. Note the function F in (58) provides the final fitness evaluation for a given trial and is not an integrand. Controllers were evolved in three stages, each with an incrementally more difficult environment. These were: (1) In the first stage, robots began each trial already holding the object, so the task would be completed when the robot had moved to the edge of the arena and dropped the object; (2) In the second stage, robots began each trial with the object directly in front of their grippers; (3) In the final stage, the robots began each trial at a random position within the arena.

Note that F in (58) can be classified as an aggregate fitness function and injects no *a priori* information into the evolved solution. However, the selection of incrementally more difficult training environments does restrict the evolved solution to a degree, and the selection of these training environments required knowledge of a feasible solution by the designers. The three environments listed above were used for 100, 400, and 100 generations respectively, and the best evolved neural controllers were able to perform the overall task when tested in the simulated robot.

3.6. Competitive fitness evaluation

Competitive and co-competitive evolution in which the fitness of one individual may directly affect the fitness evaluation of another individual represents an important but relatively small subset of the research surveyed in this paper. Several examples of co-competitive evolution, in which two distinct populations compete against each other asymmetrically (e.g. predator and prey robots), have been reported in the literature (research listed in Table 7).

The research presented in [10] co-evolved pursuit and evasion behaviors in differently configured species of predator and prey robots. The authors used competitive aggregate fitness selection methods. The fitness functions for the competing robot species were based on the time at which contact between predator and prey occurred and can be summarized as follows:

$$F_{\text{prey}} = \frac{t}{t_{\max}} \quad (59)$$

$$F_{\text{predator}} = 1 - \frac{t}{t_{\max}} \quad (60)$$

where t is the time at which contact occurred and $t_{\max}$ is the maximum length of time allowed for the evaluation trial periods. These paired fitness functions are considered aggregate because they involve only information pertaining to completion of the task, and not information related to low-level behaviors. As is often the case with aggregate fitness functions, F_{prey} and F_{predator} generate weak (but unbiased) fitness signals. To compensate for this, fitness for the evolving individuals was averaged over several complete sets of trials before selection occurred. Two populations of 100 controllers each were evolved (one for the prey and one for the predator) for 100 generations. The resulting populations of controllers achieve an initial level of competence early in evolution (after 25 generations), and then begin to cycle through reciprocal levels of higher and lower performance. The authors show that this performance cycling can be lessened to a degree by using hall-of-fame selection rather than a simple greedy selection method.

Competitive evolution of neural network-based controllers was investigated in [21] using EvBot robots equipped with color vision systems. Teams of robots competed against each other to find separate goal objects placed within a complex maze environment. The work was described in terms of a competitive game in which each robot team must try locate their opponent's goal or home marker object before the other team can locate theirs. During each generation, a tournament of games was played between teams of robots, in which the robots on one team would be controlled by copies of one controller network from the evolving population and the robots on the other team would be controlled by copies of another network from the same population. If any single controller in the population was able to win a game, then all controllers in the entire population were evaluated using the following aggregate fitness function:

$$F = 1.5wins - .5draws - 1losses \quad (61)$$

where *wins* is the number of wins achieved by the controller during a tournament, *draws* is the number of games played to a draw, and *losses* is the number of games lost by the controller. If no single controller within the entire population could win a game, then fitness selection reverted to a tailored bootstrap selection mode summarized by:

$$F = d_{\max} - c_1 B_{\text{stuck}} - c_2 B_{\text{motor}} \quad (62)$$

where $d_{\max}$ is the maximum distance traveled by the robot, B_{stuck} is a Boolean that is true if all robots on a team become immobilized, and B_{motor} is a Boolean that is true if robots on a team generate motor commands that exceed the capabilities of the drive motors. Populations of 40 controllers were evolved for 450 generations. The best evolved controllers were tested in real robots and shown to be able to compete with hand-designed controllers created to play the same competitive game.

3.7. Aggregate fitness functions

Unlike behavioral fitness functions, which measure only aspects of a robot's behavior during testing, and tailored fitness functions, which may contain behavioral terms, aggregate fitness functions measure only task completion divorced from any specific sensor-actuator behaviors. Aggregate fitness functions collect (or aggregate) the benefit (or deficit) of all aspects of a robot controller's expressed abilities into a single term. The fitness of an evolving controller is calculated based only on whether or not it completes the task it is being evolved to perform. If the task can be completed in a well-defined way, the fitness function will use only success/failure information. For example, if the task involves competing in a win-lose game, the fitness function will include a Boolean whose value depends only on whether the game was won or lost in a particular trial period. If the task can be measured by a final achieved quantity, then it will consist of a single scalar term. For instance, in a foraging task, an aggregate fitness function might simply count the number of objects correctly collected at the end of a trial period. Research using aggregate fitness functions is summarized in Table 8.

Using aggregate fitness functions, controllers have been evolved for tasks including gait evolution in legged robots [64,32,69,70], flying lift generation in a flying robot [66], and simpler locomotion [16,65,67,68] and object pushing [17] tasks. The simpler actuator coordination tasks are less environmentally situated and produce less complex reactions to environmental stimuli. In some cases, the robots do not have sensors at all. However, these works are included here for their application of evolutionary computation to design novel controllers.

In [17] the evolution of a ball-pushing behavior using a Sony AIBO is described. The function measures the degree of success of moving the ball simply by measuring the linear distance between the ball's starting and ending locations:

$$F = d_{ball}. \quad (63)$$

Here, fitness for each individual was averaged over several evaluation trials before propagation of the population to the next generation took place. This reflects an attempt to boost the limited fitness signal generated by this aggregate fitness function. In this work, information from vision and IR sensors was fused to generate virtual sensor information including angle to ball and apparent size of ball. Populations of 60 neural network controllers were evolved in simulation for 100 generations. The fittest controllers were then demonstrated in a real robot.

In [64] the authors use embodied evolution to develop gaits for a hexapod robot. The gait controllers were in the form of evolvable state lookup tables. An aggregate fitness function was used that measured the distance traveled by the robot while walking on a treadmill:

$$F = d. \quad (64)$$

The researchers reported evolution of functional gaits after 23 generations in a real robot using a controller population of size 30.

Both [16,65] describe separate examples of systems in which whole robots (bodies and controllers) were co-evolved in simulation and then constructed in the real world using modular actuators and structural units. In both cases robots were evolved for locomotion abilities and fitness was calculated simply as the distance d traveled. This was a completely aggregate fitness function and contained no other features of potential control solutions or of possible robot morphologies:

$$F = d. \quad (65)$$

These two separate research efforts evolved agents capable of locomotion, but without any sensors, and thus the evolved control structures did not interact dynamically with their environments to actively avoid obstacles or perform other tasks that might be expected of an autonomous robot. However, both systems produced functional designs that were used to construct real robots that were tested and found to be able to move in the real world.

[32] reported on the embodied evolution of a locomotion behavior in which a robot (built from a LEGO Mindstorm kit and relying on tactile sensors for object detection) must travel around an environment containing a single circuit and small obstacles placed along the walls. The controller architecture used here was a simple mapping from sensor activation states to motor states. Although not referred to as such in the paper, this was in essence a simple neural network with linear excitation functions and constant weights. The fitness function simply measured the distance traveled by the robot (as reported by a swivel wheel and odometer attached to the robot) over a given evaluation period:

$$F = d_{\text{arc\_length}} \quad (66)$$

where $d_{\text{arc\_length}}$ is the line integral (arc length) of the path traveled by the robot, rather than the net displacement. This fitness function can be considered aggregate if the task is considered to be simple locomotion in an unknown environment. After 20 generations, the best evolved controller was able to produce a continued locomotion speed of 3.5 m/s without colliding with obstacles. This compares to a slightly slower speed achieved by a hand-coded controller designed to perform the same task.

In [66] embodied evolution is used to develop lift-generating motions in a winged flying robot. A simple fitness function was used that measured height obtained by the robot at each time step:

$$f = h. \quad (67)$$

A simple genetic programming-based controller representation capable of expressing wing angles, positions and time durations was used. A very small population of 4 individuals was evolved with a tournament selection genetic algorithm. The robot did learn to generate lift, but the robot was unable to generate sufficient lift to completely loft the robot under its own power. Hence, the evolution process was not entirely successful.

In [67] an indoor floating robotic blimp equipped with a camera and placed in a small room with bar-code-like markings on the walls was evolved to produce motion and wall avoidance. Populations of neural network-based controllers were evolved. A fitness function was used that averaged magnitude of velocity over each trial period:

$$f = v \quad (68)$$

where v is the current velocity of the robotic blimp at each time step. The function is considered aggregate when selecting for the task of movement. The aspect of object avoidance is included only implicitly. Using a population size of 60, successful controllers were evolved in 20 generations with evaluations performed on the physical robot.

[68] described the evolution of morphology and recurrent neural network-based control for sensorless modular robots constructed of LEGO and servo units. Robots were evolved for locomotion abilities in simulation and then constructed in the lab with real hardware. An aggregate fitness function (the same as that used in [16,65]) was used that measured total net locomotion distance d over the course of a trial period:

$$F = d. \quad (69)$$

Note that an initial settling period occurred before each fitness-measuring period began. This was done to avoid selecting for robots that moved merely by falling and this makes the fitness function technically tailored, to a small degree. Evolution of functional locomotion abilities required on the order of 2000 generations. The best evolved robot and controller was constructed and was reported to be able to move approximately 14 cm per minute. In its virtual evolution environment, the simulated version was able to move about twice this distance in the same number of controller update cycles.

[69] presents another example of the embodied evolution of gaits in a physical robot. The robot was a pneumatic hexapod of

minimalist design. The authors used the same aggregate fitness function as did [16,64,65,68]. Distance for the aggregate fitness function was determined using images taken from an overhead camera. The evolvable controller structure consisted of a set of gait parameters within a control program looping structure.

Gait learning using a slightly different aggregate fitness function is given in [70]. In this paper, a Sony AIBO quadruped robot was used, and the evolutionary process was embodied in the real robot. Evolved controllers took the form of a set of 12 gait parameters. Fitness was measured as average speed achieved by the robot:

$$F = \frac{d}{t_{\max}} \tag{70}$$

where $t_{\max}$ is the time length of a given evaluation trial. Note that since $t_{\max}$ is constant, this function reduced to that used in [64,69]. The gaits evolved in [70] were reported to allow the robots to travel 20% faster than the best gaits resulting from hand-tuned parameter sets.

3.8. Research using simulated robots

In preceding sections of this survey we have concentrated on work that involved real robots in one form or another. ER work requires some level of physical verification. Research done with agents in a purely simulated environment with no association to any particular physical system is often classified as artificial life research. Even so, there is a quite large body of work billed as evolutionary robotics research that uses only simulated robots, or animats [103,104]. Simulation plays a major role in evolutionary robotics, and the case can be made that simulation-only work can be as valid as work verified in real robots, if the simulations are designed carefully. The work done in the 1990's demonstrated the validity of many simulated systems by directly verifying results in real robots. At the same time artificial life research and some work billed as evolutionary robotics continues to make use of simulated sensors and actuators that incorporate global environmental knowledge, sometimes in very subtle ways. The great majority of the ER research reviewed above did involve real robots, primarily in the verification phases. In only a very few cases, simulation-only work was listed because it was involved with a project that used real robots or because it was the only example of work of its type.

In this subsection we briefly list some of the major simulation-only ER research reports, starting with those that used behavioral or tailored fitness functions.

[72] reports on a cellular encoding scheme for evolvable modular neural networks for simulated legged robot control. An example of a relatively complex task achieved in simulation using a tailored fitness function is presented in [73]. The authors describe the evolution of a coordinated movement task involving several simulated robots. In [74] the authors studied the evolution of simulated robot controllers for a task in which a robot must collide with objects ("collect" them) in one zone and avoid them in another. Another example of evolving simulated robot controllers to perform a (relatively) complex task is reported in [75]. There, robot controllers evolve to produce lifetime learning in order to predict the location of a goal object based on the position of a light source. [76] presents experiments to evolve a group-robot flocking behavior in simulated robots. A simulated two-robot coordination task in which two robots evolve to move while maintaining mutual proximity is reported in [77]. In addition, the research in [78] evolved homogeneous controllers for a task in which four simulated robots must move together in a small group toward a light or sound source. In [79] groups of simulated robots (the Swarm-bots) evolve group attachment and aggregation abilities as well as group locomotion and inverted obstacle (hole) avoidance abilities. These robots have since been built [80], and [110] reports recent tests of the evolved controllers in the real robots. Other examples of behavioral and tailored fitness functions used for the evolution of behaviors in simulated robots are found in [81].

In [82] controllers created using incremental evolution in simulation are studied. In [83] the authors study functional and environmental incremental evolution and multi-objective optimization in a simulated aerial vehicle. Further application of multi-objective optimization applied in simulated evolutionary robotics systems is found in [8] as well as an extensive review of related work. [84,39] investigated the simulated co-competitive evolution of competing populations in the form of predator–prey behaviors. Finally, the co-evolution of controllers and morphologies is studied in simulation in several works including [85,51,38].

4. Discussion

The literature contains a large amount of experimental repetition, at least in terms of tasks studied and fitness functions used. Looking through the entire body of evolutionary robotics work involving real robots, we find that there are only a handful of distinct tasks for which controllers have been successfully evolved. The most common among these are: (1) simple locomotion and basic actuator control for locomotion [16,42,53,56,60,61,64–70]; (2) locomotion with object avoidance [48,13,11,26,27,15,32,43,46,50,55,57,59,71,110]; (3) goal or position homing [12,34,58]; and (4) goal homing with object avoidance [29,36,37,45,47,63]. These tasks can be considered as a set of benchmark experiments for the field of ER.

In some ways, a set of *de facto* benchmark tasks does not benefit the field of ER as much as would be the case in many other fields. The emphasis in much of the current ER research is on evolving more complex behaviors [14,20–22], and ultimately more general behaviors.

4.1. Fitness assessment methods and novel task learning

In the larger field of evolutionary computing, honing of methods and optimization of algorithms play a central role. Many of the researchers who conduct experiments in the field of ER, and whose work has been discussed in this survey, also address issues of algorithm efficiency and evolutionary conditions [5,29,43,55]. Reducing the time needed for evolution [5] and minimizing controller size [43] are examples of particular aspects of evolutionary system efficiency that have been addressed in ER experiments. However, because ER is largely focused on developing novel behaviors not seen in earlier work, and on increasing the complexity of evolvable behaviors, efficiency issues take a back seat to the more fundamental issue of fitness assessment. It is necessary to use controller representations that do not significantly restrict the controller search space and that do not result in intractable evolutionary conditions, but beyond this, efficiency is not the limiting factor in the current state of the art of ER. The fitness function governs: (1) the functional properties of the evolved controllers; (2) the point at which training plateaus due to lack of fitness signal; and (3) the degree to which novel behavior is learned. Robot controllers capable of performing new, more complex, tasks are evolved when researchers devise new fitness functions capable of selecting for the particular tasks of interest. Current state of the art ER research employs fitness functions that can select for relatively simple controllers capable of performing tasks composed of no more than three or four coordinated components. (Tasks investigated are summarized in the introduction to this section.)

One might then inquire as to which classes of fitness functions are most effective at selecting for more complex behaviors. Some work has been done comparing the effectiveness of different fitness functions aimed at evolving controllers for the same task. [52] investigated four different fitness functions applied to a box pushing task and found that all of the fitness functions produced reasonably competent controllers. In addition, research in [21] compared environmental incremental evolution to standard single-environment evolution and found no clear advantage to either form of selection. Neither of these research efforts produced truly statistically significant results that might be generally applicable to a wider range of ER applications.

In general the relative fitnesses or qualities of the evolved controllers in different research efforts are not known. Because of this, results achieved in different research efforts using different fitness functions are difficult to compare in absolute terms. Even so, some comparison can be made. All else being equal, if two ER platforms produce robot controllers capable of performing similar tasks, with one platform using a complex hand-formulated fitness function and the other using an aggregate fitness function, then one can say that the platform using the aggregate fitness function is extracting more information from the environment during evolution. From this point of view aggregate fitness functions generate a greater degree of novel environment-based learning.

4.2. Can aggregate selection generate complex behavior?

In the early 1990's the first successful ER experiments generally employed behavioral and tailored fitness functions. Since that time, many varied forms of fitness functions have been applied by different researchers to evolve controllers.

For each of the common tasks that have become benchmarks in ER, there are examples of aggregate fitness functions or fitness function contains relatively little *a priori* task solution knowledge that have been applied to successfully evolve functional controllers for real robots [16,65,32,68,63]. The majority of such work has been accomplished since the year 2000 and represents a shift informed by experiment within the field of ER [107].

The existence of examples of aggregate selection being used to drive the evolution of controllers for a variety of tasks indicates that many of the arguments presented in justification of the various more complex fitness functions are not in fact completely sound, at least for these simple tasks. Some of the most complex behaviors evolved did require very complex fitness functions [14,110], but interestingly, several of these most complex evolved behaviors were also evolved in other experiments using fitness functions containing relatively little *a priori* task solution information [20,35,21,22].

Aggregate selection bases selection only on success or failure to complete the task for which controllers are being evolved. The other forms of fitness functions currently used in ER require designers to understand important features of the tasks for which robots controllers are being evolved. In extreme cases, the fitness function essentially defines an *a priori* known solution. If aggregate selection could be achieved for much more complex tasks, it could eventually lead to the application of ER methods to environments and tasks in which humans lack sufficient knowledge to derive adequate controllers.

In order to apply aggregate fitness selection methods, the *bootstrap problem*, in which randomly initialized populations have no detectable level of fitness, must be addressed. Sub-minimally competent initial populations cannot be evolved using only aggregate fitness functions. There are several methods that might be used to overcome this difficulty and still allow for the use of aggregate fitness selection. Some such methods have been applied in research reviewed in this paper. These include: (1) applying environmental incremental evolution in conjunction with aggregate selection [35]; (2) using a bootstrap mode that gives way to aggregate selection later in evolution; and (3) applying competitive evolution so as to create an environment that continually increases in difficulty due to the evolving skills of other controllers in the population or agents in that same environment [21].

Although it may be possible to overcome some of the problems with simple aggregate selection, ER research still has not generated controllers capable of performing truly complex tasks. The great majority of tasks investigated in current and past ER research are simple enough so that un-modified aggregate selection does work. However, the generalization of current ER methods to evolve controllers capable of performing more difficult tasks may require the development of new approaches to fitness evaluation.

4.3. Co-evolution of controller and morphology

In several evolutionary robotics papers, it has been suggested that the simultaneous evolution of morphology and control provides a pathway toward the development of complex robots expressing complex behaviors, and this holds some promise. However, the underlying argument that the co-evolution of body and mind is the only way to generate a complex controller fitted to a complex body is not entirely supported by the literature or by observation. Humans are much better at designing physical systems than they are at designing intelligent control systems: complex powered machinery has been in existence for over 150 years, whereas it is safe to say that no truly intelligent autonomous machine has ever been built by a human. There is also genetic evidence from genome analysis that indicates natural evolution of cognitive function is evolving more rapidly than anatomical structure in modern humans [86]. This implies that there may be a lag in the evolution of intelligence in humans compared to physical development.

The co-evolution of bodies and minds does have potential though, and several recent works have overcome previous barriers by combining high-fidelity simulation environments with modular elemental component libraries [16,65,68]. These newer co-evolution works have used aggregate or very low bias selection methods, and complex physical robots were fabricated and tested in the real world.

4.4. The role of survival and fitness assessment in simulation

If fitness could be implemented simply as the ability to propagate within a complex environment, it is possible that systems with novel integrated behavioral intelligences could be evolved. In order to achieve this, the fundamental element of survival must be formulated in a way that is consistent with the actual physical representation of the environment. Specifically, robots should fail to survive when they physically stop functioning. It is not currently feasible to perform embodied evolutions of this type because many robots would be damaged and destroyed during evolution. For example, a population of 50 robots evolved for 100 generations with 50% failure during each generation would generate 2500 fatal robot failures. At the present time, the only alternative is simulation.

The field of artificial life (AL) studies this concept of evolution based on survival in simulated environments, but AL research very often uses artificial measures of survival (essentially objective functions), such as the ability to gather food or energy icons. These measures of survival are not consistent with the underlying representations of the simulated agents, or the simulated physics of the environments.

Simulations used in evolutionary robotics attempt to consistently integrate all necessary physical elements of the robot's environment. For example, in the peg collecting behavior evolved in [14], the pegs were simulated as physical objects, fully integrated into the object representation of the simulated environment. The fidelity of the simulation was verified by testing evolved controllers in the real world. Here, though, fitness was defined in terms of success at collecting pegs, rather than physical survival of the robots. In the competitive goal homing task studied in [21], robots, obstacles and target objects were integrated into the simulation environment using representations that were consistent with a physical environment. However, survival, enforced by the fitness function, was couched in terms of robots physically arriving at the goal locations in the environment, and not in terms of actual physical survival of the robot. Although the experiments discussed in [14,21] produced robots capable of performing their intended tasks, neither of these research efforts produced robots capable of physical survival in a complex environment *per se*.

In order for autonomous robot simulation environments to be useful in studying real evolutionary processes, and in evolving true survival abilities in robots above and beyond performing simple well-defined tasks, they must include measures of survival that are truly integrated into the fabric of the simulation environment, while at the same time being consistent with real robotic systems. Evolutionary robotics has not reached this stage of generality, and this will be an important area of research. Survival in complex environments is a fundamental goal in evolutionary robotics and autonomous intelligent robotics as a whole.

In many ways the concept of fitness to perform a specific task is at odds with fundamental survival in a complex environment. If adaptation to environmental conditions is to be studied in the general case, specific tasks should not be imposed upon the evolving systems. Developmental robotics [108,109] addresses the problem of environmental-based learning by attempting to endow robots with environmentally stimulated self-organizing abilities or environmentally stimulated novelty-seeking. This eliminates the need for an explicit objective function to drive the learning process, but does not necessarily result in the learning of any one particular behavior, beyond negotiating a particular environment.

4.5. The long-term prospects for evolving intelligence

Is it possible to automatically generate controllers for arbitrarily difficult and complex tasks in the general case? The short answer to this question is probably no. The tasks currently investigated by evolutionary robotics are relatively simple compared to natural systems and shed little light on the question of learning arbitrarily difficult tasks. Even natural intelligent systems are limited in their degree of sophistication, at least as represented by life forms observed on Earth. To make matters worse, the overall abilities of complex animals and humans are difficult to describe, much less duplicate in a holistic way.

Let us put some bounds on this general question of using artificial evolution to generating controllers for arbitrarily difficult tasks: is it possible to use artificial evolution to generate intelligent systems capable of solving large classes of tasks in the realm of intelligent autonomous systems? The answer to this question is unknown, but current evolutionary robotics results indicate that it may be possible to generate autonomous systems with limited general abilities at some point in the future.

Current ER research has demonstrated that competent autonomous environmentally situated agents can be evolved. The most complex evolved robot systems are capable of achieving three or four interconnected abilities in a coordinated fashion that allows an overall task to be completed. These results include the multiple sequential goal homing task [22], the object collection and deposition task [14], and the competitive team searching task [21], all discussed in the survey section of this paper. These systems were generated based on feedback from environmental interaction and represent a step toward more general systems.

5. Conclusion

In this paper we have reviewed the use of fitness functions in the field of evolutionary robotics. Many representative works were selected from the literature and the evolutionary processes used were summarized in terms of the fitness functions. Functions were reported using a standardized nomenclature to aid in comparison. It was found that much of the research made use of fitness functions that were selective for solutions that the researchers had envisioned before the initiation of the evolutionary processes. The degree to which features of evolved solutions reflected *a priori* knowledge on the parts of the human researchers varied. A portion of the reviewed research reported the evolution of controllers which demonstrated novel abilities not specifically defined by the fitness functions used during evolution. Recent work done in the last few years has begun to use aggregate fitness selection methods that introduce much less human knowledge and bias into the evolved controllers.

The fundamental question of how to select for truly complex intelligent autonomous behaviors during evolution remains largely unanswered and evolutionary robotics remains somewhat on the fringes of autonomous robotics research. It should be pointed out that other non-evolutionary computing-based attempts at producing robots capable of learning novel intelligent control have, in general, stumbled up against the same problem of how to drive the learning process without introducing an essentially *a priori* known control strategy into the learned systems.

Acknowledgements

The authors would like to thank Brenae Bailey for editorial input, insightful conversations, and support in preparing this manuscript.

References

[1] M. Mataric, D. Cliff, Challenges in evolving controllers for physical robots, Robotics and Autonomous Systems 19 (1) (1996) 67–83.
[2] I. Harvey, P. Husbands, D. Cliff, A. Thompson, N. Jakobi, Evolutionary robotics: The Sussex approach, Robotics and Autonomous Systems 20 (2–4) (1997) 205–224.
[3] L.A. Meeden, D. Kumar, Trends in evolutionary robotics, in: L.C. Jain, T. Fukuda (Eds.), Soft Computing for Intelligent Robotic Systems, Physica-Verlag, New York, NY, 1998, pp. 215–233.
[4] S. Nolfi, D. Floreano, Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines, The MIT Press, Cambridge, Massachusetts, 2000.
[5] J. Walker, Simon Garrett, M. Wilson, Evolving controllers for real robots: A survey of the literature, Adaptive Behavior 11 (3) (2003) 179–203.
[6] D.K. Pratihar, Evolutionary robotics—A review, Sadhana - Academy Proceedings in Engineering Sciences 28 (6) (2003) 999–1011.
[7] K.C. Tan, L.F. Wang, T.H. Lee, P. Vadakkepat, Evolvable hardware in evolutionary robotics, Autonomous Robots 16 (1) (2004) 5–21.
[8] J. Teo, H.A. Abbass, Multiobjectivity and complexity in embodied cognition, IEEE Transactions on Evolutionary Computation 9 (4) (2005) 337–360.
[9] I. Harvey, P. Husbands, D. Cliff, Seeing the light: Artificial evolution, real vision, in: D. Cliff, P. Husbands, J.-A. Meyer, S. Wilson (Eds.), From Animals to Animates 3, Proc. of 3rd Intl. Conf. on Simulation of Adaptive Behavior, SAB94, MIT Press/Bradford Books, Boston, MA, 1994, pp. 392–401.
[10] S. Nolfi, D. Floreano, Co-evolving predator and prey robots: Do 'arms races' arise in artificial evolution? Artificial Life 4 (4) (1998) 311–335.
[11] D. Floreano, F. Mondada, Evolution of homing navigation in a real mobile robot, IEEE Transactions on Systems, Man, Cybernetics Part B: Cybernetics 26 (3) (1996) 396–407.
[12] R.A. Watson, S.G. Ficici, J.B. Pollack, Embodied evolution: Distributing an evolutionary algorithm in a population of robots, Robotics and Autonomous Systems 39 (1) (2002) 1–18.

[13] H.H. Lund, O. Miglino, From simulated to real robots, in: Proceedings of IEEE International Conference on Evolutionary Computation, 1996, pp. 362–365.
[14] S. Nolfi, Evolving non-trivial behaviors on real robots, Robotics and Autonomous Systems 22 (3–4) (1997) 187–198.
[15] P. Nordin, W. Banzhaf, M. Brameier, Evolution of a world model for a miniature robot using genetic programming, Robotics and Autonomous Systems 25 (1–2) (1998) 105–116.
[16] H. Lipson, J.B. Pollack, Automatic design and manufacture of robotic lifeforms, Nature 406 (6799) (2000) 974–978.
[17] G.S. Hornby, S. Takamura, J. Yokono, O. Hanagata, M. Fujita, J. Pollack, Evolution of controllers from a high-level simulator to a high dof robot, in: J. Miller (Ed.), Evolvable Systems: From Biology to Hardware; Proceedings of the Third International Conference, ICES 2000, in: Lecture Notes in Computer Science, vol. 1801, Springer, 2000, pp. 80–89.
[18] A.L. Nelson, E. Grant, G.J. Barlow, T.C. Henderson, A colony of robots using vision sensing and evolved neural controllers, in: Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS03, Las Vegas NV, 27–31 Oct., 2003, pp. 2273–2278.
[19] F. Gomez, R. Miikkulainen, Transfer of neuroevolved controllers in unstable domains, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-04, Seattle, WA, 2004, pp. 957–968.
[20] D. Floreano, J. Urzelai, Evolutionary robots with on-line self-organization and behavioral fitness, Neural Networks 13 (4–5) (2000) 431–443.
[21] A.L. Nelson, E. Grant, Using direct competition to select for competent controllers in evolutionary robotics, Robotics and Autonomous Systems 54 (10) (2006) 840–857.
[22] G. Capi, K. Doya, Evolution of recurrent neural controllers using an extended parallel genetic algorithm, Robotics and Autonomous Systems 52 (2–3) (2005) 148–159.
[23] I. Cloete, J.M. Zurada (Eds.), Knowledge-Based Neurocomputing, The MIT Press, Cambridge, MA, London, England, ISBN: 0-262-03274-0, 2000.
[24] C. Dima, M. Herbert, A. Stentz, Enabling learning from large datasets: Applying active learning to mobile robotics, in: Proceedings of the 2004 IEEE International Conference on Robotics and Automation, ICRA, New Orleans, LA, 2004, pp. 108–114.
[25] A.L. Nelson, E. Grant, G. Lee, Using genetic algorithms to capture behavioral traits exhibited by knowledge based robot agents, in: Proceedings of the ISCA 15th International Conference: Computer Applications in Industry and Engineering, CAINE-2002, San Diego, CA, 7–9 Nov., 2002, pp. 92–97.
[26] W. Banzhaf, P. Nordin, M. Olmer, Generating adaptive behavior using function regression within genetic programming and a real robot, in: Proceedings of the Second International Conference on Genetic Programming, San Francisco, 1997, pp. 35–43.
[27] N. Jakobi, Running across the reality gap: Octopod locomotion evolved in a minimal simulation, in: P. Husbands, J.A. Meyer (Eds.), Evolutionary Robotics: First European Workshop, EvoRobot98, Springer-Verlag, 1998, pp. 39–58.
[28] A.C. Schultz, J.J. Grefenstette, W. Adams, RoboShepherd: Learning a complex behavior, Robotics and Manufacturing: Recent Trends in Research and Applications 6 (1996) 763–768.
[29] D. Keymeulen, M. Iwata, Y. Kuniyoshi, T. Higuchi, Online evolution for a self-adapting robotic navigation system using evolvable hardware, Artificial Life 4 (4) (1998) 359–393.
[30] M. Quinn, L. Smith, G. Mayley, P. Husbands, Evolving team behaviour for real robots, in: EPSRC/BBSRC International Workshop on Biologically-Inspired Robotics: The Legacy of W. Grey Walter, WGW'02, 14–16 Aug., 2002, HP Bristol Labs, UK.
[31] K. Kawai, A. Ishiguro, P. Eggenberger, Incremental evolution of neurocontrollers with a diffusion-reaction mechanism of neuromodulators, in: Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'01, vol. 4, Maui, HI, Oct. 29–Nov. 3, 2001, pp. 2384–2391.
[32] F. Hoffmann, J.C.S. Zagal Montealegre, Evolution of a tactile wall-following behavior in real time, in: The 6th Online World Conference on Soft Computing in Industrial Applications, WSC6, 10–24 Sept., 2001.
[33] W. Lee, J. Hallam, H. Lund, Applying genetic programming to evolve behavior primitives and arbitrators for mobile robots, in: Proceedings of the 1997 IEEE International Conference on Evolutionary Computation, 1997, pp. 495–499.
[34] G.J. Barlow, L.S. Mattos, E. Grant, C.K. Oh, Transference of evolved unmanned aerial vehicle controllers to a wheeled mobile robot, in: Proceedings of the IEEE International Conference on Robotics and Automation, Barcelona, Spain, April 2005.
[35] H. Nakamura, A. Ishiguro, Y. Uchilkawa, Evolutionary construction of behavior arbitration mechanisms based on dynamically-rearranging neural networks, in: Proceedings of the 2000 Congress on Evolutionary Computation, vol. 1, IEEE, 2000, pp. 158–165.
[36] F. Pasemann, U. Steinmetz, M. Hülse, B. Lara, Evolving brain structures for robot control, in: IWANN'01 Proceedings LNCS, 2085, vol. II, Springer-Verlag, Granada, Spain, 2001, pp. 410–417.
[37] O Miglino, D. Denaro, G. Tascini, D Parisi, Detour behavior in evolving robots: Are internal representations necessary? in: Proceedings of the First European Workshop on Evolutionary Robotics, Springer-Verlag, 1998, pp. 59–70.
[38] G. Buason, N. Bergfeldt, T. Ziemke, Brains, bodies, and beyond: Competitive co-evolution of robot controllers, morphologies and environments, Genetic Programming and Evolvable Machines 6 (1) (2005) 25–51.
[39] D. Cliff, G.F. Miller, Co-evolution of pursuit and evasion II: Simulation methods and results, in: P. Maes, M. Mataric, J.-A. Meyer, J. Pollack, S.W. Wilson (Eds.), From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior, SAB96, MIT Press, Bradford Books, 1996, pp. 506–515.
[40] K. Chellapilla, D.B. Fogel, Evolving an expert checkers playing program without using human expertise, IEEE Transactions on Evolutionary Computation 5 (4) (2001) 422–428.
[41] A. Lubberts, R. Miikkulainen, Co-evolving a go-playing neural network, in: Coevolution: Turning Algorithms upon Themselves, Birds-of-a-Feather Workshop, Genetic and Evolutionary Computation Conference, GECCO-2001, San Francisco, CA, 2001.
[42] T. Gomi, K. Ide, Evolution of gaits of a legged robot, in: The 1998 IEEE International Conference on Fuzzy Systems, Proceedings of the 1998 IEEE World Congress on Computational Intelligence, vol. 1, 4–9 May, 1998, pp. 159–164.
[43] V. Matellán, C. Fernández, J.M. Molina, Genetic learning of fuzzy reactive controllers, Robotics and Autonomous Systems 25 (1–2) (1998) 33–41.
[44] J. Liu, C.K. Pok, H.K. Keung, Learning coordinated maneuvers in complex environments: A sumo experiment, in: Proceedings of the 1999 Congress on Evolutionary Computation, CEC 99, vol. 1, 6–9 July, 1999, pp. 343–349.
[45] H.-S. Seok, K.-J. Lee, J.-G. Joung, B.-T. Zhang, An on-line learning method for object-locating robots using genetic programming on evolvable hardware, in: Proceedings of the Fifth International Symposium on Artificial Life and Robotics, AROB'00, vol. 1, 2000, pp. 321–324.
[46] J. Ziegler, W. Banzhaf, Evolving control metabolisms for a robot, Artificial Life 7 (2) (2001) 171–190.
[47] F. Hoffmann, G. Pfister, Evolutionary learning of a fuzzy control rule base for an autonomous vehicle, in: Proceedings of the Fifth International Conference IPMU: Information Processing and Management of Uncertainty in Knowledge-Based Systems, Granada, Spain, July 1996, pp. 659–664.
[48] A. Thompson, Evolving electronic robot controllers that exploit hardware resources, Advances in Artificial Life: Proceedings of the 3rd European Conference on Artificial Life, ECAL95, Lausanne, vol. 929, Springer-Verlag, 1995, pp. 640–656.
[49] A. Ishiguro, S. Tokura, T. Kondo, Y. Uchikawa, Reduction of the gap between simulated and real environments in evolutionary robotics: A dynamically-rearranging neural network approach, in: Proceedings of the 1999 IEEE International Conference on Systems, Man, and Cybernetics, vol. 3, 1999, pp. 239–244.
[50] M. Ebner, A. Zell, Evolving a behavior-based control architecture — From simulations to the real world, in: Proceedings of the Genetic and Evolutionary Computation Conference, vol. 2, 13–17 July 1999, pp. 1009–1014.
[51] W. Lee, Evolving autonomous robot/sensors: From controller to morphology, IEICE Transactions of Information and Systems E83-D (2) (2000) 200–210.
[52] I.G. Sprinkhuizen-Kuyper, R. Kortmann, E.O. Postma, Fitness functions for evolving box-pushing behaviour, in: A. van den Bosch, H. Weigand (Eds.), Proceedings of the Twelfth Belgium–Netherlands Artificial Intelligence Conference, 2000, pp. 275–282.
[53] K. Wolff, P. Nordin, Evolution of efficient gait with an autonomous biped robot using visual feedback, in: Proceedings of the 2nd IEEE-RAS International Conference on Humanoid Robots, Humanoids 2001, Tokyo, Japan, 2001, pp. 99–106.
[54] U. Nehmzow, Physically embedded genetic algorithm learning in multi-robot scenarios: The PEGA algorithm, in: Proceedings of the Second International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems, Edinburgh, 2002.
[55] E.D.V. Simões, D.A.C. Barone, Predation: An approach to improving the evolution of real robots with a distributed evolutionary controller, in: Proceedings of the IEEE International Conference on Robot Automation, ICRA'02, Washington, DC, May, 2002, pp. 664–669.
[56] D. Marocco, D. Floreano, Active vision and feature selection in evolutionary behavioral systems, in: J. Hallam, D. Floreano, G. Hayes, J. Meyer (Eds.), From Animals to Animats 7, MIT Press, Cambridge, MA, 2002.
[57] M. Okura, A. Matsumoto, H. Ikeda, K. Murase, Artificial evolution of FPGA that controls a miniature mobile robot Khepera, in: Proceedings of the Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE 2003), 4–6 Aug., 2003, vol. 3 pp. 2858–2863.
[58] D. Gu, H. Hu, J. Reynolds, E. Tsang, GA-based learning in behaviour based robotics, in: Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation, Kobe, Japan, 16–20 July, 2003, vol. 3, pp. 1521–1526.
[59] A.L. Nelson, E. Grant, J.M. Galeotti, S. Rhody, Maze exploration behaviors using an integrated evolutionary robotics environment, Journal of Robotics and Autonomous Systems 46 (3) (2004) 159–173.
[60] A. Boeing, S. Hanham, T. Braunl, Evolving autonomous biped control from simulation to reality, in: Proceedings of the 2nd International Conference on Autonomous Robots and Agents, Palmerston North, New Zealand, 13–15 Dec., 2004, pp. 440–445.
[61] G.S. Hornby, S. Takamura, T. Yamamoto, M. Fujita, Autonomous evolution of dynamic gaits with two quadruped robots, IEEE Transactions on Robotics 21 (3) (2005) 402–410.
[62] S. Kamio, H. Iba, Adaptation technique for integrating genetic programming and reinforcement learning for real robots, IEEE Transactions on Evolutionary Computation 9 (3) (2005) 318–333.

Please cite this article in press as: A.L. Nelson, et al., Fitness functions in evolutionary robotics: A survey and analysis, Robotics and Autonomous Systems (2008), doi:10.1016/j.robot.2008.09.009

[63] G.B. Parker, R. Georgescu, Using cyclic genetic algorithms to evolve multi-loop control programs, in: Proceedings of the 2005 IEEE International Conference on Mechatronics and Automation, ICMA 2005, July 2005, Niagara Falls, Ontario, Canada.

[64] E.J.P. Earon, T.D. Barfoot, G.M.T. D'Eleuterio, From the sea to the sidewalk: the evolution of hexapod walking gaits by a genetic algorithm, in: Proceedings of the International Conference on Evolvable Systems, ICES, Edinburgh, Scotland, 17–19 April, 2000.

[65] G.S Hornby, H. Lipson, J.B. Pollack, Evolution of generative design systems for modular physical robots, in: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA'01, vol. 4, 2001, pp. 4146–4151.

[66] P. Augustsson, K. Wolff, P. Nordin, Creation of a learning, flying robot by means of evolution, in: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2002, New York, 9–13 July, 2002, Morgan Kaufmann, 2002, pp. 1279–1285.

[67] J. Zufferey, D. Floreano, M. van Leeuwen, T. Merenda, Evolving vision based flying robot, in: Bülthoff, Lee, Poggio, Wallraven (Eds.), Proceedings of the 2nd International Workshop on Biologically Motivated Computer Vision, in: LNCS, vol. 2525, Springer-Verlag, Berlin, pp. 592–600.

[68] I. Macinnes, E. Di Paolo, Crawling out of the simulation: Evolving real robot morphologies using cheap, reusable modules, in: Proceedings of the International Conference on Artificial Life, ALIFE9, Boston, MA, 12–15 Sept., MIT Press, 2004, pp. 94–99.

[69] V. Zykov, J. Bongard, H. Lipson, Evolving dynamic gaits on a physical robot, in: 2004 Genetic and Evolutionary Computation Conference, GECCO, Seattle, WA, 2004.

[70] S. Chernova, M. Veloso, An evolutionary approach to gait learning for four-legged robots, in: Proceedings of the IEEE International Conference on Intelligent Robots and Systems, IROS'04, vol. 3, Sendai, Japan, Sept. 28–Oct. 2, 2004, pp. 2562–2567.

[71] D. Filliat, J. Kodjabachian, J.A. Meyer, Incremental evolution of neural controllers for navigation in a 6 1egged robot, in: Sugisaka, Tanaka (Eds.), Proceedings of the Fourth International Symposium on Artificial Life and Robotics, Oita Univ. Press, 1999.

[72] F. Gruau, Automatic definition of modular neural networks, Adaptive Behavior 2 (1995) 151–183.

[73] M. Quinn, Evolving communication without dedicated communication channels, in: J. Kelemen, P. Sosik (Eds.), Advances in Artificial Life: Sixth European Conference on Artificial Life, ECAL 2001, Prague, Czech Republic, Sept. 2001, Springer, 2001, pp. 357–366.

[74] T. Ziemke, Remembering how to behave: Recurrent neural networks for adaptive robot behavior, in: Medsker, Jain (Eds.), Recurrent Neural Networks: Design and Applications, CRC Press, Boca Raton, 1999.

[75] E. Tuci, M. Quinn, I. Harvey, Evolving fixed-weight networks for learning robots, in: Proc. 2002 Congress on Evolutionary Computing, Honolulu HI, vol. 2, 2002, pp. 1970–1975.

[76] I. Ashiru, C.A. Czarnecki, Evolving communicating controllers for multiple mobile robot systems, in: Proceedings of the 1998 IEEE International Conference on Robotics and Automation, vol. 4, 1998, pp. 3498–3503.

[77] M. Quinn, Evolving cooperative homogeneous multi-robot teams, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'00, vol. 3, Takamatsu Japan, 2000, pp. 1798–1803.

[78] G. Baldassarre, S. Nolfi, D. Parisi, Evolving mobile robots able to display collective behaviors, in: C.K. Hemelrijk, E. Bonabeau (Eds.), Proceedings of the International Workshop on Self-Organisation and Evolution of Social Behaviour, Monte Verità, Ascona, Switzerland, 8–13 Sept., 2002, pp. 11–22.

[79] M. Dorigo, V. Trianni, E. Sahin, T. Labella, R. Grossy, G. Baldassarre, S. Nolfi, J.-l. Deneubourg, F. Mondada, D. Floreano, L. Gambardella, Evolving self-organizing behaviors for a swarm-bot, Autonomous Robots 17 (23) (2004) 223–245.

[80] R. Groß, M. Bonani, F. Mondada, M. Dorigo, Autonomous self-assembly in a swarmbot, in: K. Murase, K. Sekiyama, N. Kubota, T. Naniwa, J. Sitte (Eds.), Proceedings of the Third International Symposium on Autonomous Minirobots for Research and Edutainment, Springer Verlag, Berlin, 2006, pp. 314–322.

[81] J.A. Driscoll, R.A. Peters II, A development environment for evolutionary robotics, in: 2000 IEEE International Conference on Systems, Man, and Cybernetics, vol, 5, 2000, pp. 3841–3845.

[82] F. Gomez, R. Miikkulainen, Incremental evolution of complex general behavior, Adaptive Behavior 5 (1997) 317–342.

[83] G.J. Barlow, C.K. Oh, E. Grant, Incremental evolution of autonomous controllers for unmanned aerial vehicles using multi-objective genetic programming, in: Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems, CIS, Singapore, Dec. 2004, pp. 688–693.

[84] D. Cliff, G.F. Miller, Tracking the red queen: Measurements of adaptive progress in co-evolutionary simulations, in: F. Moran, A. Moreno, J.J. Merelo, P. Cachon (Eds.), Proceedings of the Third European Conference on Artificial Life: Advances in Artificial Life, ECAL95, in: Lecture Notes in Artificial Intelligence, vol. 929, Springer-Verlag, 1995, pp. 200–218.

[85] W. Lee, J. Hallam, H.H. Lund, A hybrid GP/GA approach for co-evolving controllers and robot bodies to achieve fitness-specified task, in: Proceedings of the IEEE 3rd International Conference on Evolutionary Computation, 20–22 May, 1996, pp. 384–389.

[86] N. Mekel-Bobrov, S.L. Gilbert, P.D. Evans, E.J. Vallender, J.R. Anderson, R.R. Hudson, S.A. Tishkoff, B.T. Lahn, Ongoing adaptive evolution of ASPM, a brain size determinant in homo sapiens, Science 309 (5741) (2005) 1720–1722.

[87] J. Walker, S. Garrett, M. Wilson, The balance between initial training and lifelong adaptation in evolving robot controllers, IEEE Transactions on Systems, Man and Cybernetics 36 (2) (2006) 423–432.

[88] L.E. Parker, ALLIANCE: An architecture for fault tolerant multi-robot cooperation, IEEE Transactions on Robotics and Automation 14 (2) (1998) 220–240.

[89] A. Orebäck, H.I. Christensen, Evaluation of architectures for mobile robotics, Autonomous Robots 14 (1) (2003) 33–49.

[90] T.M. Mitchell, Machine Learning, McGraw Hill, 1997.

[91] W.L. Brogan, Modern Control Theory, third ed., Prentice-Hall, NJ, 1991.

[92] S. Thrun, Robotic mapping: A survey, in: G. Lakemeyer, B. Nebel (Eds.), Exploring Artificial Intelligence in the New Millennium, Morgan Kaufmann, 2002.

[93] S. Thrun, Bayesian landmark learning for mobile robot localization, Machine Learning 33 (1) (1998) 41–76.

[94] K. Dixon, P. Khosla, Learning by observation with mobile robots: a computational approach, in: Proceedings of the 2004 IEEE International Conference on Robotics and Automation, ICRA'04, New Orleans, LA, 2004, pp. 102–107.

[95] M. Mitchell, An Introduction to Genetic Algorithms, The MIT Press, Cambridge, Massachusetts, 1998.

[96] J. Galeotti, S. Rhody, A.L. Nelson, E. Grant, G. Lee, EvBots – The design and construction of a mobile robot colony for conducting evolutionary robotic experiments, in: Proceedings of the ISCA 15th International Conference: Computer Applications in Industry and Engineering, CAINE-2002, San Diego, CA, 7–9 Nov., 2002, pp. 86–91.

[97] F. Mondada, E. Franzi, P. Ienne, Mobile robot miniaturization: A tool for investigation in control algorithms, in: The 3rd International Symposium on Experimental Robotics, ISER'93, Kyoto, Japan, October 1993, in: Lecture Notes in Control and Information Sciences, vol. 200, 1993, pp. 501–513.

[98] T.M.C. Smith, P. Husbands, P. Layzell, M. O'Shea, Fitness landscapes and evolvability, Evolutionary Computation 10 (1) (2002) 1–34.

[99] F. Kaplan, P. Oudeyer, E. Kubinyi, A. Miklosi, Robotic clicker training, Robotics and Autonomous Systems 38 (3–4) (2002) 197–206.

[100] H.H. Lund, O. Miglino, L. Pagliarini, A. Billard, A. Ijspeert, Evolutionary robotics—A children's game, in: Evolutionary Computation Proceedings, 1998 IEEE World Congress on Computational Intelligence, 1998, pp. 154–158.

[101] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: Optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man and Cybernetics, Part B 26 (1) (1996) 29–41.

[102] L.N. de Castro, J. Timmis, Artificial immune systems as a novel soft computing paradigm, Soft Computing 7 (8) (2003) 526–544.

[103] J.A. Meyer, A. Guillot, Simulation of adaptive behavior in animats: Review and prospect, in: J.A. Meyer, S. Wilson (Eds.), From Animals to Animats, Proceedings of the First International Conference on Simulation of Adaptive Behavior, SAB-90, MIT Press, 1991, pp. 2–14.

[104] A. Guillot, J.A. Meyer, The animat contribution to cognitive systems research, Journal of Cognitive Systems Research 2 (2) (2001) 157–165.

[105] G.J. Barlow, C.K. Oh, Robustness analysis of genetic programming controllers for unmanned aerial vehicles, in: Proceedings of the 8th Annual Conference on Genetic And Evolutionary Computation, GECCO'06, 2006, pp. 135–142.

[106] M. Kaiser, H. Friedrich, R. Buckingham, K. Khodabandehloo, S. Tomlinson, Towards a general measure of skill for learning robots, in: Proceedings of the 5th European Workshop on Learning Robots, Bari, Italy, 1996.

[107] A. Nelson, E. Grant, Aggregate selection in evolutionary robotics, in: N. Nedjah, L. Coelho, L. Mourelle (Eds.), Mobile Robots: The Evolutionary Approach, in: Studies in Computational Intelligence, vol. 50, Springer, 2007, pp. 63–88.

[108] M. Lungarella, G. Metta, R. Pfeifer, G. Sandini, Developmental robotics: A survey, Connection Science 15 (4) (2003) 151–190.

[109] M. Asada, K.F. MacDorman, H. Ishiguro, Y. Kuniyoshi, Cognitive developmental robotics as a new paradigm for the design of humanoid robots, Robotics and Autonomous Systems 37 (2–3) (2001) 185–193.

[110] V. Trianni, M. Dorigo, Self-organisation and communication in groups of simulated and physical robots, Biological Cybernetics 95 (3) (2006) 213–231.

Dr. Andrew L. Nelson was born in Laramie, Wyoming in 1967. He received his B.S. degree with specialization in Computer Science from the Evergreen State College in Olympia Washington in 1990. He received his M.S. in Electrical Engineering from North Carolina State University in 2000. He received his Ph.D. in Electrical Engineering at the Center for Robotics and Intelligent Machines (CRIM) at North Carolina State University in 2003. Between 2003 and 2005 he was a visiting researcher at the University of South Florida. Currently he is a researcher at Androtics LLC, Tucson AZ and Santa Cruz CA. His main interests are in the fields of fully autonomous robot control, bio-inspired robot control and evolutionary robotics. His robotics work has included applying artificial evolution to synthesize controllers for swarms of autonomous robots as well as the development of fuzzy-logic-based controllers for robot navigation. He pursues work in artificial neural networks, genetic algorithms and soft computing related to autonomous machine control. He has also conducted research in diverse fields including electric machine design and molecular biology.

Please cite this article in press as: A.L. Nelson, et al., Fitness functions in evolutionary robotics: A survey and analysis, Robotics and Autonomous Systems (2008), doi:10.1016/j.robot.2008.09.009

Gregory J. Barlow is a Ph.D. candidate at the Robotics Institute at Carnegie Mellon University. He received B.S. degrees in electrical and computer engineering in 2003 from North Carolina State University in 2003 and an M.S. degree in electrical engineering from North Carolina State University in 2004. His research interests include memory-enhanced algorithms for dynamic problems, dynamic optimization with evolutionary algorithms, and evolutionary robotics.

Dr. Lefteris Doitsidis received his B.S. degree from the Production Engineering and Management Department of the Technical University of Crete, Chania, Greece, in 2000. In 2002 he was awarded his M.S degree in Production Systems and in 2008 he received his Ph.D. at the same department. Since 2002, he has been a researcher at the Intelligent Systems and Robotics Laboratory of the same department. From August 2003 to June 2004 he was a visiting scholar at the University of South Florida, FL, U.S.A. Beginning in 2004 he has been, and continues to be an instructor at the Technological Educational Institute of Crete. His research interests lie in the areas of multirobot teams, autonomous operation and navigation of unmanned vehicles and evolutionary computation.

Please cite this article in press as: A.L. Nelson, et al., Fitness functions in evolutionary robotics: A survey and analysis, Robotics and Autonomous Systems (2008), doi:10.1016/j.robot.2008.09.009

IX Introduction to Humanoid Robotics

Humanoid robotics is a field of robotics focused on designing, developing and studying robots that resemble the human body in form and function.

These robots are engineered to replicate human behaviours, movements, and interactions.

• Uncanny Valley Effect

Introduced by Masahiro Mori in 1970

Describes the phenomenon where robots or other human-like entities elicit feelings of unease or eeriness as their presence and behaviour approach, but not fully achieve, human likeness.

Key Aspects:

- Human Likeness vs Familiarity
  As an entity becomes more human-like, people generally find it more familiar and relatable.

  However, at a certain point, sable imperfections, (e.g, unnatural movements, slightly distorted features) lead to a sense of unease, resulting in a sharp drop in perceived familiarity. This drop is the uncanny valley.

- Causes of the Effect:

  - Violation of Expectations: The entity elicits human-directed expectations but fail to fulfill them, creating discomfort.

  - Fear of Death or Mortality: Some theories suggest that uncanny features subconsciously remind people of death, triggering defensive mechanisms

  - Evolutionary Mechanisms: Dislike for entities that appear sick, dead or non-human could be rooted in evolutionary survival instincts.

- Amplification by Movement
  If a human-like entity moves unnaturally, the sensation of strangeness is heightened. For instance, mechanical or jerky movements exaggerate the uncanny valley effect.

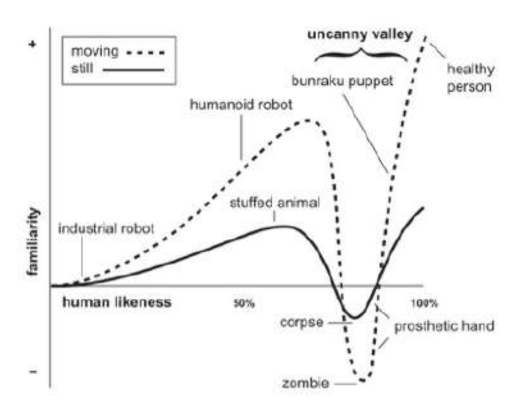


Figure 1: As a robot designer, Mori graphed what he saw as the relation between human likeness and perceived familiarity: familiarity increases with human likeness until a point is reached at which subtle differences in appearance and behavior create an unnerving effect [Mori, 1970]. This he called the *uncanny valley*. According to Mori, movement amplifies the effect.

- x-axis: Represents the likeliness of an entity, ranging from industrial robots (low likeliness) to healthy humas (high likeliness)
- y-axis: Represents the familiarity or +ve emotional response elicited by the entity.

Mori's caution:

Designers should aim for the first peak (moderate human likeliness) rather than risking the uncanny valley by attempting to create entities too close to humans.

- **Applications of Humanoid Robotics**
  - 3Ds - Dirty, dangerous or Dull work
  - House help
  - care of the young, elderly, infirm
  - entertainment applications(in a broad sense)

- **Outline Taxonomy of Humanoid Robots**
  - Level 0: Replicant

    A replicant is a robot designed to be indistinguishable from a human in appearance and behaviour.

    - Hyper-realistic human features
    - Advanced behavioural mimicry, such as natural speech, emotion simulation, and social interactions.
    - Typically used in experimental or high-fidelity applications such as human behavioural studies.

    e.g., Geminoid Robot

  - Level 1: Android

    An android is a robot with a human-like appearance and behaviour but may not fully replicate human realism.

    - Designed to resemble humans, with a focus on realistic faces, body proportions, and movements.
    - May include human-like gestures, language capabilities, and sensory inputs.

- Can be differentiated from humans upon closer inspection or in specific contexts (e.g., robotic joints, slightly mechanical movements).

- Applications: Human-robot interaction research, customer service, educational roles.

- Example: Honda's ASIMO, Sophia by Hanson Robotics.

- Level n-3: Humanoid

A humanoid robot shares the general physical structure of a human, but does not aim for exact replication.

- Focused more on functionality & versatility than appearance
- Simplified design with mechanical features such as metallic exteriors or exposed joints.
- Movement and control systems modeled after human biomechanics but may have limitations in precision or fluidity
- Applications: Industrial Automation
- Example: Pepper robot by SoftBank, Atlas by Boston Dynamics

- Level n-2: Inferior Humanoid (I.H)

Inferior humanoids are robots with limited resemblance to humans, focusing more on functional adaptation than human-like aesthetics

- May lack a fully human structure but still have human-inspired functionality (e.g., robotic arms and wheels instead of legs).

- Less emphasis on facial expressions or detailed humanoid design.

- Designed for practical tasks in human environments without complex social interaction capabilities.

- Applications: Factory automation, hazardous material handling, or simple repetitive tasks.

- Example: Robonaut by NASA

- Level n: Built-for-Human (B.F.H)

Robots built specifically to operate in human environments, often focusing on ergonomics and safety rather than aesthetics or movement.

- Designed to assist humans or complement their tasks.
- Optimized for compatibility with human tools, interfaces, or systems
- Strong emphasis on utility, reliability, and adaptability in human-centric environments.
- Applications: Domestic robots, healthcare assistance, or logistical tasks in human-designed spaces.
- Example: Roomba vacuum cleaner, Amazon's Kiva robot

X̄ Brief Overview of Selected Humanoid Robot Platforms

	Mfg
Darwin-Mini	Robotis (Korean)
Bioloid (Premium)	Robotis (Korean)
Nao	Aldebaran Robotics
iCub	
REEM-C	PAL robotics
TEO	Carlos III university of madrid

- **ASIMO**

  Advanced Step in Innovative MObility

  - 130 cm tall
  - Tasks:
    - serving at tables
    - Operate door handles
    - turn switches on & off
    - interact with everyday devices

- **E2DR**

  Experimental Robot type 2 for disaster response

  Features:
  - novel cooling system
  - ability to manoeuvre on stairs & different ladder types
  - Being to operate in narrow free spaces

- **PETMAN**
  - designed for testing chemical protective clothing
  - funding from DARPA
  - body shape - 50th percentile male body shape.

- **Atlas**

  By Boston Dynamics

  - prev version (2015) - 1.5 m tall, 80 kg
    - construction - Titanium, aircraft-grade aluminium
    - 28 DOF
    - Hydraulic actuators
    - walk over uneven & slippery surfaces, jump, backflips
  - Current (Apr 2024) : All electric
    - Better range of motion than hydraulic
    - Gripper options
    - AI - fully autonomous

LBM - Large Behaviour models (Robots doing useful stuff in physical world)

- Digit / Cassie

Cassie

bipedal Robot

Agility Robotics, 2017

passive dynamics - great part of a robot's agility derives from inherent mechanical properties of its structure.

Digit

Augmented Cassie robot lower body structure with arms, a torso & wide variety of sensors

175 cm tall

≈ 65 kg

can carry payload upto 16 kg

Features:

- Autonomous charging capability
- Ability to walk up & down inclined surfaces
- Ability to navigate over various types of unstructured terrain
- can pickup & deposit objects.

XI The Reality Gap

The discrepancy between behaviours evolved in simulation & the actual observed behaviours on the real robot, resulting in inefficient controllers.

Approaches to Resolve the Reality Gap Issue:

1. The "back to reality" approach (Zagal, 2007)

   coevolution of robot controller (and/or morphology) and its simulator.

   feedback used to alter simulator

   feedback based on differences in observed behavioural fitnesses as opposed to sensor data comparisons

2. Fitness Function Correction interleaving simulated & real data (Iocchi 2007)

   Interleaving of expt b/w real robot & simulator

   1:5 1 experiment on real robot to every 5 in simulation

   Determines discrepancies b/w the results for the simulator as opposed to those from the real robot.

   discrepancy used to alter fitness function which evaluated results from the simulator.

3. Using an EA to tune the parameters of a simulator (Laue & Hebbel, 2009)

   EA in multistage approach to determine parameters for an existing simulation engine.

   Hope: closest possible approx of behaviour of simulator to real world.

4. The Grounded simulated learning approach (Farch, 2013)

   1. Start with imperfect simulator
   2. Evolve behaviour(s) in simulator
   3. Implement evolved behaviour on robot
   4. Calc discrepancy in simulation v/s reality
   5. Modify simulator based on discrepancy, bring it closer to reality.
   6. Repeat 2-5 until satisfactory

5. The dual simulator approach (Eaton, 2016)

1. Evolve behaviours in one simulation

2. Take evolved behaviours & transfer it onto a 2^{nd} simulator for further evaluation & testing.

3. When behaviours operate successfully on both, test on real robot.

Advantage: weaknesses in one simulation is unlikely to be manifested in another.

International Journal of Social Robotics
https://doi.org/10.1007/s12369-021-00827-0

Exploring the Perceptions of Cognitive and Affective Capabilities of Four, Real, Physical Robots with a Decreasing Degree of Morphological Human Likeness

Leopoldina Fortunati[1] · Anna Maria Manganelli[2] · Joachim Höflich[3] · Giovanni Ferrin[1]

Accepted: 30 August 2021

Abstract

This paper describes an investigation of student perceptions of the cognitive and affective capabilities of four robots that have a decreasing degree of morphological human likeness. We showed and illustrated the robots (i.e., InMoov, Padbot, Joy Robot and Turtlebot) to 62 students. After showing the students each of these robots, and explaining their main features and capabilities, we administered a fill-in questionnaire to the students. Our main hypothesis was that the perception of a robot's cognitive and affective capabilities varied in correspondence with their appearance and in particular with their different degree of human likeness. The main results of this study indicate that the scores attributed to the cognitive and emotional capabilities of these robots are not modulated correspondingly to their different morphological similarity to humans. Furthermore, overall, the scores given to all of these robots regarding their ability to explicate mental functions are low, and even lower scores are given to their ability to feel emotions. There is a split between InMoov, the robot which has the highest degree of human likeness, and all of the others. Our results also indicate that: (1) morphological similarity of a robot to humans is not perceived automatically as such by observers, which is not considered a value in itself for the robot; and (2) even at lower levels of robot–human likeness, an uncanny valley effect arises but is quite mitigated by curiosity.

Keywords Robot appearance · Theories of mind · Theories of emotions · Theory of social representations · Uncanny valley theory

1 Introduction

Human–robot interaction is increasingly spreading in society and it has acquired a certain importance in the articulated domain of the interactions that individuals experience in their everyday life. Therefore, it is not surprising that the scientific community shows a growing interest in the development of robotic models that can enhance the interaction between humans and robots. This paper explores the extent to which some present social robots are perceived by people in terms of their cognitive and affective capabilities. In other words, we intend to investigate to what extent current models (beyond good intentions) are really effective in terms of simulation of mental functions and emotions by robots. In this study, we decided to use four, real, physical robots: InMoov, Padbot, Joy Robot, and Turtlebot (Fig. 1).

A lot of prior work has used images or videos rather than real, physical robots. While interesting results have emerged from this rich body of studies, these results have to be treated with caution because we know that interaction with physical robots changes perception (e.g., [2, 33, 54, 76]). Several studies have compared two or more real, physical robots with different appearances in interactions with humans (e.g., [11, 39, 44, 47, 53, 57]. For example, Li et al. [53] compared three, real physical robots with different shapes (one anthropomorphic, one zoomorphic and one machine-like). They started from the assumption that a robot with recognizable eyes and a robot with a human-like body shape are more likely to be categorized as human-like, while a robot without facial features and a robot with wheels and tracks were more likely to be categorized as machine-like robots [78]. They found

✉ Leopoldina Fortunati
leopoldina.fortunati@uniud.it

1 Department of Mathematics, Computer Science and Physics, Università degli Studi di Udine, Udine, Italy

2 Department of Philosophy, Sociology, Education and Applied Psychology, University of Padova, Padova, Italy

3 Department of Media and Communication, University of Erfurt, Erfurt, Germany

Published online: 16 November 2021

Fig. 1 The four robots used in the study: InMoov, Padbot, Joy Robot and Turtlebot (right to left, and top to bottom)

that the robot's appearance generates strong and positive correlations between interaction performance (understood as active response and engagement) and preference (intended as likeability, trust, and satisfaction, in the human–robot interaction). However, it is difficult to identify in their review which element of appearance, among all those that compose it, generates the effects that were examined. In our study thus we decided to use robots that are characterized by a decreasing degree of morphological human-likeness. To establish their decreasing morphology, we found inspiration in Eaton's taxonomy [17, pp. 36–37] of n + 1 different levels of robots, which classified robots as follows:

- 0 Replicant, which looks perfectly like a human being, both in physical aspect and behavior, except for the biological functions; people with normal senses cannot distinguish it from a human.
- 1 Android, which presents a morphology and behavior very close to those of human and is characterized by high levels of intelligence and dexterity.
- n − 3 Humanoid, which is close to a human in both "body" and "brain" but without the possibility of mistaking the robot for a human being.
- n − 2 Inferior Humanoid, which has a broad, human morphology but with a reasonable intelligence and dexterity, mainly oriented to a limited set of tasks.
- n − 1 Human-inspired, but unlike a human. It may be wheeled or have bipedal capabilities and limited intelligence as well as dexterity.
- n Built-for-Humans, which does not look like a human and is able to operate in most environments, performing a limited set of tasks.

Among our robots, InMoov corresponds to Humanoid, which is very human-like but without the possibility of mistaking the robot for a human being (n − 3). Padbot and Joy Robot correspond to a Human-inspired robot but unlike a human (n − 1); the fact that they belong to the same level n − 1, however, is not so problematic for us because this layer is so large that it includes very different robots. In our case, Padbot has a long neck and a face-like screen over this neck where the real face of the operator can be incorporated. This robot can talk and has the ability to move, which presents a certain similarity to the human body and is thus at the highest level of the layer. Joy Robot, which is similar to a doll but with expressive eyes and mouth, and arms that it can move, is at a lower level of the layer n − 1. Finally, the Turtlebot corresponds to a robot Built-for-humans, which does not look like a human and is able to operate in most environments (n). Our four robots are on the medium–low level of the human likeness scale for robots. Consequently, they should be free from any uncanny-valley effect that, according to Mori [61], should arise at the first levels of this taxonomy: Replicant = 0 and Android = 1. However, is this the case? We will return to this question in the next section. Overall, in our exploration of the attribution of mental and emotional capabilities of these four robots, we found inspiration in research that was carried out in Australia, China, and Italy [34] that has explored the mental capacities and emotions that are believed to differentiate humans from animals, robots, and supernatural beings. This study showed that 12 years ago robots were given low levels of mental capacities, and even lower levels of primary and secondary emotions by Chinese, Australian and Italian respondents. It found that distinctive patterns of mental capacities distinguished humans from non-humans in these three cultures. Robots were perceived as lacking capacities of perception and rational cognition, and were even more deficient in emotion. During the 12 years that have passed since that research, many social processes and phenomena have occurred, such as the further development of new technologies (e.g. [5, 25, 55]); the digitalization of the whole field of information, communication and entertainment (e.g. [1, 38, 63]), including mass media (e.g. [72]), and the diffusion of much more advanced social robots and virtual assistants in several sectors of society [3, 22, 30].

These phenomena have probably increased people's familiarity with different degrees of media products that simulate human life likeness (e.g., reality shows; [65]) and visual products with robots (e.g., movies, television series, cartoons, etc.) that anticipate many factual aspects of robotic innovations and merge these with elements of fantasy [24]. The increasing technological sophistication in the creation of more advanced social robots seems to be matched by increasing technological discernment on the part of potential users [73]. Consequently, these sociotechnical changes could have changed people's perceptions of robots.

We have taken three theoretical points of reference for our study: for the robots' mental functions, we chose the theory of the dimensions of mind perception proposed by Gray et al. [28]; for the robots' emotions, we chose the theory of infra-humanization elaborated by Leyens et al. [52]; and for the robot's conception, we chose the theory of social representations [62]. The first and second theoretical approaches are pioneering contributions to the study of human/non-human contrast. In the present study, we considered a selection of the classification of mental states and processes, as described by D'Andrade [14], on the basis of the dimension of mind perception approach. This taxonomy consists of four dimensions: Perceptions (hearing, seeing, smelling, tasting), Wishes (desiring, needing, wanting, wishing), Thoughts (imagining, knowing, reasoning, thinking) and Intentions (choosing, deciding, expecting, intending, planning). The second theory is infra-humanization, which regards emotions and focuses on a subtle form of dis-humanization, and on the contrast between human/non-human. While explicit dehumanization equates its target (e.g., the members of a minority group) with animals, devils, machines, or objects to deprive them of their humanity [35], infra-humanization reaches the same objective by denying these members the experience of secondary emotions. This last theory is based on the distinction between primary and secondary emotions: the first indicates the emotional reactions that humans have in common with animals (e.g., anger, fear, pleasure etc.) and the second, which is also called sentiments, indicates the emotions that are rated as specifically human (e.g., pride, nostalgia, shame, remorse etc.). The third theory is that of social representations, which helps us capture how our respondents spontaneously perceive and conceive these robots [62]. By social representations we mean the set of concepts, statements and explanations that arise in daily life in the course of interpersonal communications; they can be considered as organizing principles and specific ways of expressing knowledge in a society, shared by large groups of people. All social representations tend to transform what is extraneous or new into something familiar.

This paper is structured as follows: in the next section, we will analyze the question of robots' appearance and the uncanny-valley theory, in the following section we will illustrate the method and the measures applied. The illustration of the results of the research will be the topic of a dedicated section. These results will be discussed in the final section, along with an examination of the strong and weak points of the research, as well as the future paths of the research on the relation between perceptions of cognitive and affective capabilities of robots with a decreasing degree of morphological human likeness.

2 Robot Appearance and Uncanny Valley Theory

So far, research has shown that a robot's physiognomy affects the mental model that humans have of the robot itself. In other words, it changes people's perceptions of a robot's human likeness, knowledge, and sociability [66]. For instance, Powers and Kiesler found that a robot with a shorter chin makes the robot's eyes appear to be larger in the face. This typology of face (large eyes and small chins) is called baby faced. A baby-faced robot seems to replicate what Berry and McArthur [7] found for humans: baby-faced men are perceived to be more naive, honest, kind, and warm. As robots begin to enter people's everyday life and houses and ordinary people start to interact with them, the question of how their appearance affects people's perception and behavior becomes increasingly important. Four main fields of investigation regarding robots' appearance have been explored so far. The first concerns the facial features and their importance in respect to the perceived similarity of robots to humans [70]. Research has found that the degree to which a robot is perceived as human-like is related to the number of features that its head presents (e.g., eyes, nose, mouth; DiSalvo et al. [16]). More recently, McGinn [60] has presented a review of the literature exploring at a macro level the capability for social interaction with robots with head-like features and has proposed a taxonomy of robots' social interfaces. The second field of the influence of robots' appearance focuses on the evaluation of their cognitive abilities. A number of research studies have shown that the expectation of a robot's cognitive capabilities is connected to the robot's appearance (e.g., [29, 36, 37, 48, 78]). In particular, Krach et al. [48] found that increased perceived human-likeness (in terms of appearance) implicates increased expectations about the robot's cognitive abilities, in the sense that humans ascribe humanoid robots with mental states, though on a lower level. The degree of a robot's human-likeness modulates its perception and biases "mental" state attribution. This modulation seems to be linear: the more that a robot exhibits human-like features, the more people build a model of its "mind." The third field investigates the role played by human-like abilities in robots' mind perception. For instance, Küster et al. [50] found that human-like abilities are able to supply more potent cues to mind perception than appearance. The fourth field explores emotions: only a few studies have systematically investigated humans' emotional reactions towards robots (e.g., [69, 75]). The problem of a robots' appearance is central to the development of a computational model that can simulate human modality in social interaction. Taking this issue into account is crucial for robot designers and engineers because no consensus has thus far been reached as to whether it is better to use more human-like robots or mechanical-like robots, and the context probably matters. Just to report some examples, several studies have demonstrated that people empathize more with human-like robots [67], especially in the healthcare field in which humanoid robots have been shown to increase the user's propensity to imitation [64] and have proven to be effective in the treatment of people with autism [12, 27] and Alzheimer's disease [74]. Other studies have indicated that robots that are more human-like and more complex in terms of sophisticated functionality are less accepted [31]. More recent research suggests that the robot's appearance should be matched to the task that it performs [9]. According to another study, users prefer a fluffy robot as a companion but a more mechanical-like robot when they need to be reminded to take their medicines [8]. In several studies on robotics, the impression is that when researchers use a robot with a morphological similarity to humans, this objective human-likeness is taken for granted without verifying at the beginning of the study if the robot's appearance is really perceived as human-like by the participants. In this research, we asked if the gradual morphological similarity of these four robots was really perceived as such by the observers.

Not only is the appearance of social robots strategic but the issue of the uncanny valley also stands out.[1] Mori [61] hypothesized the 'uncanny valley' effect in human–robot interactions, demonstrating that there is no linear relationship between the robot's human-likeness and our affinity with them. The human likeness of a robot causes a feeling of affinity, but the affinity transforms into eeriness when a certain degree of human likeness is reached. For Mori, affinity with human like robots increases until an uncanny valley is reached, which is caused by some perceived imperfections in the near human-like forms of robots [71]. The more human-like the robots, the more pleasant they are experienced, until the point at which they start to elicit a negative emotional response: the uncanny feeling (UF) [75]. According to Mori, movement in human-like robots magnifies the uncanny valley [15, 59], given that movement is synonymous with life. The appearance of movement in an object that should be inanimate suggests greater concern than does the reverse, that is, the stillness of a person while looking alive [20]. When the boundaries between the various spheres blur (i.e., human beings, animals, plants, and the inanimate world), then the construction and representation of reality becomes perturbed (e.g. [18]). Here, we would like to further investigate if robots with a human-likeness that clearly cannot reach the point in which the uncanny valley effect is produced are able to generate this effect equally. Hopefully, Eaton's taxonomy will allow us to better understand the breaking point of robot humanness from which the uncanny valley is supposed to start.

This theory is problematic because although a number of studies have been conducted to test it, the results are contradictory. While there is much evidence that supports the existence of some uncanny valley effect (e.g., [40, 43, 56, 79]), there is also evidence that denies its existence (e.g., [31, 32]). These studies applied various methods and used various materials as stimuli (e.g., morphed pictures, videos of actual robots, robots and computer graphics, and pictures of actual humans) which were often not standardized (e.g., [4, 58, 68]). Furthermore, the studies that confirm the uncanny valley hypothesis demonstrate that there are uncanny valley responses tout-court, without exploring which elements of appearance contribute to a positive or negative perception.

Despite an increasing interest in conducting empirical research on the uncanny valley theory, these contradictory findings and shortcomings have raised concerns among researchers about the scientific standing of this theory, both for theoretical [10, 45, 77] and methodological reasons [51]. The present study aims to contribute to this debate by addressing one of the central aspects of this theory: the point from which the uncanny valley effect is generated. As we mentioned earlier, we want to explore if the uncanny valley effect can also be generated by robots with a moderated human likeness. We set up the present study before becoming aware of a very recent paper by Kim et al. [46]. In their study using the largest set of real-world robots (n = 251) currently available in open-source format (the ABOT Database),[2] they discovered that an additional valley emerged when the robots' appearance has low to moderate human-likeness. In our study, we will either confirm or deny this result.

Our research questions and hypotheses are as follows:

- RQ1. Is the attribution (or not attribution) of mental and emotional capabilities to each of these four robots modulated according to their decreasing degrees of morphological and behavioral similarity to humans?

  We expected that there will be a decreasing modularity in the attribution to these robots of mental and emotional capabilities corresponding to their decreasing degree of morphological human-likeness (H1).

[1] Mori developed an idea in the robotic field that was already introduced in Germany in the psychiatric and psychoanalytic field at the beginning of the twentieth century. Jentsch [42] a German psychiatrist, stated, as Author [20] recalled, that the uncanny-valley effect originates when a person deals with an unfamiliar object or event. Two types of doubt produce especially high discomfort: doubts about the ensoulment of a creature pretending to be alive through its appearance and/or movement, and doubts about the ensoulment of a non-alive machine, a feeling comparable to being in front of the living appearance of a dead human [42, p. 203].

Jentsch's analysis was further elaborated upon by Sigmund Freud [26] in his essay about 'The Uncanny', in which he applied a psychoanalytic approach. He assumed that uneasiness is based on the result of a psychological suppression process. The canny is understood as 'previously familiar, well-known. The prefix "un" indicates the suppression' (Freud, p. 267). Uncanny indicates a thing that we thought we recognized in a first moment but we then understood from some details to be something else. Androids and gynoids in particular can be perturbing because the animation of an object that should be inanimate appears illogical.

[2] http://abotdatabase.info/.

- RQ2. Is the morphological similarity of a robot to humans perceived as such by observers?

  We have hypothesized that a robot's morphological similarity to humans might not be perceived as such by people (H2).

- RQ3. Does the effect of the uncanny valley still arise even at lower levels of robot–human likeness?

  We have hypothesized that robots with a human-likeness that clearly cannot reach the point in which the uncanny valley effect is produced are not able to generate this effect equally (H3).

3 Participants, Materials, and Measures

3.1 Participants

Our convenience sample was composed of 62 students, 38 males and 24 females, with an average age of 21.8 years (SD = 3.72). Their ages ranged from 18 to 44 years, with almost three-fourths of the respondents aged from 18 to 21 years old. As to the nationality, 44 students were Italians and 18 were foreigners (mainly German). The presence of a certain number of German students depended on the fact that this study was co-organized by the University of Udine and the University of Erfurt within an Erasmus exchange program. Regarding the students' curricula, 11 were socio-humanistic students and 51 were techno-scientific students. The students' previous familiarity with robots was very limited: only four (6.5%) of the students have used robots at school, nine (14.5%) at the university and 12 (19.4%) at home, which is considered to be the hub of the robotization of the reproductive sphere [13, 21]. The few robots that were used at school are Turtlebot, Joy Robot, Padbot and robots created by students, such as a robotic arm; at the university the students used Alfa 1, Nao and Replika; and at home the students used Roomba, Bimby, and Alexa.

3.2 Materials

This research was designed as a live presentation and illustration to a group of students of four robots showing decreasing degrees of human likeness. The robots we used in this study, in descending order based on the degree of morphological and behavioral similarity to humans, are: InMoov, Padbot, Joy Robot and Turtlebot (Fig. 1).

InMoov is the first open-source life-sized humanoid robot to be 3D printed and animated. It was a personal project designed in 2012 by Gael Langevin, a French sculptor and designer, and is easily replicable in any home by means of a 3D printer. Padbot U1 is a commercial telepresence bot that was designed by a Chinese company (Tianhe, Guangzhou). It has a long neck and face-like screen over this neck, where the real face of the operator can be incorporated, which presents a vague similarity to the human body. It is quite similar to the Giraffe robot that is used by social workers who wish to be present remotely. In this procedure Padbot had the face of one of the researchers in the display, who greeted the students. This was the only exception, but it was important that students could see how this robot moved and how it could talk to them. Joy Robot is a DIY (Do It Yourself) robot that presents a certain ability to express emotions by changing the shape of the eyes and the mouth and by moving the arms in various ways. This voluntary project was born within the maker movement in Brazil in 2016 for the purpose of developing a technology that would be able to attract communities and assist children in hospitals. Turtlebot 2 is a machine-like robot with a very low degree of similarity to humans. TurtleBot is a low-cost, personal robot kit with open-source software that has two-wheel navigation, can see in 3D, and can build maps and drive around. The robots were presented to the students in the following order: first, InMoov and then Padbot, Joy Robot and finally Turtlebot.

3.3 Procedure and Measures

We brought all of the robots to an amphitheater classroom at the University of Udine. We began with InMoov, which when the students arrived was already in the chair of the professor with two Master's students near it. The robot was facing forwards, towards the Bachelor's students, while some Master's students from the Laboratory of Social Robotics illustrated its main characteristics and functionalities (15 min). After the explanations given by the Master's students on the single robot, the Bachelor's students were invited to approach the robot at close distance to familiarize themselves with it and experience directly some forms of interaction with it. This phase lasted for 30 min. In the last 15 min, the students were invited to complete a paper and pencil questionnaire that contained both closed and open-ended questions. After the first hour, InMoov was removed and Padbot entered the room by itself, moving through the classroom and up to the chair. This was the only exception in this procedure but it was important for the students to see how this robot moved and that the tablet over its neck showed the face of one of the researchers, who greeted the students. The procedure applied with InMoov was repeated also for Padbot. After the second hour, Padbot went out of the room and Joy Robot was brought into the room. We showed the students how it was able to move its eyes, mouth and arms in an expressive way. The procedure was repeated again. After the third hour, Joy Robot was removed and Turtlebot was brought into the room and the procedure was repeated for the fourth time. Overall, the students' experience lasted 4 h.

In the questionnaire, we investigated several areas. The first regarded the conception of each robot and was addressed by assigning two tasks: 'Please write the first three words that the robot you have observed evoked in you' We decided to use this method, called a free association exercise, because it enables the capture of the spontaneous resurfacing of words elicited by the cue. This technique [6], by means of its projective character, offers the advantage of bringing out the latent and implicit dimensions of the knowledge and opinions on the specific object of the representation, whereby it allows access to the figurative core of the social representations of these four robots [62].

The second area of the questionnaire concerned the human characteristics attributed to each robot. The degree of human likeness was investigated through a series of items on the perception of mental functions adapted from D'Andrade [14], as well as on the perception of primary emotions (e.g., anger, fear, surprise, pleasure, pain, and joy) and secondary emotions (e.g., hope, love, guilt, remorse, pride, and shame) as identified by Leyens et al. [52]. Operationally, we gave the students the task: 'Please, evaluate the degree in which you think that this robot is able to…". A list of 14 items followed: hearing, seeing, smelling, tasting, desiring, needing, wanting, imagining, reasoning, thinking, choosing, deciding, expecting, planning, being conscious of oneself and the world. These items were measured on a 10-point Likert scale, ranging from not at all (1) to very much (10). The first four items described the four senses and belonged to the dimension of Perceptions, followed by the dimension of Wishes including the terms desiring, needing and wanting; the dimension of Thoughts, containing the words imagining, reasoning and thinking; and, finally, the dimension of Intentions, which included the terms choosing, deciding, expecting and planning. We also added the item 'being conscious of oneself and the world,' which marks the boundary beyond which, according to Faggin [19], a robot cannot go.

Emotions were explored by asking 'Please, evaluate the degree in which you think that this robot is able to experience the following emotions…". A list of 12 items followed: anger, fear, surprise, pleasure, pain, joy, hope, love, guilt, remorse, pride, shame' (all the items were measured on a 10-point Likert scale from 1 = not at all to 10 = very much).

In addition, we asked: 'Please, write the three most relevant emotions that this robot evoked in you' (in free answer mode). We then asked the participants to evaluate some of the emotions they felt. 'Looking at the robot that you observed, what did you feel? A sense of discomfort, a sense of eeriness, a sense of curiosity' (each measured on a 10-point Likert scale from 1 = not at all to 10 = very much).

A third area that aimed to test the familiarity of students with robots contained questions about the use of robots at home, at school and at university (with yes/no response categories).

In this study, we used qualitative (free association exercises) and quantitative methods (survey). The gathered data was analyzed by means of content analysis and multivariate analysis of variance. As to content analysis, given that the free association exercise essentially collects words and that the number of our participants was low, we opted to perform the content analysis manually. As is required in these cases, three independent judges (or coders) did the analysis separately. They then confronted the results and negotiated a shared decision on the elaboration of the categories of meaning [49]. The other statistical analyses were performed with the software SPSS Statistics 21. We will give the results in the next section.

4 Results

4.1 The Robots' Conception

The results that we obtained by means of the free association exercise on how our observers viewed the robots are shown in Table 1. Overall, for the four robots, we collected a dictionary of 721 headwords and a dictionary of 575 diverse words. After the elimination of not-classifiable words, 698 remained. These words were classified by means of content analysis into seven categories, which are the same for each robot: (1) Innovation, science and technology; (2) Negative values; (3) Positive values; (4) Humankind and the body; (5) Functions and characteristics; (6) Fiction and audio-visual products; and (7) Intelligence (Table 1).

We will briefly describe each category by reporting the most frequent words that form the core of the conveyed meaning. The category 'Innovation, science and technology' contained terms such as technology, machine, future, innovation, progress, robot, vacuum cleaner, humanoid and science. The 'Negative values' category included terms such as restlessness, eeriness, anguish, inutility, incompleteness and boredom. 'Positive values' contained the terms curiosity, interest, simplicity, utility, sweetness, sympathy, tenderness, cuteness, wonder, exceptionality, amazement and extraordinariness. The category 'Functions and characteristics' contained words such as modularity, white, small, complicate, multifunctionality, interaction, work, company, communication, driver, design, art, aesthetics, movement, videocall, practical, social, toy and game. The category 'Fiction and audio-visual products' included terms such as fake and Wall-e. The category 'Intelligence' contained artificial intelligence and intelligence. Finally, the category 'Humankind and the body' included terms containing identity, children, infancy, heart, skeleton, humanized, person and anthropomorphism.

The results of Table 1 show that the conception of these robots does not vary according to their degree of human likeness, in the sense that InMoov, the most human-like robot,

Table 1 The categories of the conception of the four robots

Categories	InMoov N%	Padbot N%	Joy Robot N%	Turtlebot N%	Total N%
Innovation, science, and technology	28 (15.7)	11 (6.3)	1 (0.6)	31 (17.9)	71 (10.2)
Negative values	69 (38.8)	34 (19.4)	13 (7.6)	36 (20.8)	152 (22.0)
Positive values	48 (27.0)	76 (43.4)	112 (65.1)	33 (19.1)	269 (39.0)
Humankind and the body	5 (2.8)	0 (0.0)	9 (5.2)	7 (4.0)	21 (3.0)
Functions and characteristics	10 (5.6)	52 (29.7)	29 (16.9)	66 (38.2)	157 (22.5)
Fiction and audio-visual products	15 (8.4)	2 (1.1)	8 (4.7)	0 (0.0)	25 (3.6)
Intelligence	3 (1.7)	0 (0.0)	0 (0.0)	0 (0.0)	3 (0.4)
Total responses	178 (100.0)	175 (100.0)	172 (100.0)	173 (100.0)	698 (100.0)

is perceived as the most negative, cheater and intelligent; Padbot is seen as quite positive and functional; Joy Robot is considered as the most positive and quite functional; and Turtlebot as the most innovative and functional. Second, the category of ‘Positive values’ receives the highest number of occurrences, which suggests that the images of these robots are mainly positive. Third, the category ‘Negative values’ shows twice the number of occurrences for InMoov (the robot most similar to humans) than Padbot and Turtlebot and a much higher number than that for Joy Robot. In this case, the highest degree of human likeness is characterized by particularly strong negative values. Fourth, the category of ‘Innovation, science and technology’ has a certain importance but not for each robot. Fifth, the category of ‘Functions and characteristics’ becomes more important when the degree of similarity to humans is lower.

4.2 Are Human Mental Functions Attributed to InMoov, Padbot, Joy Robot and Turtlebot, and, If Yes, to What Extent?

We asked the participants to evaluate if each of these robots is able to carry out the following mental functions, organized into four clusters: Perceptions (hearing, seeing, smelling and tasting), Wishes (desiring, needing and wanting), Thoughts (imagining, reasoning and thinking) and Intentions (choosing, deciding, expecting and planning) [14]. The means and standard deviations of the single functions and those of the composite scores of the four clusters that define the main dimensions of mind perception are reported in Table 2 for each robot; the means of the clusters are illustrated specifically in Fig. 2.[3]

The scores are overall rather low, the only scores that overcome the mid-point of the response scale are those related to two perceptions: the ability to hear and see.

Both the scores of each single mental function and the composite scores of each cluster were subjected to multivariate analysis of variance with one within factor (the four robots). The multivariate effect of the within factor was significant in the analysis with the single mental functions as dependent variables, $F_{42,462} = 5.05$, $p < 0.0001$, $\eta^2 p = 0.32$, and in the analysis with the four clusters as dependent variables, $F_{12,492} = 7.57$, $p < 0.0001$, $\eta^2 p = 0.16$. In Table 2 the results of the univariate tests and partial eta square are also reported. In both analyses all the univariate tests were significant. In the analysis considering the four clusters of mental functions we executed post-hoc tests with Bonferroni method. These comparisons showed that the ability to have Perceptions is attributed significantly more to InMoov than to Padbot and Turtlebot (without differences between them) and much more than to Joy Robot. Also, Wishes, Thoughts, and Intentions were attributed significantly more to InMoov than to the other three robots. Regarding Intentions, Turtlebot is in second position, at a significant distance from InMoov, while Padbot and Joy Robot receive the lower scores (without differences between them).

Finally, we examined the respondents’ evaluations on the item ‘being conscious of oneself and the world’. The mean attributed to this item is 2.73 (SD = 2.59) for InMoov, 1.65 (SD = 1.73) for Padbot, 1.39 (SD = 1.26) for Joy Robot and 2.70 (SD = 3.02) for Turtlebot. Thus, the respondents do not attribute to any of these robots the capacity of being aware of itself, which confirms Faggin’s [19] argument. However, the variance analysis with a within factor (the four robots) has highlighted the principal effect of this factor: $F_{3,180} = 8.13$, $p < 0.0001$, $\eta^2{}_p = 0.12$. That is, the two robots that are judged relatively less unconscious of themselves are InMoov and Turtlebot (without differences between them); the more unaware are Padbot and Joy Robot (again without differences between them), as shown by the comparisons between pairs that were performed.

[3] The descriptive statistics reported in Table 2 and Fig. 2 are calculated considering only the participants (n = 56) who filled all the scales of the mental functions.

Table 2 Mental functions attributed to the four robots

Mental functions	InMoov	Padbot	Joy Robot	Turtlebot	F of univariate tests	η^2_p
Perceptions	*4.46 (SD = 2.03)a*	*3.19 (SD = 1.47)b*	*2.24 (SD = 1.75)c*	*3.15 (SD = 1.60)b*	*17.56****	0.24
Hearing	6.46 (SD 2.82)	4.88 (SD = 3.34)	3.39 (SD = 3.18)	3.70 (SD = 3.29)	12.83***	0.19
Seeing	6.39 (SD 2.94)	5.79 (SD = 3.15)	3.16 (SD = 2.91)	6.43 (SD = 3.19)	16.26***	0.23
Smelling	2.91 (SD 2.35)	1.05 (SD = 0.30)	1.20 (SD = 1.21)	1.34 (SD = 1.37)	22.84***	0.29
Tasting	2.05 (SD 1.97)	1.04 (SD = 0.27)	1.20 (SD = 1.21)	1.14 (SD = 0.70)	9.82***	0.15
Wishes	*2.33 (SD = 1.53)a*	*1.50 (SD = 1.40)b*	*1.41 (SD = 1.18)b*	*1.51 (SD = 1.26)b*	*8.82****	0.14
Desiring	1.86 (SD 1.35)	1.11 (SD = 0.59)	1.32 (SD = 1.27)	1.34 (SD = 1.23)	6.18**	0.10
Needing	2.59 (SD 2.27)	1.70 (SD = 2.12)	1.61 (SD = 1.67)	1.73 (SD = 2.01)	5.27*	0.09
Wanting	2.54 (SD 2.34)	1.70(SD = 2.18)	1.30 (SD = 1.25)	1.45 (SD = 1.50)	6.81***	0.11
Thoughts	*3.51 (SD = 1.96)a*	*1.48 (SD = 1.47)b*	*1.57 (SD = 1.66)b*	*1.97(SD = 1.76)bc*	*23.07****	0.30
Imagining	1.93 (SD 1.48)	1.21 (SD = 1.22)	1.41 (SD = 1.44)	1.23 (SD = 0.93)	4.38**	0.07
Reasoning	5.13 (SD 3.26)	1.75 (SD = 1.99)	1.64 (SD = 1.86)	2.63 (SD = 2.67)	32.44***	0.37
Thinking	3.46 (SD 2.49)	1.48 (SD = 1.83)	1.64 (SD = 1.74)	2.05 (SD = 2.25)	13.27***	0.19
Intentions	*4.13 (SD = 2.45)a*	*1.63 (SD = 1.65)c*	*1.52 (SD = 1.33)c*	*2.38 (SD = 2.12)b*	*29.89****	0.35
Choosing	4.70 (SD 2.92)	1.75 (SD = 1.87)	1.57 (SD = 1.54)	2.46 (SD = 2.70)	25.97***	0.32
Deciding	4.77 (SD 3.06)	1.64 (SD = 1.77)	1.68 (SD = 1.78)	2.59 (SD = 2.65)	27.49***	0.33
Expecting	2.04 (SD 1.79)	1.43 (SD = 1.58)	1.34 (SD = 1.18)	1.54 (SD = 1.84)	2.83*	0.05
Planning	5.04 (SD 3.47)	1.71 (SD = 2.09)	1.48 (SD = 1.31)	2.95 (SD = 2.98)	28.90***	0.34

The answers were given on a 10-point Likert scale, where 1 = not at all and 10 = very much. The asterisks have the following meaning: *$p < 0.05$; ** $p < 0.01$; ***$p < 0.001$. The means marked with different letters (a, b, c) are significantly different among them at least with $p < 0.05$ with Bonferroni method

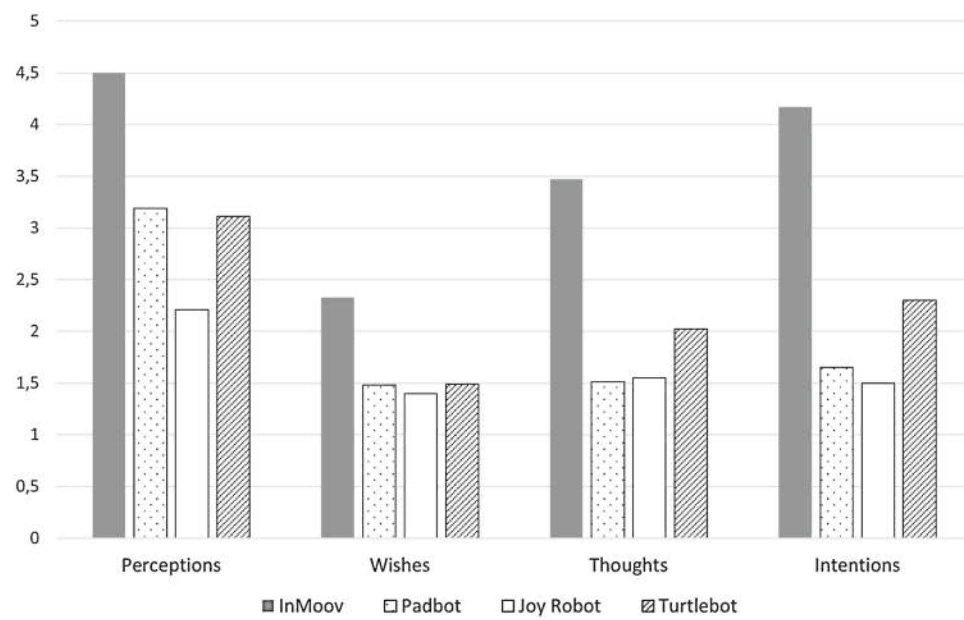


Fig. 2 The means of the four clusters of mental functions attributed to InMoov, Padbot, Joy Robot and Turtlebot

4.3 Are Emotions Attributed to InMoov, Padbot, Joy Robot and Turtlebot and, If Yes, to What Extent?

We asked the participants to evaluate if these robots were able to experience a list of Primary emotions (anger, fear, surprise, pleasure, pain and joy) and Secondary emotions (hope, love, guilt, remorse, pride and shame) that were proposed to them. The means and standard deviations of the single emotion and those of the two clusters—Primary emotions and Secondary emotions- are reported in Table 3 for each robot and illustrated in Fig. 3.

Table 3 reports also the results of the univariate tests of multivariate analyses of variance (the values of F and partial eta square).

These mean scores show that the respondents think that the four robots are fundamentally incapable of feeling emotions.

Table 3 Primary and secondary emotions attributed to InMoov, Padbot, Joy Robot and Turtlebot

Emotions	InMoov	Padbot	Joy Robot	Turtlebot	F of univariate tests	$\eta^2{}_p$
Primary emotions	*1.81 (SD = 1.15)a*	*1.08 (SD = 0.34) b*	*1.86 (SD = 1.77) a*	*1.13 (SD = 0.65)b*	*10.17****	*0.14*
Anger	1.71 (SD 1.45)	1.03 (SD = 0.25)	1.68 (SD = 1.72)	1.11 (SD = 0.63)	6.09**	0.09
Fear	1.60 (SD 1.18)	1.03 (SD = 0.25)	1.68 (SD = 1.72)	1.11 (SD = 0.68)	5.88**	0.09
Surprise	1.81 (SD 1.35)	1.06 (SD = 0.31)	1.94 (SD = 2.06)	1.18 (SD = 0.80)	8.03***	0.12
Pleasure	1.95 (SD 1.69)	1.16 (SD = 0.81)	2.13 (SD = 2.30)	1.15 (SD = 0.74)	8.92***	0.13
Pain	1.76 (SD 1.54)	1.05 (SD = 0.28)	1.60 (SD = 1.54)	1.11 (SD = .58)	5.99**	0.09
Joy	2.06 (SD 1.83)	1.11 (SD = 0.77)	2.15 (SD = 2.30)	1.11 (SD = 0.68)	10.06***	0.14
Secondary emotions	*1.55 (SD = 0.94) a*	*1.03 (SD = .014) b*	*1.60 (SD = 1.31) a*	*1.11 (SD = 0.50)b*	*8.73****	*0.13*
Hope	1.47 (SD 1.04)	1.02 (SD = 0.13)	1.79 (SD = 2.04)	1.13 (SD = 0.71)	6.08**	0.09
Love	1.50 (SD 1.36)	1.02 (SD = 0.13)	1.82 (SD = 1.78)	1.15 (SD = 0.78)	6.02**	0.09
Guilt	1.53 (SD 1.48)	1.03 (SD = 0.18)	1.53 (SD = 1.32)	1.11 (SD = 0.68)	4.55**	0.07
Remorse	1.50 (SD 1.39)	1.02 (SD = 0.13)	1.40 (SD = 1.06)	1.10 (SD = 0.53)	4.26**	0.07
Pride	1.87 (SD 1.70)	1.03 (SD = 0.18)	1.65 (SD = 1.68)	1.13 (SD = 0.78)	7.73***	0.11
Shame	1.42 (SD 0.95)	1.05 (SD = 0.28)	1.40 (SD = 1.06)	1.05 (SD = 0.28)	5.16**	0.08

The answers were given on a 10-point Likert scale, where 1 = not at all and 10 = very much. The asterisks have the following meaning: *$p < 0.05$; ** $p < 0.01$; ***$p < 0.001$. The means marked with different letters (a, b, c) are significantly different among them at least with $p < 0.05$, with Bonferroni method

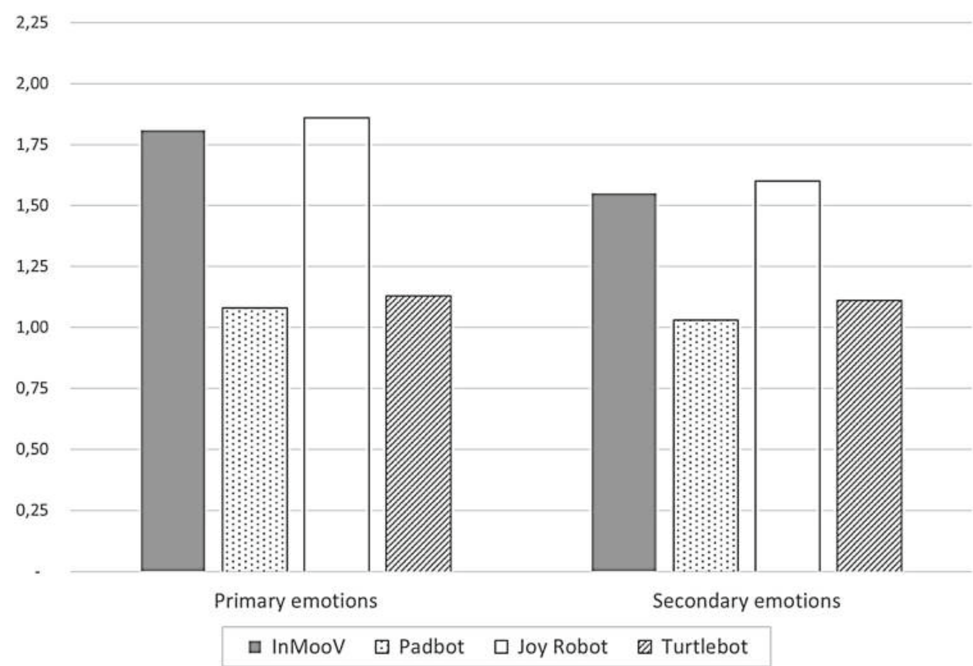


Fig. 3 The means of the two clusters of Primary and Secondary emotions attributed to InMoov, Padbot, Joy Robot and Turtlebot

There are, however, some differences. The multivariate analyses of variance with one within factor (the four robots) and the single emotions as dependent variables showed that the multivariate effect of the within factor was significant: $F_{36,522} = 1.61$, $p < 0.02$, $\eta^2{}_p = 0.10$, as well as all of the univariate tests. Also for the two clusters—Primary and Secondary emotions- the multivariate effect of the within factor was significant: $F_{6,366} = 4.90$, $p < 0.0001$, $\eta^2{}_p = 0.07$; as well as the two univariate effects. The post-hoc analyzes with Bonferroni method showed that Primary emotions were attributed a little more (but significantly) to InMoov and to Joy Robot (without differences between them). The situation is similar with regards to Secondary emotions: the post-hoc comparisons showed that respondents attributed to InMoov and to Joy Robot a little more (with significant) capacity to feel secondary emotions, which are those typical of human beings.

4.4 The Emotions Evoked by these Robots and the Uncanny Valley Effect

4.4.1 Free Association Exercises Regarding the Emotions that InMoov, Padbot, Joy Robot and Turtlebot Evoked on the Respondents

After exploring the emotions that, according to our respondents, the four robots could experience, we investigated the spontaneous emergence of the emotions that InMoov, Padbot, Joy Robot and Turtlebot aroused in our participants. In our study from the free association exercise, we collected an overall dictionary of 587 headwords and a dictionary of 172 diverse words. After removing some not classifiable words, we conducted a content analysis of the remaining words, which could be traced back to the twelve previously used categories (Table 4). These categories express the main constellations of emotions that emerged from the emotional continuum gathered around these twelve emotions. 'Anger' includes only anger. 'Surprise' comprises curiosity, wonder, surprise and interest. 'Pain' includes words such as restlessness, anxiety, discomfort, disorientation, insecurity, indifference, delusion, boredom, doubt and sadness. 'Fear' is made of the word fear. 'Pleasure' includes pleasure, serenity, calm, amusement, sympathy, empathy, enthusiasm and tenderness. 'Joy' contains terms like happiness and joy. 'Love' contains words such as love, fondness, intimacy and affection. 'Hope' and 'Pride' include only hope and pride. Remorse, guilt and shame are totally absent; others, such as anger, are almost absent while others are very much present.

The main category of emotions evoked by our robots is surprise (N = 210), the second is pleasure (N = 162), followed by pain (N = 115), with fear almost being irrelevant. As to the single robots, InMoov evokes especially pain and surprise (mostly equivalent) and, at distance, pleasure. Padbot arouses pleasure, followed by surprise and pain. Joy Robot evokes, first, pleasure, followed by surprise and joy. Turtlebot arouses surprise, followed by pain and pleasure. If we compare the robots to each other, we see that Joy Robot is unbeatable in provoking joy, Turtlebot is the robot who is able to surprise students the most, InMoov evokes more pain than any other robot and Padbot is more balanced in the emotions that it provokes.

4.4.2 Evaluation of the Discomfort, Eeriness, and Curiosity Evoked in the Participants

We further investigated this question of the emotions felt by the participants towards these robots to see if there exists a gradualness of intensity of three specific emotions (i.e., discomfort, eeriness and curiosity) corresponding to the various degrees of these robots' human likeness. The means, standard deviations and the results of variance analyses are reported in Table 5.

These results reveal that the degree of discomfort and eeriness towards these four robots is quite low (under the mid-point of the response scale), while curiosity seems to be the dominant feeling. The analyses of variance with a within factor showed that the judgments on the four robots were different. The post-hoc paired comparisons revealed that the participants felt both the highest sense of discomfort and eeriness but also of curiosity towards InMoov as compared to the other robots. Furthermore, another series of comparisons conducted with paired-samples t-test showed that, for all four robots, discomfort is not different from eeriness, while discomfort and eeriness are significantly lower than curiosity (t_s are above 5.91, $p < 0.0001$).

5 Discussion and Final Remarks

The first result of this study is that, generally, our students attributed very low scores to these robots regarding their mental functions and even lower regarding the emotions. Our overall results do not differ much from those obtained by Haslam et al. [34] because the "folk model" of the mind and emotions of robots shows only a modest change. In particular, regarding the mental functions, the respondents attributed two perceptions to these robots: the ability to hear and the ability to see, which is perfectly understandable given the level reached by current sensors. In particular, the ability to hear is recognized only for InMoov, while the ability to see is attributed to InMoov, to Turtlebot and, at distance, to Padbot. For the rest, the other mental functions that approach the mid-point of the response scales without reaching it are attributed to InMoov, and concern reasoning and planning. Regarding the attribution of emotions, the students' evaluations are even more severe—these robots are fundamentally perceived as incapable of feeling emotions.

We will use our research questions as a guide to structure further the discussion of the results. We will begin by providing an answer to RQ1. "Is the attribution (or not attribution) of mental and emotional capabilities to each of these four robots modulated according to their decreasing degrees of morphological and behavioral similarity to humans?" The first result is that students do not attribute mental functions and emotions to these robots, modulating their attribution according to their different human likeness. Thus, our H1 (i.e., the more a robot resembles humans, the more it is perceived with mental and emotional capabilities) does not find confirmation. The point is not the gradualness of the similarity to humans, but that only InMoov, which is the most human-like robot, was very partially assimilated to human mental functions. Any appearance of a robot which is not strongly similar to humans does not receive a recognition

Table 4 Emotions that emerged from the free association exercise as evoked in the participants by the four robots

Emotional categories	InMoov N (%)	Padbot N (%)	Joy Robot N (%)	Turtlebot N (%)	Total N (%)
Anger	0 (0.0)	1 (0.9)	0 (0.0)	0 (0.0)	1 (0.2)
Fear	10 (6.8)	0 (0.0)	0 (0.0)	1 (0.8)	11 (2.0)
Surprise	57 (38.5)	36 (33.0)	30 (18.6)	87 (71.9)	210 (39.0)
Pleasure	17 (11.5)	39 (35.8)	95 (59.0)	11 (9.1)	162 (30.1)
Pain	58 (39.2)	26 (23.9)	13 (8.1)	18 (14.9)	115 (21.3)
Joy	3 (2.0)	4 (3.7)	16 (9.9)	3 (2.5)	26 (4.8)
Hope	3 (2.0)	1 (0.9)	1 (0.6)	1 (0.8)	6 (1.1)
Love	0 (0.0)	2 (1.8)	5 (3.1)	0 (0.0)	7 (1.3)
Pride	0 (0.0)	0 (0.0)	1 (0.6)	0 (0.0)	1 (0.2)
Total	148 (100.0)	109 (100.0)	161 (100.0)	121 (100.0)	539 (100.0)

Table 5 Feelings evoked by InMoov, Padbot, Joy Robot and Turtlebot

Emotion	InMoov	Padbot	Joy Robot	Turtlebot	$F_{(3,177)}$	η^2_p
Sense of discomfort	5.05a (SD = 2.87)	2.48b (SD = 2.28)	2.13b (SD = 2.29)	2.50b (SD = 2.24)	28.85***	0.33
Sense of eeriness	4.58a (SD = 2.87)	2.39b (SD = 2.35)	2.05b (SD = 2.09)	2.23b (SD = 1.88)	17.82***	0.23
Sense of curiosity	7.56a (SD = 2.05)	5.06b (SD = 2.75)	5.56b (SD = 2.83)	5.13b (SD = 2.89)	16.29***	0.22

The answers were given on a 10-point Likert scale, where 1 = not at all and 10 = very much. The asterisks have the following meaning: *$p < 0.05$; ** $p < 0.01$; ***$p < 0.001$. The means marked with different letters (a, b, c) are significantly different among them at least with $p < 0.05$

of any mental function from the participants. This confirms the split described earlier and contrasts with research such as that by Krach et al. [48], who found that the more a robot exhibits human-like features, the more people build a model of its "mind."

We will next answer to RQ2: "Is the morphological similarity of a robot to humans perceived as such by observers?" This study has found that the morphological similarity of a robot to humans is not automatically perceived as such by observers and thus our hypothesis (H2) here finds a confirmation. In the figurative nucleus of the students' spontaneous conceptions of these robots, the robots' material body is in fact completely absent. This means that our participants have elaborated a disembodied image of these robots. The representations of these robots differ greatly from that described in previous research regarding robots [24], as well as from that describing information and communication technologies at the beginning of their diffusion [23]. In these studies, one of the most important dimensions of their representations was the material "body." Thus, the human body is no longer a point of reference or comparison, whereby these robots are not conceptualized in a process leading back to the similarity to, or dissimilarity from, human beings. They are instead perceived as different entities and their intelligence is not a true issue. Thus robots are perceived not on the base of their morphological similarity to humans but beyond this. This result remains unexplained at the moment because we are unable to understand the reasons behind the fact that the current mental schemes for the robots have introjected disembodied robots. Furthermore, it is important to stress that there is no unique structure of the nuclei of the figurative representation of robots, but rather each robot has its own structure. This makes it impossible to detect a gradualness of the representation that corresponds to the gradualness of these four robots' human likeness. There is, instead, a caesura between InMoov and the others; in the sense that InMoov, the most human-like robot, is perceived in the most negative way. This result would suggest that a high similarity to humans is considered a disvalue for a robot, which is a real problem that engulfs any interest in its affordances. With InMoov, people are, in fact, not interested in understanding what it can do but only in defining what it is in negative terms. Last but not least, this robot also presents some problems of fictionality in the sense that it is perceived as pretending to be a human.

Finally, we will answer to RQ3: "Does the effect of the uncanny valley arise even at lower levels of robot–human likeness?" An unexpected result comes from the category of emotions which have most frequently been evoked in the participants by our four robots—InMoov, the robot that is most similar to humans is perturbing because it evokes pain (e.g. anxiety, disorientation, insecurity) more than any other robot. This proves that even robots with a medium–high level of similarity to humans (n − 3) like InMoov can generate the uncanny valley effect, somehow confirming the study conducted by Kim et al. [46] in which they discovered an additional valley, connected to robots with low to moderate human likeness. Consequently, our H3, according to which robots with a medium human-likeness (n.3) are unable to generate the uncanny valley effect equally, does not find confirmation. A second result that comes from our exploration of the sense of discomfort, eeriness and curiosity felt towards these robots is that our participants felt towards InMoov, in comparison to the other robots, both the highest sense of discomfort and eeriness. This result is in line with the previous result. However, this second analysis adds another important finding: that the feeling of the uncanny valley is moderated in this case by the feeling of curiosity which is stronger. While discomfort and eeriness are under the mid-point, curiosity seems to be the dominant feeling. This result leads us to conclude also that a gradual intensity of emotional discomfort and eeriness that corresponds to the various degrees of these robots human likeness does not exist, but there is a contraposition between the humanoid InMoov and all the other robots whose similarity to humans is insufficient to arouse the uncanny valley effect.

The strong points of this study are first that it has used four physical robots to empirically measure both the live evaluation of their mental functions and emotions, as well as the degree to which students emotionally reacted towards these robots by obtaining a series of clear results. Second, it has matched quantitative and qualitative methods. There are several limitations of this study. First, this study was limited by the small number of participants forming a convenience sample. Second, because we used real robots (not pictures or videos), we had to limit their number. The practical exercise with the four robots lasted four hours and it would have been impossible to extend it any further even though only a few robots were used. Third, we had to choose among the robots that the laboratory of the university had, which lacked a more advanced humanoid. This was a major drawback for our analysis based on the gradualness of robot similarity to humans. Of these four robots, only InMoov and Padbot are included in the ABOT database, where they show a human-likeness score respectively of 59 and 4.13. However, the procedure of the human-likeness score applied in the ABOT Database is very specific and performed online with a picture of the robot, while we decided to show real, physical robots. In the present study, the fact that Padbot and Joy Robot belong to the same level n-1 of the Eaton's taxonomy can represent a problem, although this layer is so large that it includes very different robots. In our case, Padbot, which has a long neck, a face-like screen over this neck where the real face of the operator can be incorporated, can talk and has the ability to move, presents a certain similarity to the human body and for this reason it is situated at the highest level of the layer. Joy Robot, which is similar to a doll but with expressive eyes and mouth and with arms that it can move, is at a lower level of the layer n-1. Fourth, in this research it was not possible to control differences between these four robots, which could interfere with the objectives of this research and the tasks assigned to the students. For example, the evaluations of similarity to humans attributed to InMoov could have been favored by the fact that it was the only robot that was dressed somehow since it was presented wearing a shirt, while the other robots hadn't such a degree of anthropomorphization to make sense a cloth over them. Cultural cues, like clothes, are often used to increase anthropomorphization in addition to biological cues. Sometimes, dressing cues are part of the physical body of robots like, for example, in the case of the robots Doro (which has a bonnet) and Coro (which has a tie); other times robots are dressed with actual clothing items. But, while it makes sense to dress an anthropomorphized robot, it can be ridiculous to do this with robots more vaguely similar to humans. For this reason, only InMoov was dressed. Fifth, it was not possible to randomize the order in which we presented the robots because the participants had to be present in the classroom at the same time. Sixth, we cannot guarantee that all of the students experienced exactly the same interactions with the robots during the half an hour that they approached each robot and interacted with. This is a limitation of the study because any difference among interactions could affect the participants' perception towards the robots while each interaction should be comparable with the others. Finally, the participants were unfortunately not very familiar with actual robots and this has affected their evaluations. For future studies, this research at least recommends to involve a larger sample of participants and to add robots with a higher degree of human likeness, such as androids/gynoids.

Funding Open access funding provided by Università degli Studi di Udine within the CRUI-CARE Agreement. The authors received no financial support for the research, authorship, and/or publication of this article.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Autio E, Mudambi R, Yoo Y (2020) Digitalization and globalization in a turbulent world: centrifugal and centripetal forces. Glob Strateg J 11:3–16
2. Baindridge WA, Hart JW, Kim ES, Scassellati B (2011) The benefits of interactions with physically present robots over video-displayed agents. Int J Soc Robotics 3:41–52
3. Banks J, de Graaf MMA (2020) Toward an agent-agnostic transmission model: Synthesizing anthropocentric and technocentric paradigms in communication. Hum Mach Commun 1:19–36
4. Bartneck C, Kanda T, Ishiguro H, Hagita N (2007) Is the uncanny valley an uncanny cliff? In: Proceedings of the 16th IEEE international workshop on robot and human interactive communication. RO-MAN 2007, pp 368–373
5. Barut Tugtekin E, Koc M (2020) Understanding the relationship between new media literacy, communication skills, and democratic tendency: model development and testing. New Media Soc 22(10):1922–1941
6. Bellelli G (1990) La tecnica delle associazioni libere nello studio delle rappresentazioni sociali. Aspetti cognitivi e linguistici. Rassegna di Psicologia 7:17–30
7. Berry DS, McArthur LZ (1985) Some components and consequences of a baby face. J Pers Soc Psychol 48:312–323
8. Broadbent E, Tamagawa R, Kerse NM, Knock BW, Patience AA, MacDonald BA (2009) Retirement home staff and residents' preferences for healthcare robots. In: Proceedings of IEEE international symposium on robot and human interactive communication 18th, Toyama, Japan, Sep 27–Oct 2. IEEE, Piscataway, pp 645–650
9. Broadbent E (2017) Interactions with robots: the truths we reveal about ourselves. Annu Rev Psychol 68:627–652
10. Burleigh TJ, Schoenherr JR, Lacroix GL (2013) Does the uncanny valley exist? An empirical test of the relationship between eeriness and the human likeness of digitally created faces. Comput Hum Behav 29(3):759–771
11. Carpenter J, Davis J, Erwin-Stewart N, Lee T, Bransford J, Vye N (2009) Gender representation and humanoid robots designed for domestic use. Int J Soc Robot 1(3):261–265
12. Damiani P, Grimaldi R, Palmieri S (2013) Robotica educativa e aspetti non verbali nei Disturbi Specifici di Apprendimento 1:1211–1220. https://iris.unito.it/handle/2318/134850#.XG_jvOhKjOg
13. Damiano L, Dumouchel P (2017) Living with robots. Emerging issues on the psychological and social implications of robotics. Harvard University Press, Cambridge, Mass
14. D'Andrade R (1987) The folk model of the mind. In: Holland D, Quinn N (eds) Cultural models in language and thought. Cambridge University Press, New York, pp 112–148
15. Destephe M, Brandao M, Kishi T et al (2015) Walking in the uncanny valley: importance of the attractiveness on the acceptance of a robot as a working partner. Front Psychol. https://doi.org/10.3389/fpsyg.2015.00204
16. DiSalvo CF, Gemperle F, Forlizzi J, Kiesler SB (2002) All robots are not created equal: the design and perception of humanoid robot heads. In: Proceedings of the designing interactive systems 2002 conference, pp 321–326
17. Eaton M (2015) Evolutionary humanoid robotics. Springer, New York
18. Edwards AP (2018) Animals, humans, and machines: Interactive implications of ontological classification. In: Guzman A (ed) Human-machine communication: rethinking communication, technology, and ourselves. Peter Lang, Bern, pp 29–50
19. Faggin F (2019) Silicio. Mondadori, Milano
20. Fortunati L (2013) Afterword: robot conceptualizations between continuity and innovation. intervalla: platform for intellectual exchange, 1, Social Robots and Emotion: Transcending the Boundaries between Humans and ICTs, pp 116–129. ISSN 2296-3413. https://www.fus.edu/intervalla-files/0_intervalla_volume1_fullmanuscript_ss100513_final.pdf
21. Fortunati L (2018) Robotization and the domestic sphere. New Media Soc 20(8):2673–2690. https://doi.org/10.1177/1461444817729366
22. Fortunati L, Edwards A (2020) Opening space for theoretical, methodological, and empirical issues. Human-Machine Comm 1:7–18. https://doi.org/10.30658/hmc.1.1
23. Fortunati L, Manganelli AM (2007) The social representations of telecommunications. Pers Ubiquitous Comput 12(6):421–431. https://doi.org/10.1007/s00779-006-0139-7
24. Fortunati L, Esposito A, Sarrica M, Ferrin G (2015) Children's knowledge and imaginary about robots. Int J Soc Robot, 7(5):685–695. https://doi.org/10.1007/s12369-015-0316-9
25. Foucault Welles B, González-Bailón S (2020) The Oxford handbook of networked communication. Oxford University Press, Oxford
26. Freud S (1970) Das Unheimliche. In: Freud S (ed) Psychologische Schriften. Studienausgabe. Frankfurt am Main: Fischer, pp 241–274. (Original work published 1919)
27. Giannopulu I, Terada K, Watanabe T (2018) Communication using robots: a Perception-action scenario in moderate ASD. J Exp Theor Artif Intell 30(5):603–613
28. Gray HM, Gray K, Wegner D (2007) Dimensions of mind perception. Science 315(5812):619–619
29. Gray K, Wegner DM (2012) Feeling robots and human zombies: mind perception and the uncanny valley. Cognition 125(1):125–130. https://doi.org/10.1016/j.cognition.2012.06.007
30. Guzman A (ed) (2018) Human-machine communication: rethinking communication, technology, & ourselves. Peter Lang, Berlin
31. Hanson D (2006) Exploring the aesthetic range for humanoid robots. In: Proceedings of the Cog Sci 2006 workshop on android, science, pp 16–20
32. Hanson D, Olney A, Pereira IA, Zielke M (2005) Upending the uncanny valley. Proc Natl Conf Artif Intell 20(4):1728–1729
33. Haring KS, Silvera-Tawil D, Watanabe K, Velonaki M (2016) The influence of robot' appearance and interactivity ability in HRI: a cross-cultural study. In: Agah A, Cabibihan J-J, Howard AM, Salichs MA, He H (eds) Social robotics. Lecture Notes in Artificial Intelligence (LNAI), vol 9979. Springer, Berlin, pp 392–401
34. Haslam N, Kashima Y, Loughnan S et al (2008) Subhuman, inhuman, and superhuman: contrasting humans with nonhumans in three cultures. Soc Cogn 26(2):248–258
35. Haslam N, Loughnan S, Kashima Y et al (2008) Attributing and denying humanness to others. Eur Rev Soc Psychol 19(1):55–85
36. Hegel F, Krach S, Kircher T, Wrede B, Sagerer G (2008a) Theory of mind (ToM) on robots: A functional neuroimaging study. In: Fong T, Dautenhahn K, Scheutz M, Demiris Y (eds) Proceedings of the 3rd ACM/IEEE international conference on human robot interaction (HRI'08). ACM, Amsterdam, pp 335–342
37. Hegel F, Krach S, Kircher T, Wrede B, Sagerer G (2008b) Understanding social robots: a user study on anthropomorphism. In: Proceedings of the 17th IEEE international symposium on robot and human interactive communication. ROMAN 2008, pp 574–579
38. Hennig-Thurau T, Houston MB (2019) The fundamentals of entertainment. In: Entertainment science. Data analytics and practical theory for movies, games, books, and music. Springer, Cham, pp 41–57. https://doi.org/10.1007/978-3-319-89292-4_2

39. Hinds P, Roberts T, Jones H (2004) Whose job is it anyway? A study of human–robot interaction in a collaborative task. Hum Comput Interact 19(1):151–181
40. Ho CC, MacDorman KF (2017) Measuring the uncanny valley effect: refinements to indices for perceived humanness, attractiveness, and eeriness. Int J Soc Robotics 9(1):129–139
41. Höflich JR (2013) Social robotic experience and media communication practices: an exploration on the emotional and ritualized human-technology-relations. intervalla: platform for intellectual exchange, 1, Social Robots and Emotion: Transcending the Boundaries between Humans and ICTs, pp 49–59. ISSN 2296-3413. https://www.fus.edu/intervalla-files/0_intervalla_volume1_fullmanuscript_ss100513_final.pdf
42. Jentsch E (1906) Zur Psychologie des Unheimlichen. Psychiatrisch-Neurologische Wochenschrift 22:195–198; 23:203–205
43. Jung Y, Cho E (2018) Context-specific affective and cognitive responses to humanoid robots. In: The 22nd biennial conference of the international telecommunications society: "beyond the boundaries: Challenges for business, policy and society", International Telecommunications Society (ITS), Seoul.
44. Kanda T, Miyashita T, Osada T, Haikawa Y, Ishiguro H (2008) Analysis of humanoid appearances in human–robot interaction. IEEE Trans Robtics 24(3):725–735
45. Kätsyri J, Förger K, Mäkäräinen M, Takala T (2015) A review of empirical evidence on different uncanny valley hypotheses: support for perceptual mismatch as one road to the valley of eeriness. Front Psychol. https://doi.org/10.3389/fpsyg.2015.00390
46. Kim B, Bruce M, Brown L, de Visser E, Phillips EK (2020) A comprehensive approach to validating the uncanny valley using the anthropomorphic RoBOT (ABOT) database. https://www.researchgate.net/publication/341835020_A_Comprehensive_Approach_to_Validating_the_Uncanny_Valley_using_the_Anthropomorphic_RoBOT_ABOT_Database. Accessed 8 Nov 2020
47. Komatsu T, Yamada S (2007) Effects of robotic agents' appearances on users' interpretations of the agents' attitudes: towards an expansion of uncanny valley assumption. In: RO-MAN'07. Proceedings of the 14th IEEE international symposium on robot and human interactive communication. IEEE Press
48. Krach S, Hegel F, Wrede B, Sagerer G, Binkofski F, Kircher T et al (2008) Can machines think? Interaction and perspective taking with robots investigated via fMRI. PLoS ONE 3(7):e2597
49. Krippendorff K (2012) Content analysis: an introduction to its methodology. Sage, Thousand Oaks
50. Küster D, Swiderska A, Gunkel D (2020). I saw it on YouTube! How online videos shape perceptions of mind, morality, and fears about robots. New Media Soc 1–20
51. Lay S, Brace N, Pike G, Pollick F (2016) Circling around the uncanny valley: design principles for research into the relation between human likeness and eeriness. i-Perception 7(6):1–11
52. Leyens JPh, Demoulin S, Vaes J et al (2007) Infra-humanization: the wall of group differences. Soc Issues Policy Rev 1(1):139–172
53. Li D, Rau PLP, Li P (2010) A cross-cultural study: effect of robot appearance and task. Int J Soc Robotics 2(2):175–186
54. Li J (2015) The benefit of being physically present: a survey of experimental works comparing copresent robots, telepresent robots and virtual agents. Int J Hum Comput Stud 77:23–37
55. Lievrouw LA, Livingstone S (eds) (2012) The handbook of new media. Sage, London
56. Lischetzke T, Izydorczyk D, Hüller C, Appel M (2017) The topography of the Uncanny Valley and individuals' need for structure: a nonlinear mixed effects analysis. J Res Pers 68:96–113
57. Lohse M, Hegel F, Swadzba A, Rohlfing KJ, Wachsmuth S, Wrede B (2007) What can I do for you? Appearance and application of robots. In: Proceedings of the reign of catz and dogz? The role of virtual creatures in a computerised society, symposium at AISB'07, pp 121–126
58. MacDorman KF (2006) Subjective ratings of robot video clips for human likeness, familarity, and eeriness: an exploration of the uncanny valley. In: Proceedings of the Cog Sci 2006 workshop on android, science, pp 26–29
59. MacDorman K, Ishiguro H (2006) The uncanny advantage of using androids in social and cognitive science research. Interact Stud 7(3):297–337
60. McGinn C (2020) Why Do Robots Need a Head? The role of social interfaces on service robots. Int J Soc Robotics 12:281–295
61. Mori M (1970) The uncanny valley. Energy 7(49):33–35. (Trans. MacDorman, K. F. & Minato, T.). http://www.movingimages.info/digitalmedia/wp-content/uploads/2010/06/MorUnc.pdf
62. Moscovici S (1961) La psychanalyse, son image et son public. P.U.F, Paris
63. Musik C, Bogner A (eds) (2019) Digitalization & society. A sociology of technology perspective on current trends in data, digital security and the internet. Springer, Wiesbaden
64. Oberman LM, McCleery JP, Ramachandran VS et al (2007) EEG evidence for mirror neuron activity during the observation of human and robot actions: towards an analysis of the human qualities of interactive robots. Neurocomputing 70(13):2194–2203
65. Papacharissi Z, Mendelson AL (2007) An exploratory study of reality appeal: uses and gratifications of reality TV shows. J Broadcast Electron Media 51(2):355–370
66. Powers A, Kiesler S (2006) The advisor found robot: tracing people's mental model from a robot's physical attributes. In: Goodrich MA (ed) Proceedings of the 1st ACM SIGCHI/SIGART conference on human-robot interaction. ACM, New York, pp 218–225
67. Riek LD, Rabinowitch T-C, Chakrabarti B, Robinson P (2009a) How anthropomorphism affects empathy toward robots. In: Proceedings of the 4th ACM/IEEE international conference on Human robot interaction (HRI'09), La Jolla, California, USA, March 09–13, 2009, pp 245–246
68. Riek LD, Rabinowitch T-C, Chakrabartiz B, Robinson P (2009b) Empathizing with robots: fellow feeling along the anthropomorphic spectrum. In: Proceedings of the IEEE conference on affective computing and intelligent interaction (ACII'09), pp 43–48
69. Rosenthal-von der Pütten AM, Krämer NC, Hoffmann L, Sobieraj S, Eimler SC (2013) An experimental study on emotional reactions towards a robot. Int J Soc Robotics 5(1):17–34
70. Rosenthal-von der Pütten AM, Krämer NC (2014) How design characteristics of robots determine evaluation and uncanny valley related responses. Comput Hum Behav 36:422–443
71. Stein J-P, Ohler P (2017) Venturing into the uncanny valley of mind—the influence of mind attribution on the acceptance of human-like characters in a virtual reality setting. Cognition 160:43–50
72. Su CC, Liu J, Zhou B (2020) Comparative media studies in the digital ageltwo levels of digitalization and internet use across Europe, China, and the US. Int J Commun 14:5838–5859
73. Tinwell A, Grimshaw MN, Williams A (2011) The uncanny wall. Int J Arts Technol 4(3):326–341
74. Valentí Soler M, Agüera-Ortiz L, Olazarán Rodríguez J et al (2015) Social robots in advanced dementia. Front Ageing Neurosci. https://doi.org/10.3389/fnagi.2015.00133
75. Villacampa J, Ingram G, Corradi G, Olivera-La Rosa A (2019) Applying an implicit approach to research on the uncanny feeling. J Articles Support Null Hypothesis 16(1):11–22

76. Wainer J, Feil-Seifer DJ, Shell DA, Mataric MJ (2006) The role of physical embodiment in human–robot interaction. In: The 15th IEEE international symposium on robot and human interactive communication (RO-MAN06), Hatfield, UK, September 6–8
77. Wang S, Lilienfeld SO, Rochat P (2015) The uncanny valley: existence and explanations. Rev Gen Psychol 19(4):393–407
78. Woods SN (2006) Exploring the design space of robots: children's perspectives. Interact Comput 18(6):1390–1418. https://doi.org/10.1016/j.intcom.2006.05.001
79. Zhang J, Li S, Zhang JY, Du F, Qi Y, Liu X (2020) A literature review of the research on the uncanny valley. In: International conference on human–computer interaction, HCII 2020: cross-cultural design. User experience of products, services, and intelligent environments, pp 255–268

Androids as an Experimental Apparatus: Why Is There an Uncanny Valley and Can We Exploit It?

Karl F. MacDorman
Department of Adaptive Machine Systems and Frontier Research Center
Graduate School of Engineering
Osaka University, Suita, Osaka 565-0871 Japan

Abstract

Abstract. It seems natural to assume that the more closely robots come to resemble people, the more likely they are to elicit the kinds of responses people direct toward each other. However, subtle flaws in appearance and movement only seem eerie in very humanlike robots. This uncanny phenomenon may be symptomatic of entities that elicit a model of a human other but do not measure up to it. If so, a very humanlike robot may provide the best means of finding out what kinds of behavior are perceived as human, since deviations from a human other are more obvious. In pursuing this line of inquiry, it is essential to identify the mechanisms involved in evaluations of human likeness. One hypothesis is that an uncanny robot elicits an innate fear of death and culturally-supported defenses for coping with death's inevitability. An experiment, which borrows from the methods of terror management research, was performed to test this hypothesis. Across all questions subjects who were exposed to a still image of an uncanny humanlike robot had on average a heightened preference for worldview supporters and a diminished preference for worldview threats relative to the control group.

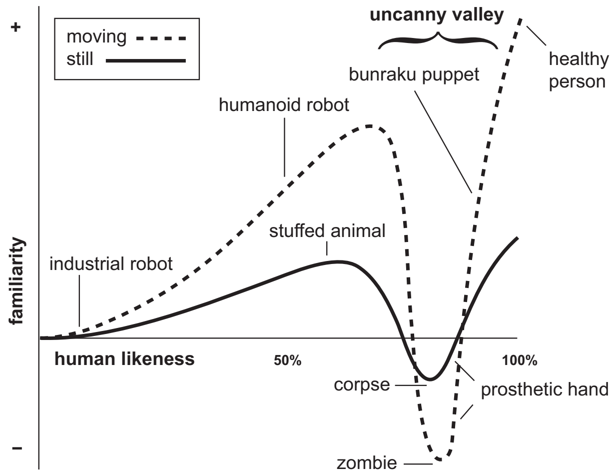


Figure 1: As a robot designer, Mori graphed what he saw as the relation between human likeness and perceived familiarity: familiarity increases with human likeness until a point is reached at which subtle differences in appearance and behavior create an unnerving effect [Mori, 1970]. This he called the *uncanny valley.* According to Mori, movement amplifies the effect.

Introduction

An experimental apparatus that is indistinguishable from a human being, at least superficially, has the potential to contribute greatly to an understanding of face-to-face interaction in the social and neurosciences. Such a device could be a perfect actor in controlled experiments, permitting scientists to vary precisely the parameters under study. It could also serve as a testbed for cognitive theories, including theories about how the brain acts as a control system in mediating whole-bodied communication. The device would also have the advantage of having the physical presence that simulated characters lack. Unfortunately, no such device yet exists, nor will one any time soon; nevertheless, robots are being built that with each new generation more closely simulate human beings in appearance, facial expression, and gesture [Minato et al., 2004] [MacDorman et al., 2005] [Matsui et al., 2005]. They are capable of eliciting some of the kinds of responses that people direct toward each other but not toward mechanical-looking robots.

Humanlike robots, often referred to as *androids* in the robotics literature to distinguish them from mechanical-looking humanoid robots, may prove more capable of eliciting a subject's model of a human other than any other contrivance to date. One apparent symptom of their potential for eliciting human-directed responses is a phenomenon Masahiro Mori identified as the *uncanny valley* [Mori, 1970].

Mori predicted that, as robots appear more human, they seem more familiar until a point is reached at which subtle imperfections create a sensation of strangeness (see Fig. 1 and Appendix B). He noted that some prosthetic hands are, at first glance, indistinguishable from real hands. They simulate muscles, tendons, veins, skin pigmentation, fingernails, and even finger prints. However, if you shook one, the lack of soft tissue and cold temperature would give you a shock. The fact that these hands can move automatically only increases the sensation of strangeness (as shown by the dashed line in Fig. 1). To build a complete android, Mori believed, would only multiply this eerie feeling many times over: Machines that appeared too lifelike would be unsettling or even frightening inasmuch as they resemble figures from nightmares or films about the living dead. Therefore, Mori cautioned robot designers not to make the second peak their goal — that is, total human likeness — but rather the first peak of humanoid appearance to avoid the risk of their robots falling into the uncanny valley.

The uncanny valley can, however, be seen in a positive light. While many nonbiological phenomena can violate our expectations, the eerie sensation associated with the uncanny valley may be particular to the violation of (largely nonconscious) human-directed expectations. If very humanlike robots are capable of eliciting human-directed expectations, then subjects can be used to evaluate the human likeness of their behavior to an extent that would be impossible if mechanical-looking robots were used instead.

Unfortunately, there has been little direct scientific investigation of Mori's uncanny valley hypothesis in the past 35 years. Clearly, there are many qualitatively different ways of deviating from human norms of appearance and movement, some of which are more uncanny than others. In addition, the relation between appearance and behavior in creating a subjective impression of familiarity or human presence has not been well explored, nor how habituation affects that impression.

This paper attempts to explore one possible explanation of the uncanny valley — that when a humanlike robot elicits an eerie sensation it is because the robot is acting as a reminder of mortality. It attempts to test this hypothesis through the experimental methods used by terror management theory (TMT). TMT studies have correlated subliminal reminders of mortality with a wide range of attitude changes. If an android affects people's attitudes without them knowing it, this raises ethical concerns that need to be addressed. If, however, an android is a conscious reminder of death, this could impede its future adoption, although people would likely habituate to the effect to some extent. In either case, the looks or movement of the device would need to be enhanced to prevent unwanted effects.

Terror Management Theory

Like other species *Homo sapiens* are highly motivated to avoid dying. Yet unlike other species they are in the potentially terrifying position of knowing that death is inevitable. Inspired by Ernest Becker's *The Denial of Death* [Becker, 1973] and other works, for more than two decades Jeff Greenberg, Tom Pyszczynski, Sheldon Solomon, and their colleagues have been developing a theory concerning how human beings manage their fear of personal extinction [Solomon et al., 1998] [Greenberg et al., 1986]. The theory has been supported by more than 200 experiments. They posit a dual-process model. Conscious thoughts of death are either suppressed (e.g., by thinking about something else) or their immediate significance is rationalized away (e.g., "My grandmother lived to be 90.") [Pyszczynski et al., 1999]. Nonconscious thoughts of death elicit defense processes that mitigate anxiety concerning the certainty of death by supporting a person's worldview and self-esteem:

> Along with the evolutionary emergence of cognitive abilities that enabled members of our species to comprehend our own mortality, our ancestors developed a solution to the problem of death in the form of a dual-component cultural anxiety buffer consisting of (a) a cultural worldview — a humanly constructed symbolic conception of reality that imbues life with order, permanence, and stability; a set of standards through which individuals can attain a sense of personal value; and some hope of either literally or symbolically transcending death for those who live up to these standards of value; and (b) self-esteem, which is acquired by believing that one is living up to the standards of value inherent in one's cultural worldview. [Pyszczynski et al., 1999]

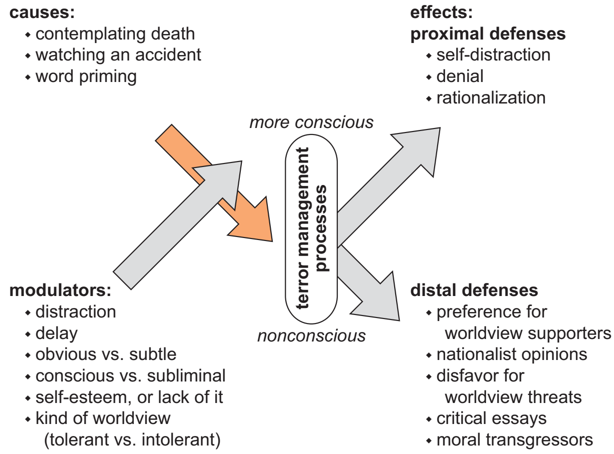


Figure 2: Terror management theory explores the relationship between reminders of death and the defense processes they elicit, including the modulating effects of intervening treatments.

Pyszczynski et al. (1999) contrast the *proximal* terror management defenses elicited by conscious thoughts of death with subliminally-elicited *distal* defenses. Distal defenses may address the threat at a level of abstraction different from that at which it is perceived and understood. Since distal defenses operate outside of consciousness (or at its fringes), they need not be rationally connected to the threat and may be best described as *experiential* in nature [Simon et al., 1997].

The *mortality salience hypothesis* predicts that, if having a worldview guards people from anxiety about the inevitability of death (e.g., by giving a literal or symbolic explanation of how death is transcended), those who have been subliminally reminded of death will react more favorably to information that supports their worldview and less favorably to information that undermines it. The hypothesis has been supported by numerous experiment, which have shown, for example, that mortality salience causes people to more strongly prefer essays that praise their country to those that criticize it [Greenberg et al., 1990] [Greenberg et al., 1994] [Greenberg et al., 2000], to prefer charismatic candidates over relationship-oriented candidates [Cohen et al., 2004], and to judge moral transgressors more harshly [Rosenblatt et al., 1989].

Such distal defenses as worldview protection are active when thoughts of death are not conscious but still accessible, such as when subjects have been distracted from death-related thoughts or after a period of delay

[Greenberg et al., 1994] [Greenberg et al., 2000]. They are, however, immediate for such subliminal priming as when the word *death* is flashed between the appearance of two other words for an interval too brief to result in one's conscious awareness of it [Arndt et al., 1997]. Although a fear of death can produce affective and physiological responses, evidence suggests that these responses do not mediate distal defenses; rather distal defences can occur in their absence [Pyszczynski et al., 1999].

Appraising human likeness by means of terror management defenses

A basic research question concerns whether very humanlike stimuli sometimes cause an eerie sensation because they remind us of death and mortality, either consciously or subliminally. For example, an android that is not animated — or not animated like a living person — may look dead. This may remind us, if only subconsciously, of the fact that we too shall die, thus setting in motion defensive mechanisms that influence our attitudes in characteristic ways. If so, we can measure these changes of attitude to explore the terrain of uncanny valley.

More specifically, the mortality salience hypothesis predicts that nonconscious but accessible thoughts of death will result in distal defenses resulting in a heightened preference for stimuli that support a person's worldview and a decreased preference for stimuli that threaten it. If the appearance or behavior of a very humanlike robot, to the extent that it is uncanny, elicits proximal or distal terror management defenses, the effects of these defenses provide a means of quantitatively appraising the human likeness of its appearance and behavior. This then places the focus on the causes of TMT defenses (see Fig. 2). So while much research on terror management explores the range of manifestations of terror management defenses (e.g., "Will people who have been reminded of their mortality be more likely to judge moral transgressors harshly?"), the current research assumes the manifestations past studies have correlated with mortality salience are valid indicators of worldview defense, and then considers the range of stimuli that elicit them.

Experiment: Does an Uncanny Appearance Elicit Distal Defenses?

The experiment was designed to test the hypothesis that an android with uncanny appearance elicits the same distal defenses that reminders of death do. The evaluation criteria are derived from known mortality salience effects in the terror management theory literature: a heightened preference for charismatic politicians relative to relationships-oriented ones [Cohen et al., 2004] and a heightened preference for foreign students who praise a participant's country relative to those who criticize it [Greenberg et al., 1990] [Greenberg et al., 1994] [Greenberg et al., 2000]. In addition, mortality salience is gauged by word completion puzzles that are expected to show a participant's preference for death-related word completions indicative of the nonconscious activation of death-related associations.

Experimental image Control image

Figure 3: The image on the left is the visual stimulus used for the experimental group. It is the head, neck, and upper torso of an android robot. The eyes are turned up and there is a gap between the eyes and eye lids because this part of the android has been powered off and disconnected from the rest of its body. The image on the right is the visual stimulus used for the control group. It depicts an Asian female in her early 20s.

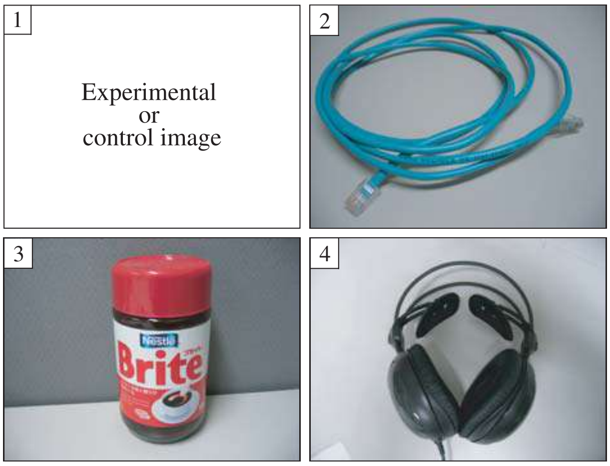


Figure 4: Participants sequentially viewed either the experimental image or the control image shown in Fig. 3 and then three other images (2–4). The images were 460 pixels in height and were captioned female figure, ethernet cable, nondairy creamer, and headphones.

Method

Participants. There were 63 participants, 25 male and 38 female, of whom 17 were 16 to 20 years old, 18 were 21 to 25, 9 were 26 to 30, 11 were 31 to 40, and 8 were over 40. All participants could communicate in English with 48 growing up in English speaking countries (5 Australia, 3 Canada, 1 Ireland, 1 New Zealand, 7 United Kingdom, and 32 United States) and 14 growing up in non-English speaking countries (1 Austria, 4 Israel, 1 Italy, 2 Korea, 1 the Netherlands, 1 Portugal, 2 Spain, 1 Turkey, and 1 Yugoslavia). Participants were recruited from Zone.com, an online gaming site. Four participants did not submit results for the word completion puzzles only. The participants were all volunteers and none received remuneration.

Procedure

Instructions. The solicitation for the experiment explained that (1) it involves filling out an online questionnaire; (2) it is for research on a cognitive mechanism that is common to all people; (3) the participant's abilities would not be evaluated; and (4) further details concerning its purpose will be revealed only after the questionnaire has been completed. Potential participants were also told (5) it takes about 10 minutes to complete the questionnaire; (6) it must be completed in order and in one uninterrupted sitting; and (7) they should relax and just give the first answer they think of. Those who agreed to participate were given a link to the questionnaire website.

The website reiterated points 5 to 7 above and summarized the contents of the questionnaire: "You'll be shown some pictures, and you'll be asked some questions to see what you remember about them. Then you will be asked about a couple of excepts from political speeches and comments made by foreign students. Then you will solve some word puzzles." The wording of the questionnaire was intentionally informal because past studies have found that an informal experimental setting is more conducive to mortality salience effects [Simon et al., 1997], perhaps because participants tend to follow base their judgments on gut feelings rather than rational arguments. The computer, session, and starting time were uniquely identified to ensure that the same individual had not filled out the questionnaire more than once, and indeed nobody had.

Participants were asked their gender, approximate age, the country where they grew up, and then had to consent to the experiment: "This questionnaire is voluntary, so you may quit at any time. Data collected may be used in future studies, but it will be stripped of personal information. Clicking *I consent* means you want to go ahead."

Group assignment and stimuli. Participants were then randomly assigned with equal probability to either an experimental group or a control group. There were 31 participants in the experimental group and 32 in the control group. Those in the experimental group viewed the uncanny image of an android, while those in the control group viewed the image of a young Asian female (see Fig. 3). (Perhaps the main reason the android looked uncanny was because the eyes were looking up and sunken because the android was powered off and separated from its base.) In all other respects, the questionnaire was identical for both the experimental and control group. The subjects then viewed in sequence three "neutral" images. For each of the four images, they were instructed to view the image for a couple of seconds and then press a button labeled *view next image.*

Delay. The participants were then asked eight questions about the images. The answers to these questions were not kept. The questions served two functions: they distracted participants about the true purpose of the questionnaire since they concerned recall, and they added a delay before the questions relevant to terror management theory. Past TMT research has found that mortality salience effects appear immediately after subliminal priming on death but only after a delay when death is perceived consciously. Without knowing in advance whether the android would serve as a reminder of death and, if so, whether participants would be conscious of it as such, it was though prudent to insert a delay.

Worldview-related questions. Participants were next asked to read campaign speeches from two political candidates and to rate on a nine-point scale how well they liked each candidate and how insightful they thought each candidate was. They were then asked which candidate they would vote for. The first speech was *charismatic* and the second was *relationship-oriented* (see Appendix A). The speeches were loosely paraphrased from a previous study that indicated subjects in whom a subconscious fear of death has been elicited are more likely to prefer charismatic leaders [Cohen et al., 2004].

The same five questions were repeated for two foreign students who commented on their experience living in the participant's home country: Participants had to rate on a nine-point scale how well they liked each student, how insightful each student was, and which student they would support if both were running for president of the student government. The first student praised the participants' country, while the second student criticized the participants' country. These questions were motivated by a previous study that indicated subjects in whom a subconscious fear of death has been elicited are more likely to prefer people who support their worldview [Greenberg et al., 1990] [Greenberg et al., 1994] [Greenberg et al., 2000].

Word completion puzzles. Participants were next given 35 word completion puzzles of the following form:

`RELA_ _G` (with `?` under the first blank)

A button under each puzzle reads, "Give yourself three seconds to think of the missing letter with a ? under it, and then click here." After clicking the button, the puzzle vanishes and several choices would appear among which the participant may select only one. In the above puzzle, for example, a participant might select X to signify *relaxing*:

◯ T ⦿ X ◯ Y ◯ other / don't know

The participants would then be taken to the next question.

Following the TMT literature, dispersed among this set of 35 puzzles are 7 that allow participants to choose among word completions, one of which is related to death. These puzzles are intended to detect a subconscious activation of death-related concepts. The puzzles in the questionnaire are listed below with italics denoting the death-related option: COFF– –: *coffin,* coffee; SK– –L: skill, *skull*; MUR– –R: murmur, *murder*; GRA– –: grace, grade, grate, *grave*, graze; BUR–E–: burden, burger, *buried*, burned/burner, burped, burred; –EAD: bead, *dead*, head, lead, mead, read; STI– –: stick, *stiff*, still/stile/stilt, stink/stint/sting.

A further 7 questions are intended to detect a subconscious activation of concepts that are roughly synonymous with the uncanny. The puzzles are listed below with with italics denoting the option related to the uncanny: WEI– –: weigh, *weird*; – –EEPY: *creepy,* sleepy; –DD: add, *odd*; UN–A– –Y: *uncanny*, unhappy, unmanly; – – – –LIAR: familiar, *peculiar*; ST– –GE: storage, *strange*; OM– –US: *ominous*, omnibus.

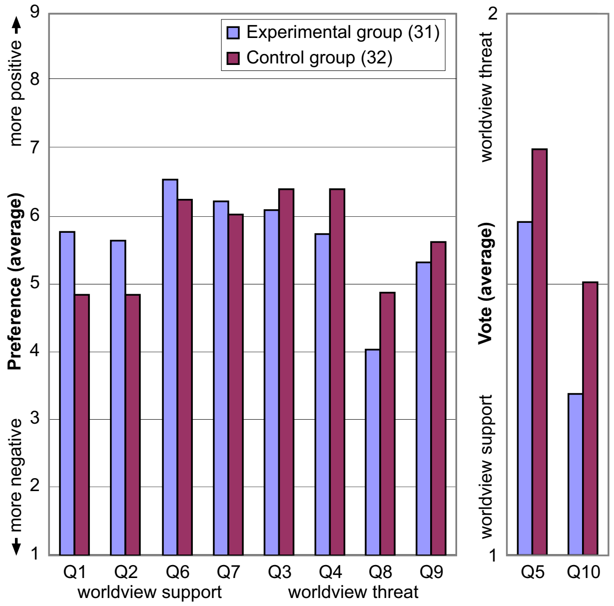


Figure 5: The experimental group shows more of a preference for the charismatic candidate (questions 1 and 2) and less of a preference for the relationship-oriented candidate (question 3 and 4) relative to the control group. They were also less likely to vote for the relationship-oriented candidate (question 5). This reproduces part of the results of [Cohen et al., 2004] but using an android as the experimental stimulus. The experimental group likewise shows more of a preference for the foreign student who praises their home country (questions 6 and 7) than the one who criticizes it (questions 8 and 9) relative to the control group, and they would vote for the praising student by a wide margin (question 10). This reproduces part of the results of [Greenberg et al., 1990] [Greenberg et al., 1994] [Greenberg et al., 2000] but using an android as the experimental stimulus.

The remaining 23 puzzles are unrelated to death and the uncanny. They are intended to disguise the true purpose of the experiment. Among 63 participants, 59 submitted results for the word completion section.

Suspicion and qualitative remarks. Finally, the participants were asked whether they had any difficulty completing the questionnaire; whether they had any suspicion concerning what the questionnaire was about; and what their impression was of the four images shown at the beginning. Some participants in the experimental group were selected for further questions concerning their impression of the uncanny image of the android. The participants were finally debriefed concerning the purpose of the experiment.

Results

Worldview-related questions. The results show on average a consistent preference for worldview supporters and against worldview threats in the experimental group (see Table 1 and Fig. 5). The experimental group rated the charismatic political candidate nearly a point higher for likeability (+0.93) and insight (+0.80) and rated the relationships-oriented candidate lower on likeability

Table 1: Worldview: Average (median) values

Question	Experimental	Control
1	5.77 (5)	4.84 (5)
2	5.65 (6)	4.84 (5)
3	6.10 (6)	6.41 (7)
4	5.74 (6)	6.41 (7)
5	1.61 (2)	1.75 (2)
6	6.55 (7)	6.25 (6)
7	6.23 (6)	6.03 (6)
8	4.03 (4)	4.88 (5)
9	5.32 (5)	5.63 (6)
10	1.29 (1)	1.5 (1.5)

(−0.31) and insight (−0.66). The experimental group rated the foreign student who praised the partipants' country higher for likeability (+0.30) and insight (+0.19) and rated the one who criticized it lower on likeability (−0.84) and insight (−0.30). (For preference questions, 1 = strongly negative, 5 = neutral, and 9 = strongly positive.)

The charismatic candidate lost by 7 votes in the experimental group (12 to 19) but by 16 votes in the control group (8 to 24), more than double the margin. The praising and critical foreign students tied in the control group (16 to 16), while the praising foreign student won by a 13 vote margin in the experimental group (22 to 9). (For voting questions, 1 = worldview-supportive candidate and 2 = worldview-critical candidate.)

It is difficult to show strong statistical significance for any isolated question in the worldview section owing to a high variance in the data. The average standard deviation for questions 1 to 4 and 6 to 9 was 2.00 and for questions 5 and 10 it was 0.48. However, when the values were summed together with sign changes on questions that the mortality salience hypothesis predicts will be disfavored by the experimental group, Student's t-test (two tails, heteroscedastic) showed statistical significance among the candidate-related questions ($t = 0.0307$) and overall ($t = 0.0348$).

Word completion puzzles. Among the 28 participants in the experimental group, there were 49 death-related word completions as compared to 38 among the 31 participants in the control group, 85 uncanny-related word completions as opposed to 66 in the control group, and 134 combined death and uncanny-related word completions as opposed to 104 in the control group (see Table 2).

On average the experimental group had 0.524 more death-related word completions, 0.907 more uncanny-related word completions, and 1.430 more death and uncanny-related word completions than the control group (see Table 3).

Student's t-test (two tails, heteroscedastic) showed strong statistical significance uncanny-related questions ($t = 0.0132$), but not for death ($t = 0.0963$) related questions. For combined death and uncanny-related questions, the statistical significance was the highest

($t = 0.00542$).

The median time spent viewing the first image was 12 seconds, and the average time was 21.9 seconds. (The first image is the android head in the experimental group and the Asian woman in the control group.) The minimum time was 2 seconds, and the three longest times were 262, 127, and 67 seconds.

The median time spent in the worldview section of the questionnaire by the 63 participants was 336 seconds, and the average time was 350.9 seconds. The minimum time was 162 seconds, and the three longest times were 735, 604, and 593 seconds. No figures were recorded for the word completion section.

When asked at the end of the questionnaire, "What was your feeling about or impression of the four images shown at the beginning?" 16 participants did not comment on the android or other figures or said that they had no particular feeling. The remaining 12 appeared to be commenting on the android. An exhaustive list of comments about the android head are as follows:

1. Scary female image.
2. I thought the first wax head was really interesting.
3. The women figure kind of freaked me out a bit.
4. Strange.
5. The first two images were disturbing.
6. I thought it was kind of bizarre. The Japanese girl's eyes were messed-up looking.
7. The first image seemed to come across as very sickly. It made me feel sick just looking at it.
8. Weird lady.
9. The woman seemed to be dead. It was a shocking image and given the chance I would have preferred not to see it.
10. The woman was frightening.
11. Weird.
12. I thought it humorous that two of them were related to Asian things.

No one made explicit reference to the content of an image other than the image of the android. The participants were not shown the images again at the end of the questionnaire, so all these comments were based on memory.

Table 2: Word completion: Totals

Type	Experimental	Control
Subjects	28	31
Death	49	38
Uncanny	85	66
Combined	134	104

Table 3: Word completion: Average (median) values

Type	Experimental	Control
Death	1.75 (2)	1.23 (1)
Uncanny	3.04 (3)	2.13 (2)
Combined	4.79 (5)	3.35 (3)

Interviews

Several participants were interviewed through instant messaging. They were all asked, **"What do you see when you look at this image?"** Margot[1] and her children thought it looked dead. Craig and Judith agreed after being prompted. Judith at first thought it was a real person who had fainted. Penelope thought it could not look dead because it had never been alive. Derek and Loretta both thought it looked human at first; however, Derek did hot have an emotional response, whereas Loretta had a strong response. Clearly, there is considerable diversity of results in this small sample.

Penelope: A scary robot.

Do you see something death related?

No, I see something not natural, but nothing to do with death. It is something that never has been alive, so it cannot be dead. The skin, the opened mouth, the right eye make it seems like it had an overload of electricity or a short circuit.

But in the experiment it is registering death saliency.

Well no, the thing that makes it scary is that it is a copy of a person. That's the truth. But not a good copy. You know, it is like the old dummies in empty shops.

Is it still as scary as it was before?

Yes.

Judith: Oh, my God, it's scary. She look like she is fainting.

You said "she." What do you mean? What is "it"? What makes it scary?

It looks like a female. The look on the face looks like she is fainting.

But it isn't a real female, right?

It is a real female, isn't it?!

No. It is a robot.

What?

That is the head of a robot.

So that's why the eyes look like that. The eyes make it look like its fainting, Yes, it looks dead.

So looking dead makes it scary?

A bit, yes, for a first impression, of course. It makes a big difference. It look so real though.

Craig: It looks like a Korean or Chinese fake woman. It gives an ugly impression.

Does it seem strange, scary, or dead?

I dislike it. Scary and dead, yes.

Derek: Being an android myself, I can't say that I really have any feelings about it. But seriously, I don't know.

Does it seem uncanny or eerie? Alive or dead? In pain? Do you have any strange feeling?

I didn't think of it in those terms.

In what terms did you think of it?

[1]The names have been changed. Grammatical errors have been fixed. Participants had to message the following consent in addition to the consent of the experiment: "I consent that my answers may be used in a publication or presentation once stripped of identifying information."

I noticed features, but didn't form an impression of the significance of the features.

Do you mean you looked at it more scientifically or analytically? Not emotionally?

E.g., the eyebrows appear to be shaved in a "non-standard" way. The eyes appear to be "askew." The skin tone is slightly mottled or uneven.

So you are not grossed out? You don't feel any emotional response?

No, of course not. It wasn't really an emotional reaction, or even a judgment of artificiality.

Does it look alive or dead?

I can't really tell.

Well, if it looks like an android, then I suppose neither?

It could go either way. I only said android because that's the name in the URL.

Oh. Well, what if that were not the name. Then what would you say?

I guess the default response would be to anthropomorphize it (or her, rather).

Human?

Yeah, probably. The default would probably start with human, pending further data. The teeth are imperfect, which might argue for human.

Loretta: I see a woman that does not look real, but there is pain in the face, I also see torture. It could just be inside though.

What is your impression of it?

I feel this is some sort of art work. I don't like the eyes, but the lips are beautiful.

How do you feel about it?

It makes me feel sorrow, sadness, and abuse. Around the neck and face area that can be seen. What are my real thoughts? This is not a real human!

What you say sound contradictory. Does it look human or mechanical?

Well, when I first looked at it, I saw torture and pain. That was my first impression. Then the more I looked, I could see it was not real and as you said now a machine. But all the hurt was there.

And does it seem alive, dead, or what?

Dead.

But at first did it look alive or dead?

At first I saw torture and then it looked dead.... I was wondering if I saw all that torture and pain because I'd been through it? What you think?

Margot: It reminds me of death. That's the first thing I thought of when I saw it, the first time I took the survey.

Oh, I see. And how do you feel about it? What do you think when you look at it? What emotions do you feel?

It's unpleasant to look at. Hmmm. Nervous, a bit fearful, uncomfortable.

I see. So for you it is definitely a reminder of death.

Yeah, it is. I asked my daughter what it reminded her of, and she said death too with no prompting. And she says it reminded her of *The Grudge*, a Japanese horror film.

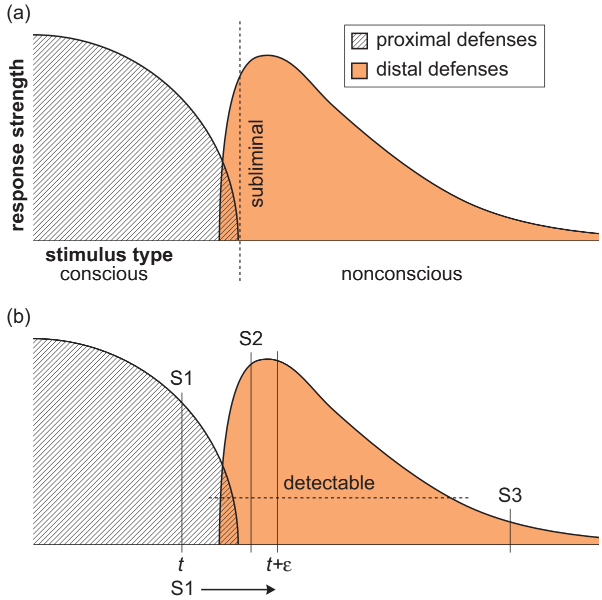


Figure 6: (a) Stimuli may be perceived in focal or fringe consciousness or subliminally. This can produce proximal or distal defenses with varying response strengths. (The curves in the figure are given only for the sake of example; their actual shapes are unknown.) (b) Different stimuli may elicit different kinds of defenses (S1 and S2) as may the same stimulus (S1) at different times $(t, t+\epsilon)$ owing to the effects of delay. In addition, for a give sample size, the effects of some stimuli may be impossible to measure owing to variance in the data (S3.)

Oh, I saw it! Wow, you let her see that?!

No! Well, some parts like two minutes at the start. Then I sent her away. She didn't want to see anymore anyway. I'll ask my other daughter. One sec. I'll call her.

Great.

Yep, death also.

No prompting?

Nope, and this is also without hearing my other daughter's answer because she was on the phone.

Discussion

Most people had no difficulties with the questionnaire. A few people had to do extra scrolling in the word completion questions because they had missed a question. A few complained that the word completion puzzles were difficult, and a couple non-native speakers said that was owing to the language barrier. Only a few people had suspicions about the questionnaire, but their suspicions had nothing to do with what the questionnaire was actually about.

There was a minor flaw in the experimental procedure. It would have been useful to record the precise amount of time a participant spent on each part of the experiment so that the influence of delay could be considered in more detail.

The effects of distal terror management defenses may not be such a reliable indicator of the degree of mortality salience of a given stimulus. The same stimulus will affect the attitudes of individuals differently. (For example, in one study "low authoritarian individuals did not derogate attitudinally dissimilar others when mortality was made more salient," while high authoritarian individuals did ([Solomon et al., 1998], p. 40, citing an experiment in [Greenberg et al., 1992]). In addition, proximal and distal defenses have varying response strengths depending on whether the stimulus is perceived in focal or fringe consciousness or subliminally (Fig. 6(a)). The same stimulus can produce varying effects owing to delay, some of which will be too weak to detect owing to the high degree of variance in the data (Fig. 6(b)).

A more fundamental concern relates to affect. The eerie sensation identified with the uncanny valley may be characterized as affective, although it seems difficult to identify it with one or more primary emotions like disgust. Terror management studies have indicated that affect does not mediate distal defenses [Pyszczynski et al., 1999]. And it seems that subliminal priming of death-related words does not create an eerie sensation. That leaves the question of what nonconscious processes do underpin the eerie sensation. The uncanny android still seems to be a reminder of death, but perhaps a conscious reminder, in which case the distal (i.e., nonconscious) defenses that showed up in the experimental results may have occurred owing to delay.

Probably at the end of the study all participants should have been showed the uncanny android and asked to rate how eerie they felt it was and given a questionnaire to evaluate its emotional impact (e.g., PANAS), but already some were complaining about the questionnaire's length. In addition, another experiment could have measured the experimental stimulus's impact on such physiological responses as heart rate, respiration, and galvanic skin response.

The fact that androids are often not capable of satisfying the human-directed expectations they elicit may be one reason why we may perceive them to not be fully alive. However, it also suggests alternative explanations of the uncanny. If one person elicits expectations in another, that person elicits contextually-appropriate behavior (i.e., behavior that can be described in terms of norms). Given further expectations for real-time responding (regarding the responsive behavior that is appropriate in this context) androids tend to violate human expectations for how to go on. This suggests that some of the peculiarities of interacting with androids may be owing to failures to model the microbehavior central to the expectational cycle.

This opens the door to hypotheses about the uncanny valley that are unrelated to reminders of mortality. So an important question is whether there is something distinctive about expectations elicited by the human form that causes their violation to result in sensations that are qualitatively different from those elicited by other forms. Anecdotal evidence suggests that there is.

Conclusion

Many kinds of media (e.g., computers, robots, films) are capable of eliciting, to varying degrees, different kinds of social responses (e.g., verbal [Reeves and Nass, 1996], gestural, gaze [MacDorman et al., 2005]). Nevertheless, qualitative (and quantitative) differences emerge depending on the type of media and how it acts. The eerie feeling elicited by human-looking but not mechanical-looking robots is one such qualitative difference, and its significance is worth exploring.

This study has hypothesized that an uncanny-looking android may be uncanny because it elicits a fear of death, and it has attempted to verify this through questions designed to measure such distal terror management defenses as worldview protection. The results are favorable. On average the group exposed to an image of an uncanny robot consistently preferred information sources that supported their worldview relative to the control group.

The results, however, only apply to one particular stimulus, so it is important to ascertain whether they generalize across uncanny stimuli and, in particular, to uncanny movement in a robot that otherwise looks human and natural.

Acknowledgments

Much appreciation goes to Sara Kiesler for her suggestion that conflicting evidence for human likeness from a humanlike entity may elicit the same terror management defenses that reminders of death do. The android was developed in a collaboration between Kokoro, Co., Ltd. and Osaka University's Intelligent Robotics Laboratory, directed by Hiroshi Ishiguro.

Appendix A: Worldview-related Statements

Politicians. Two candidates are running for our nation's highest office. An excerpt from each candidate's campaign speech is given below:

"I will be the best leader of this great nation because I am committed to a brighter vision of our future. I set high standards for my cabinet and myself and expect them to work as hard as I do to achieve these standards. I want everyone, both public employees and private citizens, to do their best for our great nation, so that we can all achieve our full potential. My goal is to do things differently from my predecessors, and I am willing to take risks to show voters how things can be improved. You are not just ordinary citizens. You are part of a great nation, and by working together we can have a big impact." – J.N.

"I will be the best leader of this nation because I am concerned about people's welfare. I treat everyone with consideration and respect, no matter how high the political tensions may rise. I stress communication among my staff and the general public. I keep everyone informed about proposed legislation, and I am open to suggestions. I frequently have meetings with constituents to discuss policies. I encourage all citizens to take a role in improving things because I know that each individual can make a difference. Everyone's contributions are recognized and appreciated." – B.E.

Foreign Students. Two foreign students comment on their experiences living in your country:

"I am a foreign student from Phnom Penh, which is the capital of Cambodia. I have been studying civil engineering at university in your country for three years. I love your country because there are many opportunities for me here that do not exist in my home country. Most people I have met have been kind to me, although they know I am a foreigner. I already had a positive impression of your country before coming here because your government has contributed much to the United Nations to support the clearance of landmines in Cambodia, and there are also many private citizens who either volunteer to work in Cambodia or make donations to support aid. After graduation, I would like to help develop my country too." – S.L.

"I am a foreign student from Bandung, which is the largest city in West Java, Indonesia. Last year, I started a Master's degree here. Although my course is challenging, I am surprised how ignorant other students are about other countries. That may be one reason the people have been mislead by your government, which follows short-sighted policies, which appear to be aligned with the interests of the United States, but which actually won't help anyone in the long run. If your country really wants to bring an end to terrorism, they should stop provoking Moslems with bellicose policies or propping up autocratic sheiks." – H.M.

Appendix B: *The Uncanny Valley*

Mori, M. (1970). *Bukimi no tani* [The uncanny valley]. *Energy*, 7(4), pp. 33-35. Translated by Karl F. MacDorman and Takashi Minato.

Valley of familiarity

There are mathematical functions of the form $y = f(x)$ for which the value of y increases (or decreases) continuously with the value of x. For example, as the effort x increases, income y increases, or as a car's accelerator is pressed, the car moves faster. This kind of relationship is ubiquitous and easily understood. In fact, it covers most phenomena, so we might think that this function can represent all relations. That is why people are usually puzzled when faced with some phenomenon it cannot represent.

Climbing a mountain is an example of a function that does not increase continuously: a person's altitude y does not always increase as the distance from the summit decreases owning to the intervening hills and valleys. I have noticed that, as robots appear more humanlike, our sense of their familiarity increases until we come to a valley. I call this relation the *uncanny valley.*

Recently there are many industrial robots, and as we know the robots do not have a face or legs, and just rotate or extend or contract their arms, and they bear no resemblance to human beings. Certainly the policy for designing these kinds of robots is based on functionality. From this standpoint, the robots must perform functions similar to those of human factory workers, but their appearance is not evaluated. If we plot these industrial robots on a graph of familiarity versus appearance, they lie near the origin (see Fig. 1). So they bear little resemblance to a human being, and in general people do not find them to be familiar. But if the designer of a toy robot puts importance on a robot's appearance rather than its function, the robot will have a somewhat humanlike appearance with a face, two arms, two legs, and a torso. This design lets children enjoy a sense of familiarity with the humanoid toy. So the toy robot is approaching the top of the first peak.

Of course, human beings themselves lie at the final goal of robotics, which is why we make an effort to build humanlike robots. For example, a robot's arms may be composed of a metal cylinder with many bolts, but to achieve a more humanlike appearance, we paint over the metal in skin tones. These cosmetic efforts cause a resultant increase in our sense of the robot's familiarity. Some readers may have felt sympathy for handicapped people they have seen who attach a prosthetic arm or leg to replace a missing limb. But recently prosthetic hands have improved greatly, and we cannot distinguish them from real hands at a glance. Some prosthetic hands attempt to simulate veins, muscles, tendons, finger nails, and finger prints, and their color resembles human pigmentation. So maybe the prosthetic arm has achieved a degree of human verisimilitude on par with false teeth. But this kind of prosthetic hand is too real and when we notice it is prosthetic, we have a sense of strangeness. So if we shake the hand, we are surprised by the lack of soft tissue and cold temperature. In this case, there is no longer a sense of familiarity. It is uncanny. In mathematical terms, strangeness can be represented by negative familiarity, so the prosthetic hand is at the bottom of the valley. So in this case, the appearance is quite human like, but the familiarity is negative. This is the uncanny valley.

I don't think a *bunraku* puppet is similar to human beings on close observation. Its realism in terms of size, skin, and so on, does not approach that of a prosthetic hand. But when we enjoy a puppet show in the theater, we are seated far from the puppets. Their absolute size is ignored, and their total appearance including eye and hand movements is close to that of human beings. So although the puppets' bodies are not humanlike, we can feel that they are humanlike because their bodies and movements when taken together are humanlike. And from this evidence I think their familiarity is very high.

From the above maybe readers can understand the concept of the uncanny valley. So in the following I consider the relationship between movement and the uncanny valley.

The effects of movement

For creatures, including robots, movement is generally a sign of life. Adding movement changes the shape of the uncanny valley by exaggerating the peaks and valley (see Fig. 1). For the industrial robot, the impact of movement is relatively slight because we see it as just a machine. If it stops moving, it becomes a mere oily machine. But if programmed properly to generate humanlike movements, we can enjoy some sense of familiarity. Humanlike movement requires similarity of velocity and acceleration. Conversely, if we add movement to a prosthetic hand, which is at the bottom of the uncanny valley, our sensation of strangeness grows quite large. Some readers may know that recent technology has enabled prosthetic fingers to move automatically. A commercially available prosthetic hand made with the highest technique was developed in Vienna. To explain how it works, the intention to move the forearm, even if missing, produces current in the arm muscles that can be detected by an electromyogram. So the prosthetic hand detects the current by means of electrodes and amplifies the signal to activates a small motor in the prosthetic arm to move the fingers. This hand can move in a way that causes some ordinary people to feel uneasy. If you shook a woman's hand with this hand in a dark place, the woman must be shocked!

Since these effects are apparent for just a prosthetic arm, the strangeness will be magnified if we build an entire robot. You can imagine going to a work place where there are many mannequins: if a mannequin started to move, you might be shocked. This is a kind of horror.

In the World Expo held in Osaka this year, the robots displayed a more elaborate design. For example, one robot has 29 artificial muscles in the face to make humanlike facial expressions. According to the designer, laughing is a kind of sequence of face distortions, and the distortion speed is an important factor. If we cut the speed in half, laughing looks unnatural. This illustrates how slight variations in movement can cause a robot, puppet, or prosthetic hand to tumble down into the uncanny valley.

Escape by design

We hope to design robots or prosthetic hands that will not fall into the uncanny valley. So I recommend designers take the first peak as the goal in building robots rather than the second. Although the second peak is higher, there is a far greater risk of falling into the uncanny valley. We predict that it is possible to produce a safe familiarity by a nonhumanlike design. So designers please consider this point. A good example is glasses. Glasses do not resemble the real eyeball, but this design is adequate and can make the eyes more charming. So we should follow this principle when we design prosthetic eyes. We can create an elegant prosthetic hand - one that must be fashionable. Artist who makes statues of Buddhas created a model of a human hand that is made from wood. The fingers bend at their joints. The hand has no finger print, and it assumes the natural color of wood. But we feel it is beautiful and there is no sense of the uncanny. Maybe wooden hand can serve as a reference for future design.

The significance of the uncanny

In Fig. 1, a healthy person is at the top of the second peak. And when we die, we fall into the trough of the uncanny valley. Our body becomes cold, our color changes, and movement ceases. Therefore, our impression of death can be explained by the movement from the second peak to the uncanny valley as shown by the dashed line in the figure. We might be happy this line is into the still valley of a corpse and that of not the living dead! I think this explains the mystery of the uncanny valley: Why do we humans have such a feeling of strangeness? Is this necessary? I have not yet considered it deeply, but it may be important to our self-preservation.

We must complete the map of the uncanny valley to know what is human or to establish the design methodology for creating familiar devices through robotics research.

References

[Arndt et al., 1997] Arndt, J., Greenberg, J., Pyszczynski, T., and Solomon, S. (1997). Subliminal presentation of death reminders leads to increased defense of the cultural worldview. *Psychological Science*, 8:379–385.

[Becker, 1973] Becker, E. (1973). *The denial of death*. Free Press, New York.

[Cohen et al., 2004] Cohen, F., Solomon, S., Maxfield, M., Pyszczynski, T., and Greenberg, J. (2004). Fatal attraction: The effects of mortality salience on evaluations of charismatic, task-oriented, and relationship-oriented leaders. *Psychological Science*, 15:846–851.

[Greenberg et al., 1986] Greenberg, J., Pyszczynski, T., and Solomon, S. (1986). The causes and consequences of the need for self-esteem: A terror management theory. In Baumeister, R. F., editor, *Public self and private self*, pages 189–212, New York. Springer-Verlag.

[Greenberg et al., 1990] Greenberg, J., Pyszczynski, T., Solomon, S., Rosenblatt, A., Veeder, M., Kirkland, S., and Lyon, D. (1990). Evidence for terror management theory ii: The effects of mortality salience reactions to those who threaten or bolster the cultural worldview. *Journal of Personality and Social Psychology*, 58:308–318.

[Greenberg et al., 1994] Greenberg, J., Pyszczynski, T., Solomon, S., Simon, L., and Breus, M. (1994). Role of consciousness and accessibility of death-related thoughts in mortality salience effects. *Journal of Personality and Social Psychology*, 67:627–637.

[Greenberg et al., 2000] Greenberg, J., Simon, L., Arndt, J., Pyszczynski, T., and Solomon, S. (2000). Proximal and distal defenses in response to reminders of ones mortality: Evidence of a temporal sequence. *Personality and Social Psychology Bulletin*, 26:91–99.

[Greenberg et al., 1992] Greenberg, J., Simon, L., Pyszczynski, T., Solomon, S., and Chatel, D. (1992). Terror management and toleraance: Does mortality salience always intensify negative reactions to others who threaten one's worldview? *Journal of Personality and Social Psychology*, 62:212–220.

[MacDorman et al., 2005] MacDorman, K., Minato, T., Shimada, M., Itakura, S., Cowley, S., and Ishiguro, H. (2005). Assessing human likeness by eye contact in an android testbed. In *Proceedings of the XXVII Annual Meeting of the Cognitive Science Society*, Stresa, Italy.

[Matsui et al., 2005] Matsui, D., Minato, T., MacDorman, K., and Ishiguro, H. (2005). Generating natural motion in an android by mapping human motion. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Canada.

[Minato et al., 2004] Minato, T., Shimada, M., Ishiguro, H., and Itakura, S. (2004). Development of an android robot for studying human-robot interaction. In Orchard, R., Yang, C., and Ali, M., editors, *Innovations in Applied Artificial Intelligence*, pages 424–434, Berlin.

[Mori, 1970] Mori, M. (1970). Bukimi no tani [the uncanny valley]. *Energy*, 7:33–35.

[Pyszczynski et al., 1999] Pyszczynski, T., Greenberg, J., and Solomon, S. (1999). A dual-process model of defense against conscious and unconscious death-related thoughts: An extension of terror management theory. *Psychological Review*, 106(4):835–845.

[Reeves and Nass, 1996] Reeves, B. and Nass, C. (1996). *The media equation: How people treat computers, television, and new media like real people and places.* Cambridge University Press (CSLI), New York.

[Rosenblatt et al., 1989] Rosenblatt, A., Greenberg, J., Solomon, S., Pyszczynski, T., and Lyon, D. (1989). Evidence for terror management theory: I. the effects of mortality salience on reactions to those who violate or uphold cultural values. *Journal of Personality and Social Psychology*, 57:681–690.

[Simon et al., 1997] Simon, L., Greenberg, J., Harmon-Jones, E., Solomon, S., Pyszczynski, T., Arndt, J., and Abend, T. (1997). Terror management and cognitive-experiential self-theory: Evidence that terror management occurs in the experiential system. *Journal of Personality and Social Psychology*, 72(5):1132–1146.

[Solomon et al., 1998] Solomon, S., Greenberg, J., and Pyszczynski, T. (1998). Tales from the crypt: The role of death in life. *Zygon: Journal of Religion and Science*, 33:9–43.

Virtual Reality & Intelligent Hardware

• Article •

October 2022 Vol. 4 No. 5: 393–405
10.1016/j.vrih.2022.08.003

Uncanny valley for interactive social agents: an experimental study

Nidhi MISHRA[1], Manoj RAMANATHAN[2*], Gauri TULSULKAR[1],
Nadia Magneat THALMANN[3,4]

1. *Institute for Media Innovation, Nanyang Technological University,* 637553, *Singapore;*
2. *Rehabilitation Research Institute of Singapore, Nanyang Technological University,* 308232, *Singapore;*
3. *Institute for Media Innovation, Nanyang Technological University,* 637553, *Singapore;*
4. *MIRALab, Battelle, Building A 7, Route de Drize, CH-*1227 *Carouge, Geneva, Switzerland*

Received 10 June 2022; **Revised** 22 July 2022; **Accepted** 18 August 2022

Abstract: Background The uncanny valley hypothesis states that users may experience discomfort when interacting with almost human-like artificial characters. Advancements in artificial intelligence, robotics, and computer graphics have led to the development of life-like virtual humans and humanoid robots. Revisiting this hypothesis is necessary to check whether they positively or negatively affect the current population, who are highly accustomed to the latest technologies. **Methods** In this study, we present a unique evaluation of the uncanny valley hypothesis by allowing participants to interact live with four humanoid robots that have varying levels of human-likeness. Each participant completed a survey questionnaire to evaluate the affinity of each robot. Additionally, we used deep learning methods to quantify the participants' emotional states using multimodal cues, including visual, audio, and text cues, by recording the participant–robot interactions. **Results** Multi-modal analysis and surveys provided interesting results and insights into the uncanny valley hypothesis.

Keywords: Uncanny valley hypothesis; Human robot interaction; Interactive robots; Humanoid robots; Virtual humans

Supported by the National Research Foundation, Singapore under its International Research Centers in Singapore Funding Initiative; and Institute for Media Innovation, Nanyang Technological University (IMI-NTU).

Citation: Nidhi MISHRA, Manoj RAMANATHAN, Gauri TULSULKAR, Nadia Magneat THALMANN. Uncanny valley for interactive social agents: an experimental study. Virtual Reality & Intelligent Hardware, 2022, 4(5): 393–405

1 Introduction

In the last decade, significant progress has been made in the science of robotics and artificial intelligence (AI). This has led to the development of humanoid robots or virtual humans with a human-like appearance, intelligence, and behavior (verbal and non-verbal), such as Nadine[1], Erica[2], and Sophia. Despite being validated in real-life applications such as banking[3], newscasting[2], therapy, and other roles[4–7], a main concern for humanoid research and study is the uncanny valley hypothesis[8]. A previous study[8] hypothesized that a

*Corresponding author, mramanathan@ntu.edu.sg

person′s emotional or affective response varies depending on the appearance of the robot with which they interact. A person may feel uneasy and unnerved with a more human-like robot. This hypothesis is considered primarily when designing human-like robots, although other studies[9,10] have found inconsistent empirical evidence supporting it. The original hypothesis generalizes the definition of a humanoid robot into a single data point, which may be inaccurate. For example, Eaton provided a comprehensive taxonomy of different types of possible humanoids[11]. With the development of human-sized robots, such as Nadine[1], Erica[2], and Ishiguro[12,13] that can be placed in social scenarios to interact directly with users, it is crucial to reassess the uncanny valley hypothesis for its relevance.

A more detailed examination of this hypothesis for interactive, social humanoid robots is required to evaluate present-day opinions of such agents. No specific study has examined interactive robots for the uncanny valley problem in action. Most studies involved no interactions[14–16] in their assessments. They instead showed videos of the robots or virtual avatars to participants and then asked the participants questions to gauge the impact of the robots' appearances. In this study, we evaluate the uncanny valley theory through live human interactions with four human-like entities:

1. Maya: simple voice assistant (only a human voice).
2. Nao[17]: child-sized programmable humanoid robot with articulated limbs but without skin or hair.
3. Nicole: virtual human with a complete virtual human-like embodiment.
4. Nadine[1]: complete life-sized humanoid robot with skin, articulated hands, and other human-like features.

Our study intends to answer the following research questions.

・ **Uncanny valley for Interactive Humanoid Robots**: Exploring this theory to provide an in-depth look into how people's emotions and perceptions vary for different types of human-like interactive robots.

Examining how the Uncanny valley affects the current human generation, which is more accustomed to advanced technologies and may be more open to human-like entities.

・**Using AI for Uncanny valley quantification**: Quantifying the emotional responses of participants using surveys and multimodal emotion and sentiment analyses (using visual, audio, and text).

This remainder of this paper is organized as follows:

・ Section 2 reviews previous research in the field of the uncanny valley hypothesis.

・ Section 3 addresses our proposed experimental setup and the details of each humanoid robot used in the current research.

・ Section 4 provides details of the data collection procedures and emotion analysis methods employed in this study.

・ Section 5 details the results obtained and provides insights from visual, audio, text, and survey data analyses.

・ Finally, Section 6 provides the conclusions and discussions.

2 Related work

According to the uncanny valley hypothesis[8], users may experience negative affective emotions or a state of eeriness when interacting with near-human entities or agents. Owing to this, increasing the agent's anthropomorphic realism would have a counterproductive effect on the users′ subjective experiences[10]. Thus, the hypothesis remains a guiding principle in robot design and cross-modal technologies, such as virtual character design[18], video games, and animations[19].

Since the proposal of the hypothesis, several studies have recreated or visualized the effect[20], tested its validity[9,10] and used perceptual analyses, cognitive analyses[21–23], and other procedures to investigate this effect. A previous study[10] noted that the original hypothesis was not validated by any empirical tests. Studies examining the validity of the hypothesis[9] have uncovered no empirical evidence to support it and have reported inconsistent findings with different conceptualizations. A few other studies[9,10] primarily perceived uncanny valleys owing to the perceptual mismatch, categorical ambiguities, and other factors. For example, unusual physical attributes or inconsistent human-like realism can lead to negative emotions.

People also dislike human-like robots making moral decisions compared to the same choices made by humans or non-human robots[24]. These studies have also noted that an uncanny effect is not generalizable across different individuals, stimuli, situations, tasks, and time. A nuanced understanding is required to precisely know when and under what conditions the negative emotions are observed.

A major drawback of the original hypothesis is it does not provide the exact definition or standards of human-likeness, affinity, eeriness, and means of quantifying these[9,25], which causes methodological circularity[25]. In recent years, researchers have expanded their investigations to examine the possibility of observing the same uncanny valley in zoomorphic robots[26] or virtual animals[27]. Like humanoid robots, zoomorphic robots that combine realistic and nonrealistic features are less preferred[26]. After recent technological advancements, the development of AI, and realistic looking social humanoid robots, such as Nadine[1], Erica[2], and Ishiguro[12,13], it has become essential to define the type of morphological traits that can cause uncanny valley effects. Additionally, several studies have insufficiently controlled the variation in human likeness portrayed in stimulus images, i.e., the nature of the stimuli that elicit the uncanny valley is not well defined or quantified[25]. Therefore, analyzing users' affective states when interacting with robots with different levels of human likeness is necessary. In this study, we consider the robots or agents as mentioned above, which have varying degrees of human likeness in their appearance.

In the past, this hypothesis was primarily validated by allowing participants to view non-morphed[28–30] or morphed images of robots[20,31], video clips of agents performing tasks[16,32,33], or computer-generated models[34]. Because the facial features of a human are an essential characteristic that lend to the realism of the agents, many studies have focused on simply showing virtual faces[31,35] and facial images[30,36] to participants. However, these methods consider only the faces and avoid other possible causes of the uncanny valley, such as the movement of robots. The study reported in [37] showed how attributes, such as skin and body movements, are essential to how humans perceive such agents. Most robots are designed to interact with humans and their environment to accomplish tasks. With social humanoids, the interaction capabilities of robots have become essential. Assessing or studying the uncanny valley would be difficult by viewing only images or video clips, as people don’t interact with these robots or use them for any purpose. In contrast, in this study, we let participants interact live with four different robots of varying human likeness. The participants communicated with the robots directly via Zoom calls as no in-person communication was possible, owing to COVID-19 restrictions. We analyzed human-robot interactions to observe any uncanny valley effects on the participants.

Another critical aspect of this hypothesis is how user-related affinity and eeriness caused by a robot’s appearance are quantified or measured. Several studies have attempted to examine the psychological aspect of the uncanny valley hypothesis[21–23,38] using functional MRI (fMRI) to characterize human behavior and observe the uncanny effect on participants. However, these studies hooked subjects up to bulky MRI machines. While methods such as those in [9,20] scrutinized past studies to find empirical evidence of the uncanny effect, studies reported in [39,40] provided a Bayesian explanation of the observed phenomena. The study reported in [39] concluded that human-looking robots have a huge potential to improve social interactions in individuals with autism. A defined protocol for determining and validating the affective or emotional state of a participant is unavailable. Most of these studies[30] use ad-hoc self-rating scales[10]. Few studies[14,41] have considered valid psychometric and behavioral evaluation methods to study how the human mind perceives human and nonhuman entities. Because we allowed subjects to directly interact with the robots, we recorded interaction via videos, audio, and text (via Zoom with consent from participants). We analyzed these various modalities using state-of-the-art deep learning video, audio, and text emotion and sentiment analysis methods to determine if any negative affective traits were visible during these interactions. In addition to these modalities, participants completed a Godspeed questionnaire[42] for each robot. Using both the questionnaire and multimodal deep learning emotion and sentiment analysis, we quantified the likeability of each robot and scrutinized the uncanny valley phenomena for socially interactive humanoid robots (each robot with a varying level of humanness).

3 Experiment setup

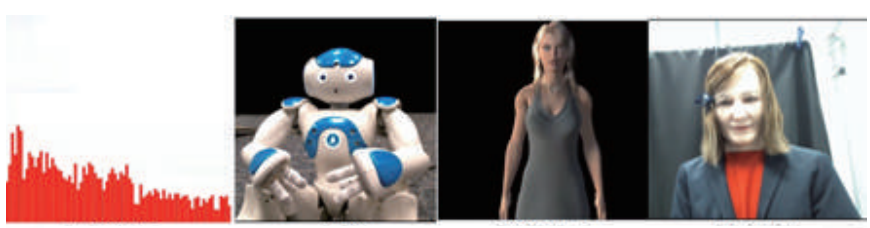
The experiment setup is intended to study the uncanny valley hypothesis for interactive humanoid robots and determine whether we can observe the presence of any uncanny effects from the participants. Unlike previous studies, we allowed participants to interact with the robots (or agents).

We conducted our experiment online over Zoom video calls because of COVID-19 pandemic-related restrictions. On the calls, we first introduced all robots to the participants and explained the nature of the interactions. This information was provided to eliminate any categorical ambiguity that could result in uncanny valley effects. Consent was obtained for audio and video recording during the initial explanation. At the end of the human-robot interaction, the participants were asked to fill out a questionnaire.

Participants interacted with the following four types of robots (or agents) in the same scenario:

1. Maya, the voice assistant;
2. Nao, the child-sized humanoid robot;
3. Nicole, the virtual human with human-like appearance;
4. Nadine, the complete life-size humanoid social robot with skin, articulated hands, and other human-like features.

Figure 1 shows these human-like agents/entities. All robots share the same architecture. Please refer to [1] for more information on the architecture and how the input information is processed. The robots varied only in their physical appearances, as indicated.



Mava (Voice Assistant) Nao Robot Nicole (Virtual Human) Nadine Social Robot

Figure 1 The four different robots used in the experiment.

The audio and video of user interactions were recorded for analysis. Despite the lack of physical presence-related constraints during the conversation sessions, we ensured fluid dialogues between the participants and robots. During these interactions, participants were free to interact with the robots on a one-to-one basis in any form, which covered both verbal and non-verbal aspects. They could ask and discuss any topic of their choice. The humanoid robots, Nao and Nadine, were capable of greeting and waving at the participants owing to their physical capabilities. Nadine's controller generated physical gestures, whereas Nicole's controller generated animated gestures. Nadine and Nicole could both emote expressions with lip synchronization, gaze at the participants, and generate gestures based on the context of the conversations. Because Maya was a voice assistant, physical or virtual nonverbal forms of communication were nonexistent.

The robots were placed in an isolated room, where external noise was excluded, and a black background was used to enhance the participants′ attention. All conversations were conducted in English. At the end of the study, five sets of questionnaires were completed by the participants to obtain their overall feedback and experience about their one-to-one sessions with the robots.

4 Data collection and analysis

To obtain a holistic idea of the effect of each of the robot (or agent) interaction sessions with the participants, we recorded every session and analyzed the videos of 77 participants. The objective tools are based on natural language processing and computer vision techniques and use state-of-the-art deep neural networks (DNNs) to automatically evaluate the mental states of the participants during conversations. The subjective tools consisted of five questionnaires for each robot, including a generic survey targeted at the participants. Furthermore, we performed a statistical analysis to draw meaningful comparisons. In our analysis, we aimed to identify the uncanny valley by measuring the participants' eerie or creepy experiences. Studies such as [14,43] have experimentally shown that negative emotions such as fear, shock, disgust, anxiety, and nervousness are associated with eeriness. The fMRI samples shown in [14] have been proven to show correlations between the uncanny valley and such emotions. In the same manner, we used these objective and subjective analysis tools to identify negative emotions during human-robot interactions and used them as evidence of the uncanny valley.

4.1 Video analysis

Non-verbal cues, such as expressions and gestures, and speech cues are equally important in determining the engagement in conversation[44]. To recognize the facial emotions of participants in the video, we first detected faces using a technique based on a convolutional neural network (CNN) with Dlib[1]. Using ResNet-50[45] as the backbone, we trained an emotion recognizer with eight expression classes: neutral, happy, sad, surprise, fear, disgust, anger, and contempt. We then used the recognizer for facial expression detection on the video frames. Our emotion recognizer was trained on the largest in-the-wild facial expression dataset called AffectNet[46] (with approximately 320000 images excluding none, uncertain, and non-face categories) until the network converged. For the analysis, the model provided confidence levels for each emotion observed on every detected face in each video frame. Therefore, we defined the average emotions displayed by a participant during interaction in a complete video stream as follows:

$$\langle emotion \rangle = \frac{1}{L}\sum_{l=1}^{L} emotion_l \tag{1}$$

where $emotion_l \in [0, 1]$ denotes the probability of a participant's detected facial emotion in the l-th frame belonging to each of the eight classes which were estimated by our emotion recognizer. Figure 2 shows some of the emotions detected in the frames. We used statistical methods to evaluate the differences in interactions with different agents (or robots) and compared them.

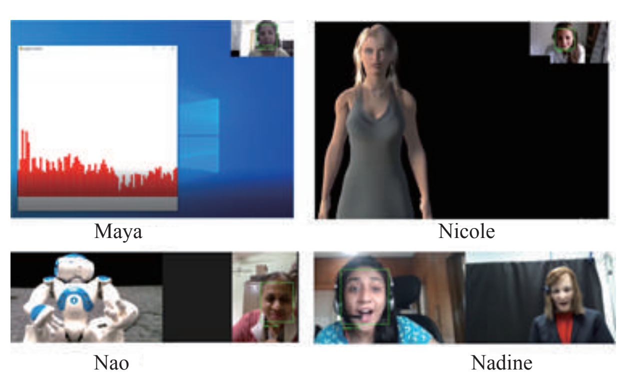


Figure 2 Different emotions detected in the video analysis.

4.2 Audio analysis

A subject's audio or speech pattern is another direct indication of their emotional state. Audio data can contain several explicit and subtle cues that reflect a subject′s mental state. Although numerous audio

[1]http://dlib.net/

emotion recognition investigations have been conducted, we encountered two issues when implementing them. First, several studies had no pretrained models[47,48]. Second, such studies did not focus on a set of emotions that could be useful in completely studying and evaluating the uncanny valley. For instance, the study reported in[49] classified only five emotions and identified gender. Therefore, we used an implementation[2] to train the audio emotion classifier. An RNN-based deep learning model was adopted and trained using four different datasets: RAVDESS[50], TESS[51,52], EmoDB, and a custom dataset. Our model included two RNN layers and two dense layers each having 128 units. We ran a train-test-validation cycle on these datasets to obtain the final model. The model considered nine different emotions: “neutral”, “calm”, “happy”, “sad”, “angry”, “fear”, “disgust”, “ps” (pleasant surprise), and “boredom”. These varied emotions were selected because they allowed us to better gauge the uncanny valley effect. An entire audio recording was analyzed, and the intensity of each emotion observed was provided by the model.

4.3 Text analysis

For text analysis, we converted the obtained audio files into text using Google Speech-to-Text[3]. To identify emotions in context during conversation between participants and agents, we used the emotion categorization model in SenticNet[53]. This API uses the hourglass of emotions[54], a biologically inspired and psychologically motivated emotion categorization model for sentiment analysis, in conjunction with SenticNet and deep learning to extract emotion labels from texts. The input of this API is a piece of text (a sentence or paragraph), and the output is a list of emotion labels. Dominant emotions were determined using statistical analysis based on the data collected for each agent.

4.4 Questionnaire

The Godspeed questionnaires defined in[42] were used to assess participants′ perceptions of the four robots. The Godspeed questionnaire measures five factors: perceived anthropomorphism, animacy, likeability, intelligence, and safety[4]. There were 24 semantic differential items for these five indices. Participants were required to rate their impressions of each robot according to these 24 semantic traits, which made this a comprehensive survey. We used the collected survey data to evaluate participants′ impression of each agent and the agent′s effect on the five indices. The survey was conducted online using Survey Monkey. In total, there were five different surveys–four surveys for each of the robots and a generic questionnaire to collect demographic information about the participants, such as age, education status, and prior experience with robots. We formulated a 100-point scale for comprehensive subjective scoring. The scores assigned by participants for every item were used to validate the relationships or comparisons captured using the video, audio, and text data. Additionally, our survey included general questions regarding all robots (Table 1).

Table 1 General Questionnaire

Survey Questionnaire
Have you interacted with Robots before?
Have you interacted with virtual characters?
Have you interacted with Voice assistants before?
Which robot/agent do you think is most human-like?
Which robot/agent did you like most?
Did the robot/agent’s human-like appearance affect your interaction positively?
Did the robot/agent’s human-like voice affect your interaction positively?

5 Results

In this section, we examine the results for each analysis method: video, audio, text, and survey questionnaire analyses. Based on these evaluations, we provide insights into what each modality reveals regarding parti-

[2]https://github.com/x4nth055/emotion-recognition-using-speech

[3]https://cloud.google.com/speech-to-text/

[4]https://www.bartneck.de/2008/03/11/the-godspeedquestionnaire-series/

cipants′ emotions and affective states during their interactions with our agents. The collected data for each modality were statistically analyzed to determine the emotions observed during interactions with each agent.

5.1 Video analysis results

Figure 3 shows the average emotions expressed by participants during their encounters with each robot. We performed repeated measures analysis of variance (ANOVA) tests, to determine the presence or absence of significant differences between emotions regarding the robots on each of the variables. Because the same facial expressions were tracked for each of the four robots, the scores for the emotions expressed for each of them were considered as independent variables.

When variables violated the assumptions of sphericity, the Greenhouse-Geisser modification[55] for degrees of freedom was used. Table 2 presents the ANOVA results.

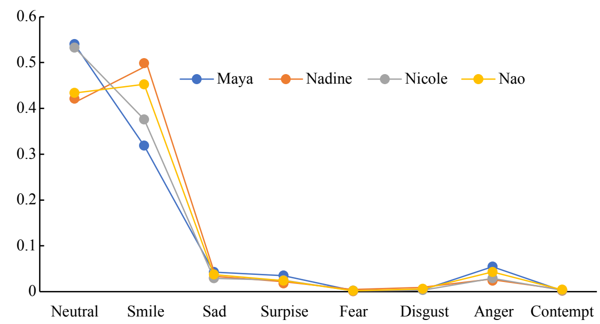


Figure 3 Average levels of each emotion observed from facial expressions over the ensemble of experiments for each of the robots.

Table 2 Results of repeated measures ANOVA tests for all emotions; video-based facial expressions analysis

Emotion	Degree of Freedom (df)	F statistic (F)	Value of test (p)
Neutral	(2.547,152.824)	7.655	0.000
Smile	(2.429,145.715)	15.882	0.000
Sad	(1.882,112.910)	2.612	0.081
Surprise	(2.625,157.494)	2.752	0.052
Fear	(2.201,132.045)	0.791	0.466
Disgust	(2.402,144.146)	0.419	0.695
Anger	(2.286,137.133)	2.944	0.049
Contempt	(1.351,81.085)	0.711	0.442

5.2 Audio analysis results

Figure 4 shows the average emotions of participants for each robot. Like the video modality, repeated measures ANOVA tests were conducted for the audio modality and the Greenhouse-Geisser modification[55] was applied as necessary. Table 3 lists the results from these ANOVA tests and shows significant differences in all emotions between the robots. Thus, post-hoc analyses were conducted for all outcomes to determine the specific differences in the emotions elicited by the robots.

・The post-hoc test for anger showed that it was the highest for interactions with Maya, significantly different from the rest (p<0.001 for all three pairs). The other robots did not differ in this measure (p-values ranging from 0.530 to 0.962).

・ The post-hoc test for boredom showed that it was also the highest for interactions with Maya, which was significantly different from the other three (p-values ranging from 0.001 to 0.002). Nicole differed marginally from Nao (p=0.046) but did not differ from Nadine (p=0.380). Nao and Nadine showed no significant differences (p=0.160).

・ The post-hoc test for calmness showed that Maya incited the lowest levels, which were significantly different from the other three (p-values

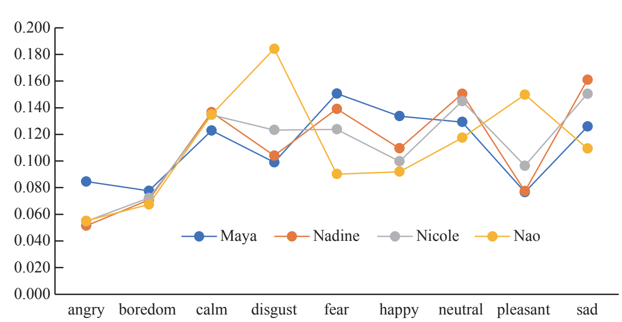


Figure 4 Different emotions derived from audio recordings.

Table 3 Results of repeated measures ANOVA tests for all emotions; audio recordings analysis

Emotion	Degree of freedom (df)	*F* statistic (*F*)	Value of test (*p*)
Angry	(2.636,168.711)	11.498	0.000
Bored	(2.186,139.885)	7.244	0.001
Calm	(3, 192)	6.224	0.000
Disgust	(3, 192)	21.969	0.000
Fear	(2.599,166.348)	18.895	0.000
Happy	(2.683,171.731)	16.043	0.000
Neutral	(3, 192)	20.514	0.000
ps	(3, 192)	24.299	0.000
Sad	(3, 192)	41.499	0.000

ranging from 0.001 to 0.003). The other three robots showed no significant differences (p-values ranging from 0.553 to 0.943).

・ The post-hoc test for disgust showed that it was expressed the most towards Nao (p<0.001). The other three robots had no significant differences in this emotion (p-values ranging from 0.072 to 0.628).

・The post-hoc tests for fear showed that Nao's scores were the lowest, statistically different from the remaining robots (p<0.001 for all pairs). The next lowest was Nicole, marginally different from Nadine (p=0 .049) and significantly different from Maya (p=0.012). Nadine and Maya did not differ significantly (p=0.210).

・ The post-hoc tests for happiness showed that Maya incited the most of this emotion, significantly more than the others (p<0.001 for all pairs). Nadine was significantly higher than Nao (p=0.002) but not higher than Nicole (p=0.082). Nicole and Nao did not significantly differ (p=0.217).

5.3 Text analysis results

Figure 5 presents the sentiment analysis results. The most prevalent emotions were ecstasy, enthusiasm, and delight. There were numerical differences in the prevalence of emotions towards the various robots for each emotion, but these differences could not be tested statistically because of the nature of the data collection. However, they were useful in observing the ranking of the robots for each emotion scrutinized.

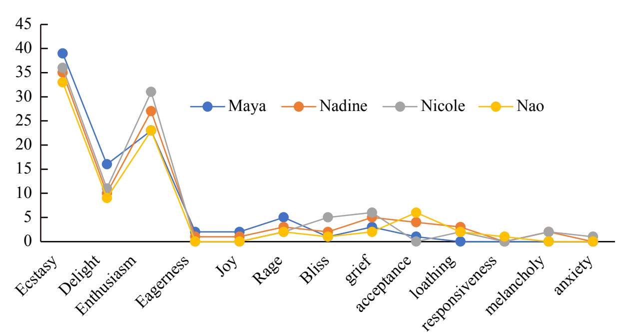


Figure 5 Number of detections of each emotion towards each robot in the spoken text.

Maya elicited the most ecstasy and delight, followed by Nicole, Nadine, and Nao. In terms of enthusiasm, Nicole had the highest scores, followed by Nadine. The third highest were Nao and Maya with the same number of occurrences. Rage was shown most towards Maya, with the other three agents incurring equal amounts. Bliss was mostly directed in Nicole's regard, with less of it directed towards the others. Grief was primarily shown in interactions with Nicole and Nadine and not so much with Maya and Nao. Acceptance was the highest for Nao and lowest for the rest. Eagerness, joy, loathing, responsiveness, melancholy, and anxiety were rarely displayed (i.e., a total of five times for all robots combined).

5.4 Questionnaire results

Figure 6 shows the participants' average scores on the five scales for the four robots. To establish the presence or absence of significant differences among the robots on each scale, five ANOVAs and a post-hoc test were conducted to determine whether specific pairs of robots differed among themselves. As listed in Table 4, all five multiple measure ANOVAs were significant at the <0.01 level, and all but the perceived safety scale were significant at the <0.001 level.

This indicates that the differences between the robots are significant for each scale. To determine which robots elicited the differences, we conducted post hoc analyses and obtained the following results:

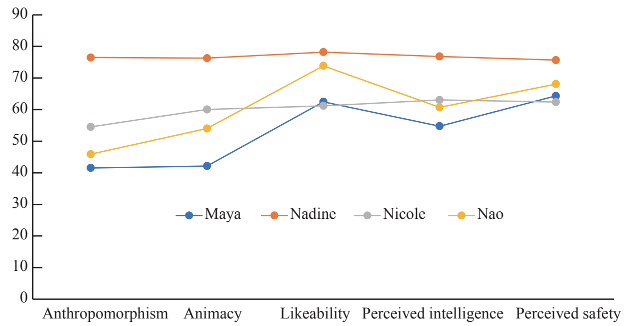


Figure 6 Average scores on each of the five scales for each of the four robots.

・ The post-hoc analyses for anthropomorphism showed that Nadine had the highest scores, which were significantly different from those of the other robots (all $p<0.001$). Nicole′s scores were the second-highest, considerably different from those of Maya ($p=0.004$) and marginally indifferent from Nao′s score ($p=0.061$). Nao and Maya′s scores were insignificantly different ($p=0.314$).

Table 4 Multiple ANOVA results for the five scales

Emotion	Degree of freedom(df)	*F* statistic (*F*)	Value of test (*p*)
Anthropomorphism	(3, 161)	25.295	0.000
Animacy	(3, 161)	18.513	0.000
Likeability	(3, 164)	6.701	0.000
Perceived intelligence	(3, 160)	12.108	0.000
Perceived safety	(3, 164)	5.218	0.002

・ The post-hoc analyses for animacy were also higher for Nadine than for the other robots ($p<0.001$). Nicole′s animacy scores were significantly higher than those of Maya ($p<0.001$) but not from those of Nao ($p=0.215$). Finally, Nao′s score was significantly higher than that of Maya′s ($p<0.001$).

・ The post-hoc analyses for likeability showed that Nadine′s scores were significantly higher than those of Maya and Nicole ($p<0.001$) but not higher than those of Nao ($p=0.351$). Nao also had significantly different scores than Maya and Nicole′s ($p=0.013$ and 0.008, respectively). The scores of Maya and Nicole did not differ significantly ($p=0.779$).

・ The post-hoc tests for perceived intelligence showed that Nadine′s scores were significantly higher than those of the others ($p<0.001$). Nicole′s scores were significantly higher than those of Maya ($p=0.029$) but not higher than those of Nao ($p=0.551$). There were no significant differences between the scores of Nao and Maya ($p=0.116$).

・ The post-hoc tests for perceived safety showed that Nadine had the highest scores that were significantly different from those of the other robots (*p*-values ranging from 0.001 to 0.04). The remaining robots did not differ in this measure (*p*-values ranging from 0.125 to 0.587).

The participants also completed a general questionnaire regarding their past experiences with robots; 45% of them had previous experience with robots, 60% had previous experience with virtual characters, and 95% had previous experience with voice assistants. They selected Nadine as the most human-like robot, as shown in Figure 7a. For the “most liked robot”, most participants also gave Nadine the highest likeability. However, as shown in Figure 7b, the differences between all robots were not substantial.

Finally, the participants rated how the dimensions of human-like appearances, gestures, voice, and facial expressions impacted the quality of their interactions. As shown in Figure 8, participants rated

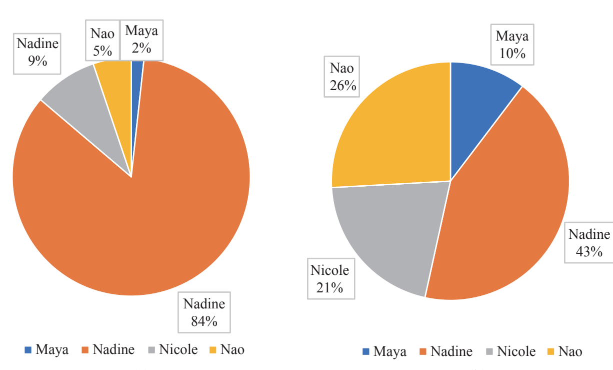


Figure 7 (a) Most human-like robot based on participant perception. (b) Most liked robot.

all features as important.

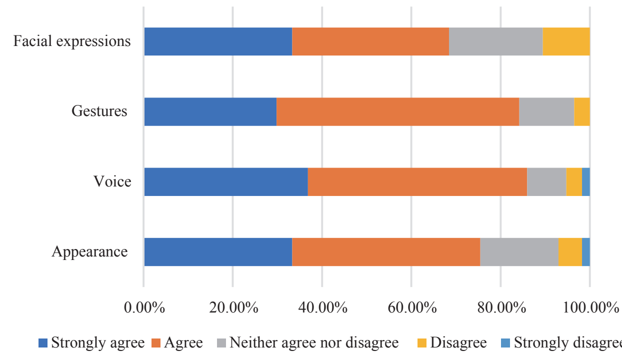


Figure 8 Importance of the various human-like characteristics of the robots.

6 Conclusion and discussion

The results show that all four robots were perceived in various manners and that the emotions expressed varied in their regard. As shown in Figure 6, Nadine was chosen as the "favorite" robot; she was seen as the most anthropomorphic, animatic, intelligent, and safe. The only characteristic that she failed to top was likeability, where she tied with Nao. Furthermore, Maya was seen as the least anthropomorphic and animate, which is predictable given that she is only a voice assistant. Nao and Nicole scored the same on anthropomorphism and animacy. However, Nao was significantly more likeable, which indicated that a physical body may elicit a higher degree of likeability. Findings from the multi-modal analysis and surveys do not clearly show any effect of the "uncanny valley"[56]. Nadine, the most humanoid robot, was seen as the most likable robot, with similar findings obtained through the sentiment and facial expression analyses. Nao, the humanoid-but-toyish robot, was also seen as likable and provoked the highest positive surprise. However, it incurred high disgust, as determined through the analyses of audio data, without evoking many emotions. However, while being the most likable, Nadine generated the most sadness and was the second most feared. This fear is possibly indicative of the "uncanny valley" effect. Nonetheless, with the "uncanny valley," eeriness would be expected to increase with the increasing degree of anthropomorphism, which was not the case. The highest fear was expressed towards Maya, the voice assistant without visual characteristics. Furthermore, owing to the human-like appearance of Nadine, most participants were tentative and fearful as they wanted to impress her. This fear can be considered good, as it indicates that people want to connect with her. Extending our multimodal analysis to other cues such as pose estimation[57,58], body language cues[59–61] could be considered for detecting the evidence of an "uncanny valley".

While Nao had a toy-like, childish appeal, Nadine had a human-resembling body, which was clearly sufficient for increasing her likability compared to a bodyless or virtual agent. However, all robots were well-liked. The lowest-ranked robot scored above 60/100, which indicated that no "uncanny valley" effect was determined in this study. Furthermore, the lack of a correlation between being more anthropomorphic and less likeable and provoking more negative emotions provides evidence against the hypothesis[10]. Compared to previous research, the robots used in this investigation might not have provoked the uncanny valley effect, as they have been carefully and coherently designed and constructed. As shown in [9,10], the uncanny valley is triggered by notable characteristic traits or aspects (for example, the non-human characteristics of agents with a human appearance and vice-versa).

The results showed that the robots incited different emotions, but the most anthropomorphic robots were the most liked. Furthermore, all robots were well-liked, and there was no correlation between the anthropomorphism of the robots and the negative emotions they provoked. Therefore, this study, like previous studies[9,10,25], did not observe or substantiate the uncanny valley hypothesis. This could be because of the characteristics of these specific robots or the specifics of today's world in which both humanoid and non-humanoid robots are becoming increasingly prevalent and people are accustomed to them. Regardless, the future design of interactive robots should be open to creating anthropomorphic robots, while ensuring a coherent design.

Declaration of competing interest

We declare that we have no conflict of interest.

Acknowledgement

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore. We thank our colleague Tham Yiep Soon for his strong technical support during the study.

References

1 Ramanathan M, Mishra N, Thalmann N M. Nadine humanoid social robotics platform. In: Advances in Computer Graphics. Springer International Publishing, 2019, 490–496
DOI: 10.1007/978-3-030-22514-8_49

2 Pranav D. A robot called erica set to become news anchor in Japan. 2018

3 Mishra N, Ramanathan M, Satapathy R, Cambria E, Magnenat-Thalmann N. Can a humanoid robot be part of the organizational workforce? A user study leveraging sentiment analysis. In: 2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN). 2019
DOI:10.1109/ro-man46459.2019.8956349

4 Mishra N, Ramanathan M, Satapathy R, Cambria E, Magnenat-Thalmann N. Can a humanoid robot be part of the organizational workforce? A User Study Leveraging Sentiment Analysis. 2019

5 Mishra N, Baka E, Magnenat Thalmann N. Exploring potential and acceptance of socially intelligent robot. In: Intelligent Scene Modeling and Human-Computer Interaction. Springer International Publishing, 2021, 259–282
DOI:10.1007/978-3-030-71002-6_15

6 Tulsulkar G, Mishra N, Thalmann N M, Lim H E, Lee M P, Cheng S K. Can a humanoid social robot stimulate the interactivity of cognitively impaired elderly? A thorough study based on computer vision methods. The Visual Computer, 2021, 37(12): 3019–3038
DOI: 10.1007/s00371-021-02242-y

7 Mishra N, Tulsulkar G, Li H H, Thalmann N M. Does elderly enjoy playing bingo with a robot? A case study with the humanoid robot Nadine. 2021

8 Mori M. Bukimi no tani [the uncanny valley]. Energy, 1970, 7: 33–35

9 Kätsyri J, Förger K, Mäkäräinen M, Takala T. A review of empirical evidence on different uncanny valley hypotheses: support for perceptual mismatch as one road to the valley of eeriness. Frontiers in Psychology, 2015, 6390
DOI: 10.3389/fpsyg.2015.00390

10 Cheetham M. Editorial: the uncanny valley hypothesis and beyond. Frontiers in Psychology, 2017, 81738
DOI: 10.3389/fpsyg.2017.01738

11 Eaton M. Humanoid robots, their simulators, and the reality gap. In: Evolutionary Humanoid Robotics. Part of SpringerBriefs in Intelligent Systems (Artificial Intelligence, Multiagent Systems, and Cognitive Robotics). 2015

12 Minato T, Shimada M, Ishiguro H, Itakura S. Development of an android robot for studying human-robot interaction. Innovations in Applied Artificial Intelligence, 2004

13 Guizzo E. Hiroshi ishiguro: The man who made a copy of himself, 2010

14 Ho C C, MacDorman K F. Revisiting the uncanny valley theory: developing and validating an alternative to the Godspeed indices. Computers in Human Behavior, 2010, 26(6): 1508–1518
DOI: 10.1016/j.chb.2010.05.015

15 Seyama J, Nagayama R S. The uncanny valley: effect of realism on the impression of artificial human faces. Presence: Teleoperators and Virtual Environments, 2007, 16(4): 337–351
DOI: 10.1162/pres.16.4.337

16 Ho C C, MacDorman K F. Measuring the uncanny valley effect. International Journal of Social Robotics, 2017, 9(1): 129–139
DOI: 10.1007/s12369-016-0380-9

17 Softbank Robotics. Nao the humanoid and programmable robot, 2021

18 Tinwell A, Grimshaw M, Nabi D A, Williams A. Facial expression of emotion and perception of the Uncanny Valley in virtual characters. Computers in Human Behavior, 2011, 27(2): 741–749
DOI: 10.1016/j.chb.2010.10.018

19 Tinwell A. The uncanny valley in games and animation. Boca Raton: CRC Press, 2014

20 Palomäki J, Kunnari A, Drosinou M, Koverola M, Lehtonen N, Halonen J, Repo M, Laakasuo M. Evaluating the replicability of the uncanny valley effect. Heliyon, 2018, 4(11): e00939
DOI: 10.1016/j.heliyon.2018.e00939

21 Pollick F. Rearch of the Uncanny valley. 2009, 40(12): 69–78

22 Cheetham M, Suter P, Jäncke L. The human likeness dimension of the "uncanny valley hypothesis": Behavioral and functional MRI findings. Frontiers in Human Neuroscience, 2011, 5126
DOI: 10.3389/fnhum.2011.00126

23 Saygin A P, Chaminade T, Ishiguro H, Driver J, Frith C. The thing that should not be: predictive coding and the uncanny valley in perceiving human and humanoid robot actions. Social Cognitive and Affective Neuroscience, 2012, 7(4): 413–422
DOI: 10.1093/scan/nsr025

24 Laakasuo M, Palomäki J, Köbis N. Moral uncanny valley: a robot's appearance moderates how its decisions are judged. International Journal of Social Robotics, 2021, 13(7): 1679–1688
DOI: 10.1007/s12369-020-00738-6

25 Lay S, Brace N, Pike G, Pollick F. Circling around the uncanny valley: design principles for research into the relation between human likeness and eeriness. i-Perception, 2016, 7(6): 204166951668130
DOI: 10.1177/2041669516681309

26 Löffler D, Dörrenbächer J, Hassenzahl M. The uncanny valley effect in zoomorphic robots: the U-shaped relation between animal likeness and likeability. Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction. Cambridge, United Kingdom. New York, ACM, 2020, 261–270
DOI:10.1145/3319502.3374788

27 Schwind V, Leicht K, Jäger S, Wolf K, Henze N. Is there an uncanny valley of virtual animals? A quantitative and qualitative investigation. International Journal of Human-Computer Studies, 2018, 11149–11161
DOI: 10.1016/j.ijhcs.2017.11.003

28 Bartneck C, Kanda T, Ishiguro H, Hagita N. Is the uncanny valley an uncanny cliff? 2007, 9: 368–373

29 MacDorman K F, Ishiguro H. The uncanny advantage of using androids in cognitive and social science research. Interaction Studies. Social Behaviour and Communication in Biological and Artificial Systems, 2006, 7(3): 297–337
DOI: 10.1075/is.7.3.03mac

30 Looser C E, Wheatley T. The tipping point of animacy. Psychological Science, 2010, 21(12): 1854–1862
DOI: 10.1177/0956797610388044

31 MacDorman K F, Green R D, Ho C C, Koch C T. Too real for comfort? Uncanny responses to computer generated faces. Computers in Human Behavior, 2009, 25(3): 695–710
DOI: 10.1016/j.chb.2008.12.026

32 MacDorman K F. Subjective ratings of robot video clips for human likeness, familiarity, and eeriness: an exploration of the uncanny valley. In: ICCS/CogSci-2006 Long Symposium: Toward Social Mechanisms of Android Science. 2006

33 Walters M L, Syrdal D S, Dautenhahn K, te Boekhorst R, Koay K L. Avoiding the uncanny valley: robot appearance, personality and consistency of behavior in an attention-seeking home scenario for a robot companion. Autonomous Robots, 2008, 24(2): 159–178
DOI: 10.1007/s10514-007-9058-3

34 Łupkowski P, Rybka M, Dziedzic D, Włodarczyk W. Human-likeness assessment for the uncanny valley hypothesis. Bio-Algorithms and Med-Systems, 2017, 13(3): 125
DOI: 10.1515/bams-2017-0008

35 McDonnell R, Breidt M. Face reality: investigating the uncanny valley for virtual faces. ACM SIGGRAPH ASIA 2010 Sketches. Seoul, Republic of Korea. New York, ACM, 2010, 1–2
DOI: 10.1145/1899950.1899991

36 Stephanie Claire Lay. The uncanny valley effect. The Open University, 2015

37 Silvera-Tawil D, Garbutt M. The far side of the uncanny valley: 'healthy persons', androids, and radical uncertainty. In: 2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), 2015

38 Rosenthal-von der Pütten A M, Krämer N C, Maderwald S, Brand M, Grabenhorst F. Neural mechanisms for accepting and rejecting artificial social partners in the uncanny valley. The Journal of Neuroscience, 2019, 39(33): 6555–6570
DOI: 10.1523/jneurosci.2956-18.2019

39 Ueyama Y. A Bayesian model of the uncanny valley effect for explaining the effects of therapeutic robots in autism spectrum disorder. PLOS ONE, 2015, 10(9): e0138642
DOI: 10.1371/journal.pone.0138642

40 Moore R K. A Bayesian explanation of the 'Uncanny Valley' effect and related psychological phenomena. Scientific Reports, 2012, 2(1): 864
DOI: 10.1038/srep00864

41 Cheetham M, Wu L, Pauli P, Jancke L. Arousal, valence, and the uncanny valley: psychophysiological and self-report findings. Frontiers in Psychology, 2015, 6981
DOI: 10.3389/fpsyg.2015.00981

42 Bartneck C, Kulić D, Croft E, Zoghbi S. Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots. International Journal of Social Robotics, 2009, 1(1): 71–81
DOI: 10.1007/s12369-008-0001-3

43 Ho C C, MacDorman K F, Pramono Z A D D. Human emotion and the uncanny valley: a GLM, MDS, and Isomap analysis of robot video ratings. In: Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction. Amsterdam, The Netherlands. New York, ACM, 2008, 169–176
DOI: 10.1145/1349822.1349845
44 Ramanathan M, Satapathy R, Magnenat Thalmann N. Survey of speechless interaction techniques in social robotics. In: Intelligent Scene Modeling and Human-Computer Interaction. Springer International Publishing, 2021, 241–257
DOI: 10.1007/978-3-030-71002-6_14
45 He K M, Zhang X Y, Ren S Q, Sun J. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA, IEEE, 770–778
DOI: 10.1109/cvpr.2016.90
46 Mollahosseini A, Hasani B, Mahoor M H. AffectNet: a database for facial expression, valence, and arousal computing in the wild. IEEE Transactions on Affective Computing, 2019, 10(1): 18–31
DOI: 10.1109/taffc.2017.2740923
47 Aouani H, Ayed Y B. Speech emotion recognition with deep learning. Procedia Computer Science, 2020, 176251–176260
DOI: 10.1016/j.procs.2020.08.027
48 Cunningham S, Ridley H, Weinel J, Picking R. Supervised machine learning for audio emotion recognition. Personal and Ubiquitous Computing, 2021, 25(4): 637–650
DOI: 10.1007/s00779-020-01389-0
49 Trigeorgis G, Ringeval F, Brueckner R, Marchi E, Nicolaou M A, Schuller B, Zafeiriou S. Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing. Shanghai, China, IEEE, 5200–5204
DOI: 10.1109/icassp.2016.7472669
50 Livingstone S R, Russo F A. The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): a dynamic, multimodal set of facial and vocal expressions in North American English. PLOS ONE, 2018, 13(5): e0196391
DOI: 10.1371/journal.pone.0196391
51 Dupuis K, Pichora-Fuller M. Toronto emotional speech set (TESS)-Younger talker_Neutral, 2010
52 Pichora-Fuller M K, Dupuis K. Toronto emotional speech set (TESS). Scholars Portal Dataverse, 2020
53 Susanto Y, Livingstone A G, Ng B C, Cambria E. The hourglass model revisited. IEEE Intelligent Systems, 2020, 35(5): 96–102
DOI: 10.1109/MIS.2020.2992799
54 Cambria E, Livingstone A, Hussain A. The hourglass of emotions. In: Cognitive behavioural systems, 2012
55 Greenhouse S W, Geisser S. On methods in the analysis of profile data. Psychometrika, 1959, 24(2): 95–112
DOI: 10.1007/bf02289823
56 Mori M, MacDorman K F, Kageki N. The uncanny valley. IEEE Robotics & Automation Magazine, 2012, 19(2): 98–100
57 Kamel A, Sheng B, Li P, Kim J, Feng D D. Hybrid refinement-correction heatmaps for human pose estimation. IEEE Transactions on Multimedia, 2021, 231330–231342
DOI: 10.1109/tmm.2020.2999181
58 Ramanathan M, Yau W Y, Teoh E K. Improving human body part detection using deep learning and motion consistency. In: 2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV). Phuket, Thailand, IEEE, 2016, 1–5
DOI: 10.1109/icarcv.2016.7838651
59 Ramanathan M, Yau W, Teoh E. Human posture detection using H-ELM body part and whole person detectors for human-robot interaction. 2016
60 Aouaidjia K, Sheng B, Li P, Kim J, Feng D D. Efficient body motion quantification and similarity evaluation using 3-D joints skeleton coordinates. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2021, 51(5): 2774–2788
DOI: 10.1109/TSMC.2019.2916896
61 Ertugrul E, Li P, Sheng B. On attaining user-friendly hand gesture interfaces to control existing GUIs. Virtual Reality & Intelligent Hardware, 2020, 2(2): 153–161
DOI: 10.1016/j.vrih.2020.02.001

Quantifying the Reality Gap in Robotic Manipulation Tasks

Jack Collins[1,2], David Howard[2] and Jürgen Leitner[1,3]

***Abstract*— We quantify the accuracy of various simulators compared to a real world robotic reaching and interaction task. Simulators are used in robotics to design solutions for real world hardware without the need for physical access. The 'reality gap' prevents solutions developed or learnt in simulation from performing well, or at at all, when transferred to real-world hardware. Making use of a Kinova robotic manipulator and a motion capture system, we record a ground truth enabling comparisons with various simulators, and present quantitative data for various manipulation-oriented robotic tasks. We show the relative strengths and weaknesses of numerous contemporary simulators, highlighting areas of significant discrepancy, and assisting researchers in the field in their selection of appropriate simulators for their use cases. All code and parameter listings are publicly available from: https://bitbucket.csiro.au/scm/~col549/quantifying-the-reality-gap-in-robotic-manipulation-tasks.git.**

I. Introduction

Simulators are widely used in the robotics community as they allow for real world systems to be quickly and cheaply prototyped without the need for physical access to hardware. Although used throughout robotics as a whole, simulators are particularly amenable to usage in robotic learning research.

The advent of data-hungry Deep Learning approaches, particularly Reinforcement Learning, heavily employ simulation to overcome the high costs intrinsic to repeated real-world data collection experiments, as well as obviating potential damage to expensive hardware during the early stages of learning. Simulated environments harness increasingly powerful and ubiquitous compute resources to cheaply and quickly generate synthetic data to accelerate the learning process. The use of simulation over reality carries numerous advantages, namely:

- No wear or damage to real-world hardware.
- Many instantiations of a simulation can run in parallel;
- (Often) Faster than real-time operation;
- Instant access to robots without having to purchase; and
- Human intervention is not required;

However, these benefits come with downsides; primarily that there are discrepancies between simulations and the real world brought about by the necessity to abstract, approximate, or remove certain physical phenomena, which prevents control systems created in simulation from performing to the same standard in reality. Learning-based approaches are known to exploit situations and achieve goals which are

This research was supported by a Data61 PhD Scholarship. This research was further supported by the Australian Research Council Centre of Excellence for Robotic Vision (project number CE140100016).

[1] Queensland University of Technology (QUT), Brisbane, Australia
[2] Data61/CSIRO, Brisbane, Australia
[3] Australian Centre for Robotic Vision (ACRV)

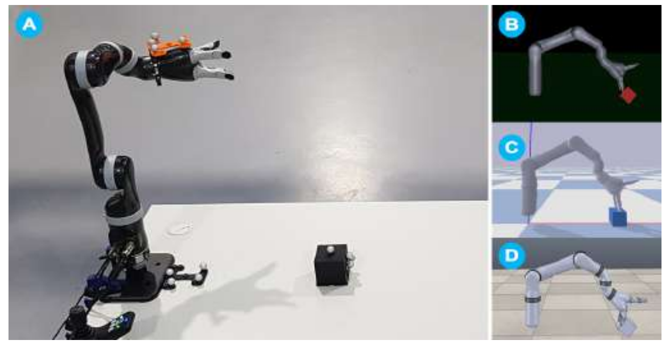


Fig. 1. Real-World and simulated environments with the Kinova arm and cube visualised. (A) Real-world setup with tracking markers attached to dynamic elements; (B) the MuJoCo environment; (C) the PyBullet environment; (D) the V-Rep environment.

simulated artefacts and not realistically plausible in the real world [1], further complicating the transfer to real world robotic applications. This disparity is a prominent issue with recent efforts in sim-to-real learning [2], [3], as well as in Evolutionary Robotics (where simulations are crucial to speeding up these iterative, population-based algorithms), where it is referred to as 'Reality Gap' [4]. Gaps mainly relate to actuators (i.e. torque characteristics, gear backlash, ...), sensors (i.e. sensor noise, latencies, and faults), temporal dynamics, and the physics that govern interactions between robots and objects in their environment (i.e. deformable objects, fluid dynamics, ...).

Here we focus on reality gaps found in robotic grasping, which is selected as a 'grand challenge' that is actively harnessing simulation-based learning [5], [6], and is relevant to a vast swathe of application domains, ranging from industrial assembly to assisted living. From a simulation perspective, grasping is particularly challenging as interactions frequently occur between the robot and objects in its environment, which are rarely captured with any real veracity.

With a growing selection of physics engines and simulation environments available to researchers, the 'correct' combination of simulator/physics for a given task is becoming harder to ascertain. It is also becoming more and more important to know where these gaps exist, and how large they are, as a precursor to overcoming them such that simulation and reality more seamlessly meld. It is therefore timely and important to know how accurate these simulators are when performing various tasks, both to appropriately select a simulator for a particular research endeavour.

In this paper, we attempt to *quantify the reality gap*. We do this for a range of robotic manipulation experiments performed by a real 6DOF Kinova Mico2 arm. We simulate the same scenarios across a range of popular simulators and physics engines, and compare the data from the simulation runs to the real movements of the manipulator as recorded by a highly accurate motion capture setup, which we use as a ground truth (Fig. 1).

The question we endeavour to answer is; *to what accuracy can a range of popular robotic simulators replicate real world manipulation-related tasks?* In particular, we ask;

- What are the differences between the chosen physics engines when simulating the same scenario?
- Are there specific *types* of interactions that some simulators can accurately model, compared to other simulators we test?

We provide a detailed statistical analysis of these simulators when approximating movements of the real Kinova arm. Results quantify the disparity between the trajectory of the simulated Kinova arm and the real-world arm, and highlight that certain movements of the arm are more susceptible to misrepresentation in the simulator.

Our work provides novel contributions to several fields of research in robotics, Deep Machine Learning, Evolutionary Robotics and Manipulation to name a few. We supply strong evidence to measure the accuracy of various simulators and physics engines when compared to a real-world ground truth. Additionally, we provide evidence that simulators are able to model the control and kinematics of manipulators accurately, but the dynamic interactions of a simulation remain unsolved. Our research is set to assist fellow researchers in the selection of simulators for their manipulation tasks.

II. Related Work

Robotics is an embodied discipline focused on building systems that act in the physical world. However, for numerous reasons highlighted in Section I, simulation is a key tool to many successful robotic engineering and integration efforts. Simulation is fast, cheap, and allows for rapid prototyping and iteration over the composition and control of a robotic system. These benefits are perhaps most strongly felt when learning is used, due to the data-hungry nature of many contemporary learning approaches. Because simulators necessarily abstract various features (e.g., sensory delays, actuator slop), away from the physical reality, there exists a gap between what is simulated and how the final system performs in the real world. Of course, we can in some situations learn directly on real hardware, however this requires sophisticated learning testbeds [7], [8], [9] and, depending on the amount of data required, may be prohibitive in terms of required resources [10]. Here we focus on simulated efforts to learn.

A. Bridging the Reality Gap

This 'reality gap' is of increasing importance, as current deep learning approaches require a significant amount of data to achieve acceptable performance. Although increased computing power has narrowed this gap by facilitating more complex, high-fidelity simulations [11], the issue is as yet unsolved.

Domain randomisation is a popular technique in robotic vision, whereby a trained model is subjected to randomised inputs (i.e., colour, shading, rendering, camera position, etc.) [12], [13]. Tobin *et al.* [14] employ visual randomisation to teach a manipulator the 3D object position in simulation with a reasonable transfer to the real-world. Such approaches trace their lineage back to the first mention of the 'reality gap', in the context of evolutionary robotics, which found success by introducing sensory noise into a simulator to discourage overspecialisation to simulated artefacts [4].

This sim-to-real transfer problem has recently been tackled by numerous research groups. Earlier approaches mainly highlight the issues around this transfer [15], with more recent efforts proposing solutions, including domain adaptation and Generative Adversarial Networks (GAN) which requires both real world and simulated data [3]. Results showing the early promise of these techniques — using domain adaptation, Bousmalis *et al.* [16] were able to achieve a success rate for real world grasping trained in simulation of 76.7% on a dataset of unseen objects.

An alternative method to randomisation is the optimisation of the simulated environments, with the goal to emulate the real world better. This approach requires real world data for the simulator to be able to fit to the real world observations, making it robot and application specific [17], [18].

There are several methods originating in Evolutionary Robotics that focus on grading a simulation based on the confidence of its prediction in an attempt to avoid poorly simulated scenarios. One such method implemented by Koos *et al.* [19] offers a multi-objective approach that optimises both the fitness and the transferability of controllers. The transferability of a controller is evaluated using a surrogate model generated from data collected from controllers previously transferred to the test robot. Mouret *et al.* [20] state that a promising idea to cross the reality gap is to teach the limits of the simulator to a supervised learning algorithm with access to a real robot. This is then used to provide an accuracy prediction for simulated controllers. They report increased performance of the generated controllers. These scoring methods reduce the Reality Gap, but do so by limiting the simulator to predicting only things that it can accurately calculate, which reduces the applicability of the approach. They also require real-world data recorded directly from the platform to improve the simulation.

Other approaches employ multiple simulators to overcome the biases from a single simulator. Boeing *et al.* [21] created the Physics Abstraction Layer (PAL), a unified interface between multiple physics engines and successfully evolved a PID controller for an Autonomous Underwater Vehicle. More recently Eaton *et al.* [22] evolved behaviour for a Nao robot using first the V-Rep simulator and then for successful controllers the Webots simulator to remove controllers that were exploiting unrealistic scenarios. The evolved controllers showed improved real-world performance after a small amount of human intervention to rectify an instability of the humanoid robot.

B. Physics Engines

There are many physics engines targeting such diverse fields as gaming, movie effects, and robotics. Physics engines are created to model real-world physical properties in computer simulations with properties such as gravity, friction and contacts typically computed. These models are a simplification of the real-world, to compute a reasonable approximation within a restricted time and resource budget.

Reviews of physics engines in the past have proven many times over that no one engine is capable of modelling all scenarios. Boeing *et al.* [23] compared PhysX, Bullet, JigLib, Newton, Open Dynamics Engine (ODE), Tokamak and True Axis; they reported that Bullet performed best overall however no physics engine was best at all tasks. Chung *et al.* [24] likewise found when testing Bullet, Dynamic Animation and Robotics Toolkit (DART), MuJoCo, and ODE, that no one engine performed better at all tasks, stating that for different tasks and different conditions a different physics engine was found to be better. These findings are further corroborated by Gonzalez-Badillo *et al.* [25], who showed that PhysX performs better than Bullet for non-complex geometries but is unable to simulate more complex geometries to the same degree as Bullet.

One aim of our research is to provide a comprehensive study focused specifically around manipulation tasks, which we believe will be useful to the research community given the ongoing popularity of ‘learning to grasp’. Although several other researchers have evaluated physics engines, varying complexity and tasks [26], [25], [27], little research has been done on comparing real-world data to simulated data to draw conclusions as to the accuracy of physics engines and simulators. To our knowledge this is the first research that compares a highly accurate motion capture baseline with modern physics engines and simulators for real-world robot interaction tasks.

C. Simulation Selection

The list of robotic simulators is long, with many niche areas targeted by specific simulators. We are interested in robotic manipulation and looking for mature, well maintained simulators with active communities and good documentation practices to facilitate the development of robotics research. Additionally, we wanted to find a collection of simulators that provided a common programming language interface whilst also providing access to the Robot Operating System (ROS). We were left with the following: V-Rep, MuJoCo and PyBullet. These simulators expose the following 5 physics engines: Bullet, ODE, Vortex, Newton and Mujoco. This range of physics engines and simulators is attractive due to the range and crossover that the simulators afford whilst also providing a mature user interface.

III. METHOD

The setup consists of a Kinova Mico2 6DOF arm with an attached KG-3 gripper. The arm sits on a table, next to a manipulable cube. Simulators use the official Kinova URDF file. In all cases, the Kinova arm is controlled using joint velocities as it allows higher fidelity over the position controller, and avoids the issue of a simulators position controller interfering with the sent motion commands. A basic proportional controller updated at 5Hz is used to control the joints as this rate was feasible for all selected simulators. We perform sets of identical movements, such that each movement happens in each simulator, and on the real arm, and record the results in each case.

Fig. 2. Motion Capture System: 24 cameras fixed on a $8 \times 8 \times 4$ metre gantry records marker position at $100Hz$ to within 1 millimetre accuracy.

A. Real World Ground Truth

A Qualisys motion capture system utilising 24 cameras mounted to a $8 \times 8 \times 4$ metre gantry records the real-world data (Fig. 2). A 0.75×1.8 metre table with a laminated table top acts as the ground plane for all experiments. Four tracking markers were attached to the wrist of the Kinova arm using a 3D printed mount and rigid marker base. Another rigid base with 4 markers lies on the table top flush against the bracket supporting the arm. The global coordinate frame sits on the opposite end of the table, out of reach of the arm. A 3D printed cube, made from ABS plastic, sits on top of the table. The cube is 7.5cm per side, with a weight of 88.4 grams, including 4 tracking markers (Fig. 3).

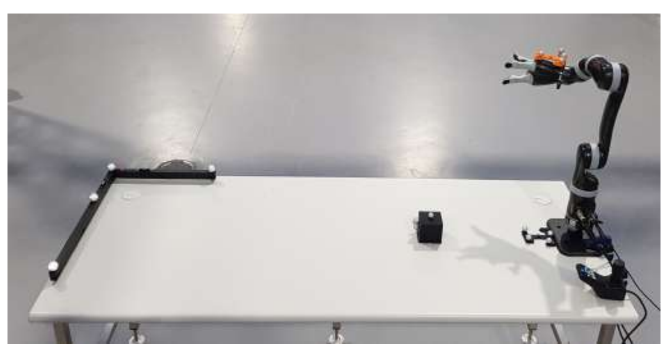


Fig. 3. The real-world setup of the 6DOF Kinova Mico2 Arm with tracking markers attached to wrist via a rigid marker base. On the opposite end of the table sits the L-frame which acts as the global co-ordinate system. The cube with 4 markers attached is also visible.

We recorded 6DOF poses – x, y and z positions and orientation as Euler angles – of the rigid bases with offset rotations and translations for the co-ordinate systems. The co-ordinate system of the wrist mounted base was set to be at the centre of the wrist, analogue to the simulations. The co-ordinate system at the centre of the cube also followed the simulators XYZ frame. The base marker was used as the global co-ordinate system for both the cube and the wrist tracking, allowing for comparative results between simulation and motion tracking.

Control of the Kinova arm was through ROS using the official Kinova package, which supplies joint rotations in degrees and allows for joint velocity commands to be sent at a rate of $100Hz$. Using the same proportional controller and actions as generated in simulation, scenarios were able to be run in python using the ROS interface[1]

B. Simulation

Three leading simulators are compared in our experiments: V-Rep [28], Mujoco [29] and PyBullet [30]. The Kinova arm was imported into each simulator's scene with the cube modelled as a primitive cuboid object. The following points highlight the additional simulator specific changes required after importing the manipulator, with the only shared changes being the starting pose and the starting position (elevated 0.055 metres to account for the Kinova base plate). Other general setup included modelling the weight (0.0884 kilograms), size ($0.075m^3$) and position (0.5,0,0.375) of the cube. All other parameters of the simulations were kept to each simulator's defaults, unless otherwise stated. These include friction models, inertia properties, actuator settings, simulation step sizes, integrators, solvers, etc. The majority of settings are left to their default value as we want to see how well a generic scene can perform without the knowledge of an expert.

1) V-Rep: The scene was imported using V-Reps plug-in and saved as a .ttt binary file after creation. The joint settings were changed to "Lock motor when target velocity is zero".

2) PyBullet: PyBullet's time step was explicitly fixed to the value of 0.01 seconds.

3) MuJoCo: The *m*ujoco-py Python wrapper maintained by OpenAI was used as the interface for MuJoCo. The URDF needed to be converted to an Extensible Markup Language (XML) file with MuJoCo modelling layout; this was done using the MuJoCo compile script. The actuator type and sensors needed to be added manually with the only altered parameter in the XML file being the kv velocity feedback gain. The simulator time step of the simulation was set at 0.0001 seconds as this provided a stable simulation.

IV. Experiments and Results

The experimentation is designed to assess the ability of robotic physics simulations to reproduce real-world scenarios. All experiments are repeated 20 times to ensure reproducible, unbiased results. Data collected for each experiment is limited to the 6DOF pose of the Kinova wrist joint and for one experiment the 6DOF pose of the cube. There are three scenarios in total: (a) beginning with a very basic robotic movement of one joint, (b) moving onto more complex multi-joint movement tasks, and (c) finally an interaction task where the robot arm is pushing the cube along the table.

The motion capture system once calibrated provides accuracy to within 1 millimetre and records 100% "measured" data without the need for interpolation. We therefore consider the motion capture an accurate approximation of the ground truth in our real-world experiments, and the baseline we compare the simulators to.

A. Scene 1: Single Joint Movement

This experiment was designed to compare the control of a single joint of the Kinova arm. For that (Joint #2) rotates from a starting to a final pose, for a rotation of 100 degrees, in about 6 seconds (i.e. 120 control cycles at $5Hz$). All other joints are controlled to the set rotation of 0 degrees.

Fig. 4 presents the results for Scene 1 data as a graph where each plot is the euclidean distance error (Eq. 1) plotted over time. Where $p_{x,y,z}$ are the mean position of the motion capture/physics engine at each time step and $g_{x,y,z}$ is the goal position of the wrist at 100 degrees.

$$e = \sqrt{(p_x - g_x)^2 + (p_y - g_y)^2 + (p_z - g_z)^2} \qquad (1)$$

Most noticeable in Fig. 4 is the lack of results for the ODE physics engine, this is due to the instability of the V-Rep simulation which appears to be caused by self-contacts of the Kinova arm model. The accumulated error for ODE as seen in Table I proves the instability of the physics engine through the comparatively large value; this could not be rectified by tuning the parameters of the simulation. All plotted results begin at the same start position, with Vortex, Newton and PyBullet following the Motion Capture error most closely. This is quantified in Table I where the accumulated error for Vortex, Newton and PyBullet is markedly lower then the other physics engines.

The convergence of the physics engines to the goal position is also of note, as Bullet283 and Bullet278 arrive approximately 1 second earlier than all other plots. Bullet283 and Bullet278 also oscillate noticeably before reaching their final state (Fig. 4 (i)) which replicates the motion capture convergence as it too oscillates (Fig. 4 (ii)). The remaining physics engines show very little to no oscillation. Also included in the plot is the standard deviation of the motion capture system for comparison. Standard deviations for other plots are not displayed as the discrepancies in simulation are negligible. Finally, it appears that the motion capture is the only one to reach exactly 100 degrees as no other plots reach the same final position.

B. Scene 2: Multi Joint Movement

Scene 2 is a more complex scene where joints two and five are moved multiple times within 20 seconds. Joint 2 is programmed to move between $0 \mid 90 \mid 0 \mid 90 \mid 0$ degrees and joint 5 moves between $0 \mid 90 \mid 0 \mid -90 \mid 0$ degrees.

[1]All code and parameter listings are publicly available from: https://bitbucket.csiro.au/scm/~col549/quantifying-the-reality-gap-in-robotic-manipulation-tasks.git

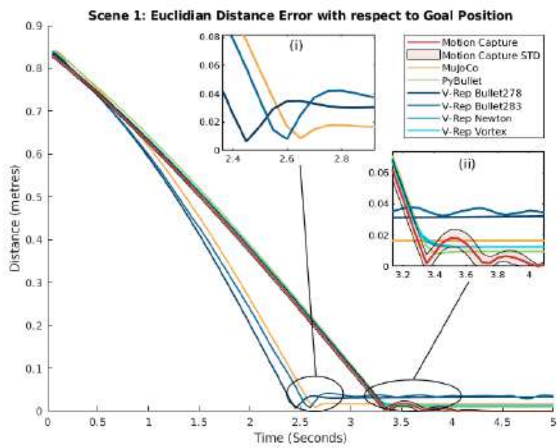


Fig. 4. A single joint motion performed on the Kinova arm both in the real world and in simulators. Plotted is the mean Euclidean distance from the goal position, calculated from 20 runs. (i) and (ii) are areas of note within the plot.

Fig. 5 depicts the euclidean error plot (Eq. 1) where $g_{x,y,z}$ for (a) is the final goal position of the wrist and (b) is the equivalent time step motion capture position. Newton and Vortex follow the motion capture path closely with an accumulated error of $\pm 5.5 - 6$ metres, while PyBullet also has a low accumulated error of ± 7 metres. MuJoCo, Bullet283 and Bullet278 model the motion capture closer between $0-5$ seconds and $10-15$seconds, this is due to the arms moving with gravity towards the goal state and then during the errorfull periods slowly moving aginst gravity and accruing more error. When moving against gravity only MuJoCo is able to reach the final position before changing trajectory. Some of the simulators (i.e. Mujoco and Bullet283) also generate the oscillation seen by the motion capture as the proportional controller attempts to correct the rotation, although none are able to imitate the exact motion of real robot.

C. Scene 3: Interaction with the cube

The most complex scene where joints two, three and four move in a sequence and push a cube along a flat plane within 20 seconds. There are three phases of this scene, the first two position the arm to make contact with the cube and the third initiates the contact and pushes the cube. This scene tests both the control and the physics of the system through the movement of the Kinova arm and the interaction with the cube. Fig. 6 shows three plots: (i) the euclidean distance error (Eq. 1) where $g_{x,y,z}$ is the goal position of the wrist; (ii) the euclidean distance error of the cube (Eq. 1) where $g_{x,y,z}$ is the start position of the cube (i.e. x:0.5, y:0, z:0.0375); and (iii) the rotation of the cube around the y-axis. The first plot shows that no physics engine outperforms any other by a distinguishable margin. This is reinforced by the results in Table I, where the accumulated error for Newton, Vortex and Pybullet are approximately ± 45 metres. It also appears that the motion capture is the closest to reach the goal state, however all plots settle close to the x-axis. The second plot shows the physical interaction between two rigid objects. The greatest displacement is made by MuJoCo followed by the Motion Capture, Bullet283, and then Bullet278. Vortex has very little displacement as the cube makes minimal contact with the Kinova gripper due to the large error seen in the first plot at 15 seconds. Pybullet does not interact with the cube at all, with the gripper moving over the cube. Mujoco is the first to interact with the cube and does so early at about 11.4 seconds whereas all other physics engines begin at about 14 seconds; this is at the conclusion of the previous phase designed to get the gripper in a position to interact with the cube. The final plot shows the pitch of the cube and this is important due to the discrepancies between the physics engines and the ground truth. The plot clearly shows that both Bullet283 and MuJoCo knock the cube in such a way that it rotates 90 degrees. The same movement in reality moves the cube forward with only the smallest amount of discernible rotation. The only physics engine which is able to match the lack of rotation is PyBullet and that is due to it not interacting with the gripper at all. This is particularly relevant given our focus on robotic manipulation, which would benefit greatly from reasonable modelling of these multi-body interactions.

The crossover between physics engines shows PyBullets implementation of Bullet and V-Rep's two implementations of Bullet. The total column from Table I shows a vast difference between the two simulators, with PyBullet drastically better at simulating the chosen scenes with a cumulative position error of ± 45.5 metres while V-Rep's implementations of Bullet both attained ± 131 metres. This could be due to several effects, the first being the default values being set differently (i.e. we set the timestep of PyBullet to be 0.01 seconds while V-Rep generically uses 0.05 seconds), the second could be the underlying implementation between the V-Rep simulator and the physics engine at a scale inaccessible to the user.

For the control of the manipulator Newton, Bullet (PyBullet implementation) and Vortex were considerably and consistently better. For interaction between objects there was no physics engine that modelled the collision well. In reality the cube moved a total of 0.0975 metres in x,y and z with a change in rotation of roll: 0.01, pitch: 0.11 and yaw: 0.89 degrees. The physics engines that were closest to modelling position (Mujoco: 0.1137 metres and Bullet283: 0.0869 metres) had incorrect rotations (MuJoCo and Bullet283 pitch: 90 degrees) and those that had similar real-world rotations had minimal positional movement (< 0.0134 metres).

By stitching together physics engines for discrete periods within a simulation it is believed that a model capable of further reducing the reality gap can be generated. This is backed up by the results which show for the control of the manipulator we should select Newton, PyBullet and Vortex to model the kinematics, without using the results of the remaining physics engines. The control segments could then be combined to the results of MuJoCo, Bullet283 and Bullet278 for the period of interaction with the cube. By populating periods in the simulation timeline with only the

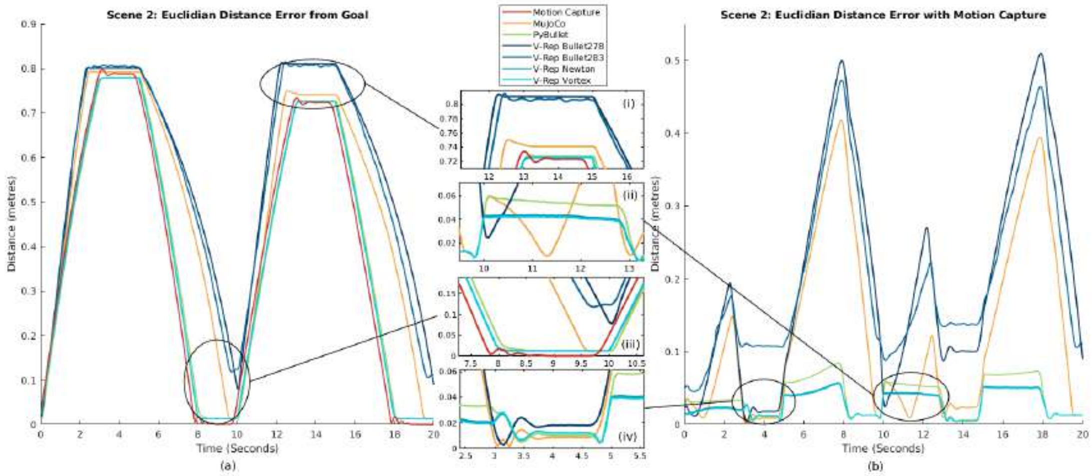


Fig. 5. Plot of Scene 2 with the goal position set to the ground truth. Plotted lines are the euclidean distance from the goal position calculated using the mean of 20 results. (i) and (ii) are areas of note within the plot.

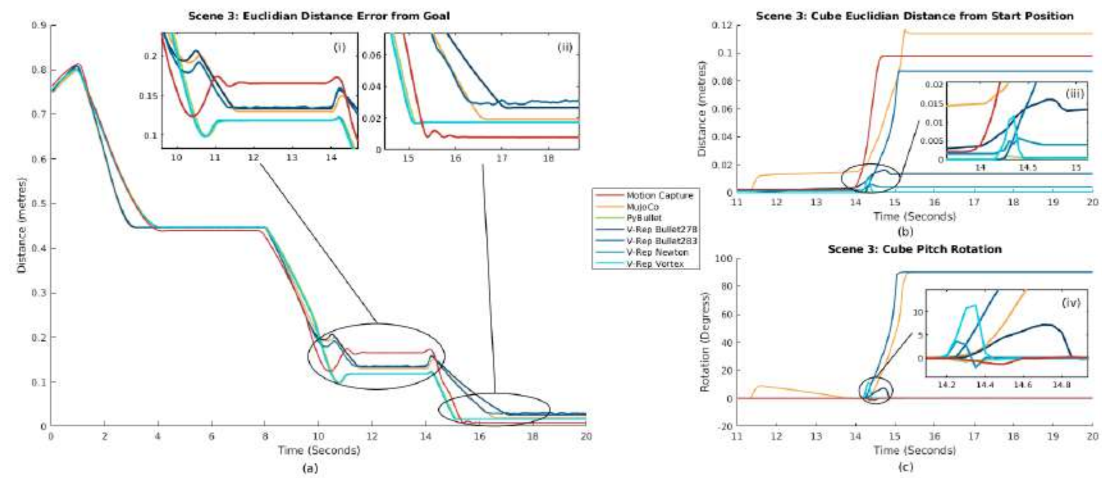


Fig. 6. Three plots of Scene 3: (a) plotted lines represent the wrist's euclidean distance from the final goal position calculated using the mean of 20 results; (b) the x-axis represents the start position of the cube with lines the cubes euclidean distance away; and (c) the pitch of the cube (where pitch is the rotation around the y-axis).(i-iv) are areas of note within the plot.

TABLE I
ACCUMULATED (OVER TIMESTEPS) EUCLIDEAN ERROR (M) COMPARED TO THE GROUND TRUTH

	1 Joint	**2 Joints**	**Cube**	**Total**
MuJoCo	24.237	49.430	23.471	97.138
PyBullet	**18.429**	7.000	**20.084**	45.513
V-Rep (Bullet2.78)	27.412	81.166	25.034	133.611
V-Rep (Bullet2.83)	26.698	80.004	25.215	131.916
V-Rep (Newton)	18.810	**5.579**	21.069	45.458
V-Rep (Vortex)	18.887	5.664	21.130	45.680
V-Rep (ODE)	1.31e+17	1.19e+18	1.88e+18	3.20e+18

optimal performing physics engine the resulting simulation should display a closer realisation of reality.

V. CONCLUSION

We have demonstrated the ability of a range of physics engines to simulate a set of manipulation tasks. Using a motion capture system as the ground truth we record 6DOF pose of a robotic manipulator and directly compare it to simulated data collected from the MuJoCo, PyBullet and V-Rep simulators. The range of tasks test the kinematic and dynamic modelling capabilities of Bullet, Mujoco, Newton, ODE and Vortex. Contributions are both the quantified evidence of the capability of physics engines to model manipulation tasks and the analysis of the simulated and real-world data, including a highly accurate ground truth.

We show the simulation of the kinematic model and control of manipulators is largely solved when compared to the real world, however there are considerable developments necessary for interactions between simulated objects. The physics behind contacts remains a complex problem that is difficult to replicate in simulated environments, and we suggest that a focus on such interactions will bring increasing benefits for the 'learning to grasp' community.

The results highlight the strengths and weaknesses of contemporary simulators with focus on discrepancies between the real-world ground truth. Our contributions will assist researches in the field in their selection of simulators.

References

[1] J. Lehman, J. Clune, D. Misevic, C. Adami, J. Beaulieu, P. J. Bentley, S. Bernard, G. Belson, D. M. Bryson, N. Cheney *et al.*, "The surprising creativity of digital evolution: A collection of anecdotes from the evolutionary computation and artificial life research communities," *arXiv preprint arXiv:1803.03453*, 2018.

[2] N. Sünderhauf, O. Brock, W. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Upcroft, P. Abbeel, W. Burgard, M. Milford, and P. Corke, "The limits and potentials of deep learning for robotics," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 405–420, 2018. [Online]. Available: https://doi.org/10.1177/0278364918770733

[3] F. Zhang, J. Leitner, M. Milford, and P. Corke, "Modular Deep Q Networks for Sim-to-real Transfer of Visuo-motor Policies," Tech. Rep. [Online]. Available: http://cattle3d.com/pap146s1-file1.pdf

[4] N. Jakobi, P. Husbands, and I. Harvey, "Noise and the reality gap: The use of simulation in evolutionary robotics," in *European Conference on Artificial Life*. Springer Berlin Heidelberg, 1995, pp. 704–720.

[5] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[6] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

[7] D. Howard and T. Merz, "A platform for the direct hardware evolution of quadcopter controllers," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-Decem. IEEE, 9 2015, pp. 4614–4619. [Online]. Available: http://ieeexplore.ieee.org/document/7354034/

[8] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 3406–3413.

[9] H. Heijnen, D. Howard, and N. Kottege, "A testbed that evolves hexapod controllers in hardware," in *Proceedings - IEEE International Conference on Robotics and Automation*. IEEE, 5 2017, pp. 1065–1071. [Online]. Available: http://ieeexplore.ieee.org/document/7989128/

[10] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *CoRR*, vol. abs/1603.02199, 2016. [Online]. Available: http://arxiv.org/abs/1603.02199

[11] J. Collins, D. Howard, W. Geles, and F. Maire, "Towards the targeted environment-specific evolution of robot components," in *GECCO 2018 - Proceedings of the 2018 Genetic and Evolutionary Computation Conference*, 2018.

[12] S. James, A. J. Davison, and E. Johns, "Transferring End-to-End Visuomotor Control from Simulation to Real World for a Multi-Stage Task," 7 2017. [Online]. Available: http://arxiv.org/abs/1707.02267

[13] J. Borrego, R. Figueiredo, A. Dehban, P. Moreno, A. Bernardino, and J. Santos-Victor, "A generic visual perception domain randomisation framework for Gazebo," in *18th IEEE International Conference on Autonomous Robot Systems and Competitions, ICARSC 2018*. IEEE, 4 2018, pp. 237–242. [Online]. Available: https://ieeexplore.ieee.org/document/8374189/

[14] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-Septe. IEEE, 9 2017, pp. 23–30. [Online]. Available: http://ieeexplore.ieee.org/document/8202133/

[15] F. Zhang, J. Leitner, M. Milford, B. Upcroft, and P. Corke, "Towards Vision-Based Deep Reinforcement Learning for Robotic Motion Control," 11 2015. [Online]. Available: http://arxiv.org/abs/1511.03791

[16] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, S. Levine, and V. Vanhoucke, "Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping," 2017. [Online]. Available: http://arxiv.org/abs/1709.07857

[17] J. C. Zagal, J. Ruiz-del Solar, and P. Vallejos, "Back to reality: Crossing the reality gap in evolutionary robotics," *IFAC Proceedings Volumes*, vol. 37, no. 8, pp. 834–839, 7 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667017320840

[18] T. Laue and M. Hebbel, "Automatic Parameter Optimization for a Dynamic Robot Simulation," in *RoboCup 2008: Robot Soccer World Cup XII*, L. Iocchi, H. Matsubara, A. Weitzenfeld, and C. Zhou, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 121–132.

[19] S. Koos, J.-B. Mouret, and S. Doncieux, "Crossing the reality gap in evolutionary robotics by promoting transferable controllers," in *Proceedings of the 12th annual conference on Genetic and evolutionary computation - GECCO '10*, 2010, p. 119. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1830483.1830505

[20] J.-B. Mouret and K. Chatzilygeroudis, "20 Years of Reality Gap," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion on - GECCO '17*, 2017, pp. 1121–1124. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3067695.3082052

[21] A. Boeing and T. Bräunl, "Leveraging multiple simulators for crossing the reality gap," in *2012 12th International Conference on Control, Automation, Robotics and Vision, ICARCV 2012*, 2012, pp. 1113–1119.

[22] M. Eaton, "Bridging the reality gap - A dual simulator approach to the evolution of whole-body motion for the nao humanoid robot," in *IJCCI 2016 - Proceedings of the 8th International Joint Conference on Computational Intelligence*, vol. 1, 2016, pp. 186–192.

[23] A. Boeing and T. Bräunl, "Evaluation of real-time physics simulation systems," in *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia - GRAPHITE '07*. New York, New York, USA: ACM Press, 2007, p. 281. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1321261.1321312

[24] S. J. Chung and N. Pollard, "Predictable behavior during contact simulation: A comparison of selected physics engines," *Computer Animation and Virtual Worlds*, vol. 27, no. 3-4, pp. 262–270, 2016.

[25] G. Gonzalez-Badillo, H. I. Medellin-Castillo, T. Lim, J. M. Ritchie, R. C. Sung, and S. Garbaya, "A new methodology to evaluate the performance of physics simulation engines in haptic virtual assembly," *Assembly Automation*, vol. 34, no. 2, pp. 128–140, 2014.

[26] J. Fabry and S. Sinclair, "Interactive visualizations for testing physics engines in robotics," in *Proceedings - 2016 IEEE Working Conference on Software Visualization, VISSOFT 2016*, 2016, pp. 106–110.

[27] T. Erez, Y. Tassa, and E. Todorov, "Simulation tools for model-based robotics: Comparison of Bullet, Havok, MuJoCo, ODE and PhysX," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 5 2015, pp. 4397–4404. [Online]. Available: http://ieeexplore.ieee.org/document/7139807/

[28] E. Rohmer, S. P. Singh, and M. Freese, "V-REP: A versatile and scalable robot simulation framework," in *IEEE International Conference on Intelligent Robots and Systems*. IEEE, 11 2013, pp. 1321–1326. [Online]. Available: http://ieeexplore.ieee.org/document/6696520/

[29] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *IEEE International Conference on Intelligent Robots and Systems*. IEEE, 10 2012, pp. 5026–5033. [Online]. Available: http://ieeexplore.ieee.org/document/6386109/

[30] E. Coumans and Y. Bai, "Pybullet, a Python Module for Physics Simulation for Games, Robotics and Machine Learning," 2016. [Online]. Available: https://pybullet.org

Evolutionary Humanoid Robotics: Past, Present and Future

Malachy Eaton

Department of Computer Science and Information Systems
University of Limerick, Ireland
malachy.eaton@ul.ie

Abstract. Evolutionary robotics is a methodology for the creation of autonomous robots using evolutionary principles. Humanoid robotics is concerned specifically with autonomous robots that are human-like in that they mimic the body or aspects of the sensory, processing and/or motor functions of humans to a greater or lesser degree. We investigate how these twin strands of advanced research in the field of autonomous mobile robotics have progressed over the last decade or so, and their current recent convergence in the new field of evolutionary humanoid robotics. We describe our current work in the evolution of controllers for bipedal locomotion in a simulated humanoid robot using an accurate physics simulator, and briefly discuss the effects of changes in robot mobility and of environmental changes. We then describe our current work in the implementation of these simulated robots using the Bioloid robot platform. We conclude with a look at possible visions for the future.

Keywords: Artificial evolution, humanoid robotics.

1 Introduction and Motivation

Evolutionary humanoid robotics is a branch of evolutionary robotics dealing with the application of the laws of genetics and the principle of natural selection to the design of humanoid robots. For a good introduction to the general field see the book by Nolfi and Floreano[1]. Evolutionary techniques have been applied to the design of both robot body and 'brain' for a variety of different wheeled and legged robots[2-6]. In this article we are primarily concerned with the application of evolutionary techniques to autonomous robots whose morphology and/or control/sensory apparatus is broadly human-like.

In Brooks' paper on the subject [7] he lists two main motivations for the construction (or evolution) of humanoid robots. He presents the argument that the form of human bodies may well be critical to the representations we use for both language and thought. Thus, if we wish (for whatever reason) to build a human-like intelligence the robot body must also be human-like. This is a view supported by Pfeifer and Bongard in their recent book, aptly titled "How the body shapes the way we think". [8]

The second motivation he suggests relates to the area of human-robot interaction. If the robot has a human-like form then people should find it easier and more natural

M. Lungarella et al. (Eds.): 50 Years of AI, Festschrift, LNAI 4850, pp. 42–52, 2007.

to interact with it just as if it were human. However it is important not to ignore the so-called "uncanny valley" effect as presented by Mashahiro Mori [9] and further discussed by Mac Dorman [10]. This suggests that there is a positive correlation between the likeness of a robot to a human; with how familiar and hence how easy they are to interact with, from a human perspective. This is as we would expect. However after a certain point small further increases produce a sharp decrease in familiarity (the *"uncanny valley"*) which only then increases again as the robot becomes almost indistinguishable from a human. This effect is seen to increase for a moving robot as opposed to a stationary one. It is thought that this unnerving effect is correlated to an innate fear of mortality and culturally evolved mechanisms for coping with this. This may suggest the desirability of, for the present, striving to produce humanoid robots with useful skills (discussed further below), but without at this stage attempting to imbue them with over human-like features or expressions.

A third possible motivation, not discussed in Brooks' paper is that the humanoid robot may be able to operate with ease in environments and situations where humans operate, such as opening and closing door, climbing up and down stairs, bending down to put washing in a washing machine etc. This will allow the robot to be useful in a whole host of situations in which a non-humanoid robot would be quite powerless. A major application in the future could be in the area of home helps for the elderly. In modern developed countries like Italy and Japan fertility rates are dropping dramatically and they are left with a disproportionate elderly population and a relatively small population of potential carers.

The hope is that by using artificial evolution robots may be evolved which are stable and robust, and which would be difficult to design by conventional techniques alone. However we should bear in mind the caveat put forward by Mataric and Cliff [11]; that the effort expended in designing and configuring the evolutionary algorithm should ideally be considerably less than that required to do a manual design.

For a brief introduction to the current state of the art with regard to Humanoid robotics including the HRP-3, KHR-1 and KHR-2, Sony QRIO and Honda ASIMO and P2 see Akachi et al [12]. See references [13-20] for other articles of specific interest in the design of autonomous robots, and humanoid robots in particular. The increasingly important issue of the benchmarking and evaluation of future autonomous robots, which is an area that will be of increasing relevance and significance as intelligent robots, especially of the humanoid variety, play a greater role in our everyday lives is discussed in references [21] and [22].

For other work in the specific area of the evolution of bipedal locomotion see references [23-30]. Space precludes a detailed discussion of this other work, some of it very interesting, however in these papers walking generally operates either solely on a simulated robot[24-26,28-30], and/or on robots with a restricted number of degrees of freedom (typically 6-10 DOF). Some of these systems also require the incorporation of quite a high degree of domain specific knowledge in the genetic algorithm. In addition there is some very interesting work on bipedal locomotion without the use of complex control algorithms in Cornell University and Delft University among others (passive-dynamic walkers)[31], which have five DOF each, and in the MIT learning biped which has six DOF and uses reinforcement learning to acquire a control policy [31], but as these are not evolutionary based techniques they fall outside the scope of this discussion.

2 Evolution of Bipedal Locomotion

We now concisely describe a specific set of experiments for the evolution of bipedal locomotion in a high-DOF simulated humanoid robot. Bipedal locomotion is a difficult task, which, in the past, was thought to separate us from the higher primates. In the experiments outlined here we use a genetic algorithm to choose the joint values for a simulated humanoid robot with a total of 20 degrees of freedom (elbows, ankles, knees, etc.) for specific time intervals (keyframes) together with maximum joint ranges in order to evolve bipedal locomotion. An existing interpolation function fills in the values between keyframes; once a cycle of 4 keyframes is completed it repeats until the end of the run, or until the robot falls over. The humanoid robot is simulated using the Webots mobile robot simulation package and is broadly modeled on the Sony QRIO humanoid robot [32,33]. In order to get the robot to walk a simple function based on the product of the length of time the robot remains standing by the total distance traveled by the robot was devised. This was later modified to reward walking in a forward (rather than backward) direction and to promote walking in a more upright position, by taking the robots final height into account.

In previous experiments [34] it was found that the range of movement allowed to the joints by the evolutionary algorithm: that is the proportion of the maximum range of movement allowed to the robot for each joint, was an important factor in evolving successful walks. Initial experiments placed no restriction on the range of movement allowed and walks did not evolve unless the robot was restricted to a stooped posture and a symmetrical gait, even then results were not impressive. By restricting possible movement to different fractions of the maximum range walks did evolve, however as this was seen as a critical factor in the evolutionary process it was decided in the current work to include a value specifying the fraction of the total range allowed in the humanoid robots genome.

The genome length is 328 bits comprising 4 bits determining the position of the 20 motors for each of 4 keyframes; 80 strings are used per generation. 8 bits define the fraction of the maximum movement range allowed. The maximum range allowed for a particular genome is the value specified in the corresponding to each motor divided by the number of bits set in this 8 bit field plus 1. 8 bits was chosen as reasonable walking patterns were seen to evolve when the range was restricted by a factor of 4 or thereabouts in previous experiments. The genetic algorithm uses roulette wheel selection with elitism; the top string being guaranteed safe passage to the next generation, together with standard crossover and mutation. Maximum fitness values may rise as well as fall because of the realistic nature of the Webots simulation. Two-point crossover is applied with a probability of 0.5 and the probability of a bit being mutated is 0.04. These values were arrived at after some experimentation.

3 Experimental Results

We ran three trials of the evolutionary algorithm on a population size of 80 controllers for 700 generations of simulated robots, taking approximately 2.5 weeks simulation time on a reasonably fast computer, corresponding to approximately 7 weeks of "real time" experimentation. A fitness value over about 100 corresponds to the robot at least staying standing for some period of time, over 500 corresponds to a walk of some description. The results obtained were interesting; walks developed in all three runs, on average after about 30 generations with fine walking gaits after about 300 generations. This is about half the time on average that walking developed with a fixed joint range. We can see from Fig. 1 that the joint range associated with the individual with maximum fitness fluctuates wildly in early generations; typically low values (high movement ranges) initially predominate as the robot moves in a "thrashing" fashion. Then the movement range becomes restricted for the highest performing individuals as a smaller range of movement increases the likelihood that the robot will at least remain standing for a period, while hopefully moving a little. Then in later generations typically the movement range gradually becomes relaxed again, as a greater range of movement facilitates more rapid walking once the robot has "learnt" how to remain upright.

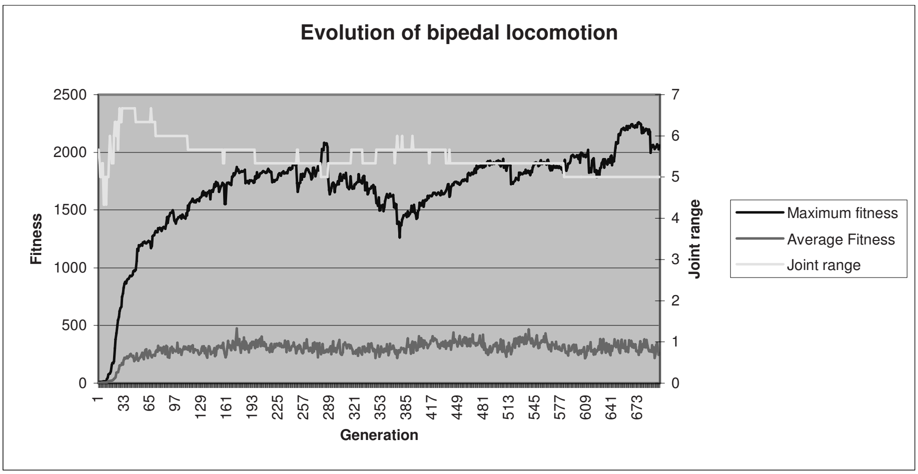


Fig. 1. Fitness and joint range graphs

4 Effect of Restraining Joints and of Environment Modification

We now investigate the effect of restraining motion in part of the robot. We do this by immobilising the robots right knee joint, and both ankle joints. This might correspond to a situation of a person walking with a prosthetic leg. Figure 2 shows the results of this experiment again averaged over 3 runs. The robot learns to walk albeit with a reduced maximum fitness compared to the robot with no constraints. Figure 3 illustrates a typical walk which develops. The right (constrained) leg moves sideways and forwards, coming well off the ground, as the right arm moves backwards in a steadying motion. The left leg follows in a shuffling motion, and the cycle repeats. In order to gain some insight into the evolutionary process we use a slightly modified version of the "*degree of population diversity*" described in by Leung et al. [35]. This measure provides an easy to calculate and useful measure of population diversity: i.e. how alike the different strings in a population. We subtract this value from the genome bit length to produce our inverse degree of population diversity measure (IDOPD). This value will vary from 0 (no similarity in the strings) to a value corresponding to the genome length (all genomes have the same value at every bit location). Diversity is measured on the right hand vertical axis. In addition for these experiments the number of bits encoding the joint range was increased from 8 to 16 giving a total genome length of 336 bits. We have also conducted successful experiments on the robot walking (skating) in low-friction environments, and in conditions of reduced gravity.

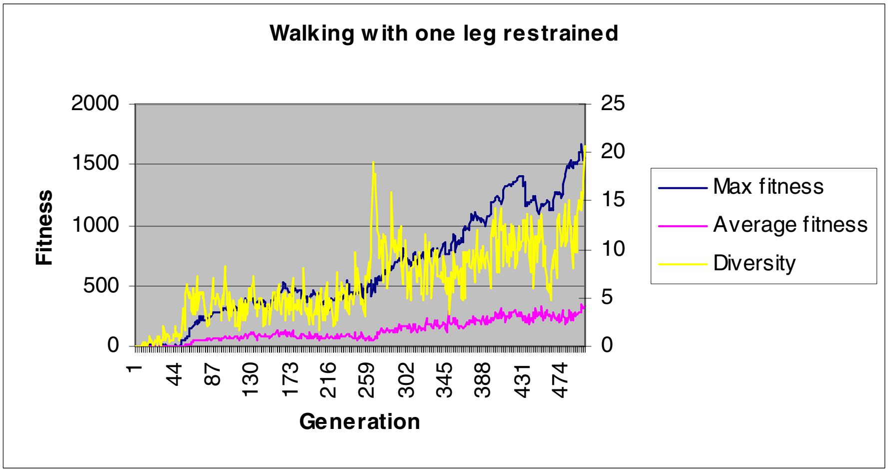


Fig. 2. Walking with right leg immobilised

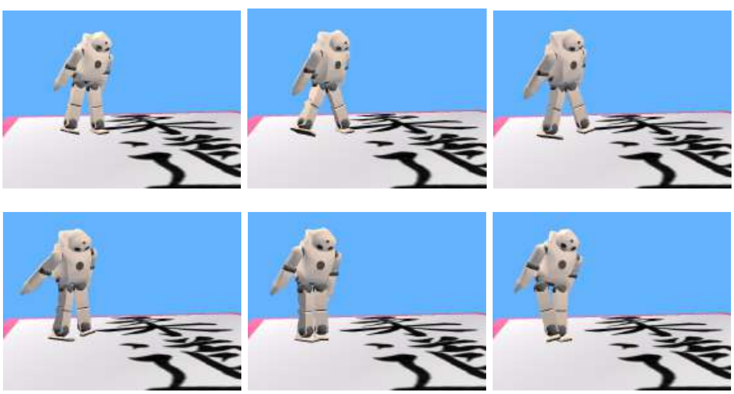


Fig. 3. The evolved sideways walk

5 The Bioloid Robotic Platform

We have been working for some time to implement our simulated robots in the real world using the Bioloid robot platform which is produced by Robotis Inc. Korea. This platform consists of a CPU (the CM-5), a number of senso-motoric actuators (Dynamixel AX12+) and a large number of universal frame construction pieces.

Using this platform it is possible to construct a wide variety of robots, from simple wheeled robots to complex humanoid robots with many degrees of freedom. In addition, because of the ability to construct a range of robots with slightly different morphologies, it lends itself well to evolutionary robotics experiments in which both robot "body and brain" are evolved. To gain initial experience with this kit we initially constructed a "puppy-bot" (Fig. 4) which can walk on four legs, avoid obstacles and perform several cute tricks. With this experience we then constructed the Bioloid humanoid robot which has 18 degrees of freedom in total. A modified version of this humanoid robot was used for Humanoid Team Humboldt in the RoboCup competitions in Bremen 2006. [36]

Two pieces of software are provided with the Bioloid system; the behaviour control programmer, and the motion editor. The behaviour control programmer programs the humanoids' response to different environmental stimuli, while the motion editor describes individual motions based on the keyframe concept described in our work.

Fig. 4. The puppy robot (left) and Bioloid humanoid robot (right)

6 Implementation of Simulated Robots

We have now built an accurate model of the Bioloid humanoid in Webots, and can translate the information in our sequence control file into a format understandable by the Bioloid motion editor. This translation is currently done partly by hand but we are working on fully automating this process. It is now possible to evolve walking, and other behaviours, in Webots using our model, and then transfer the evolved behaviour directly to the Bioloid humanoid robot. The graph below shows the maximum evolved fitness (right hand axis) and joint range and diversity measure (left hand axis) for the simulated Bioloid humanoid, averaged over five runs for 200 generations. Good walking patterns generally developed by generation 100 or so. Note the fitness function is similar to the one used for the PINO-like robot, but as the robot is a lot shorter (approx 35cm), obviously the distance traveled in a successful walk is less, hence the fitness values returned will be lower. Also the joint range values vary as in

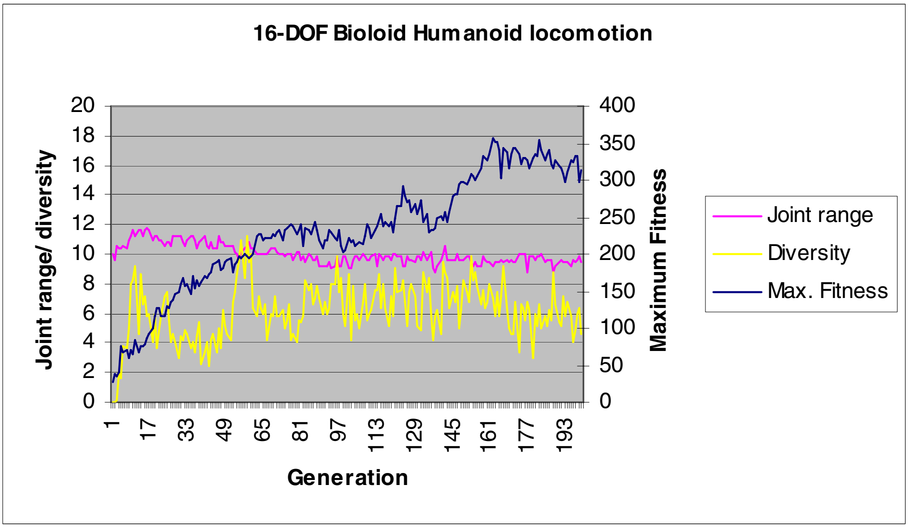


Fig. 5. Maximum fitness, Joint range and diversity graphs for 200 generations of the Webots simulation of the 16-DOF Bioloid Humanoid robot averaged over 5 runs

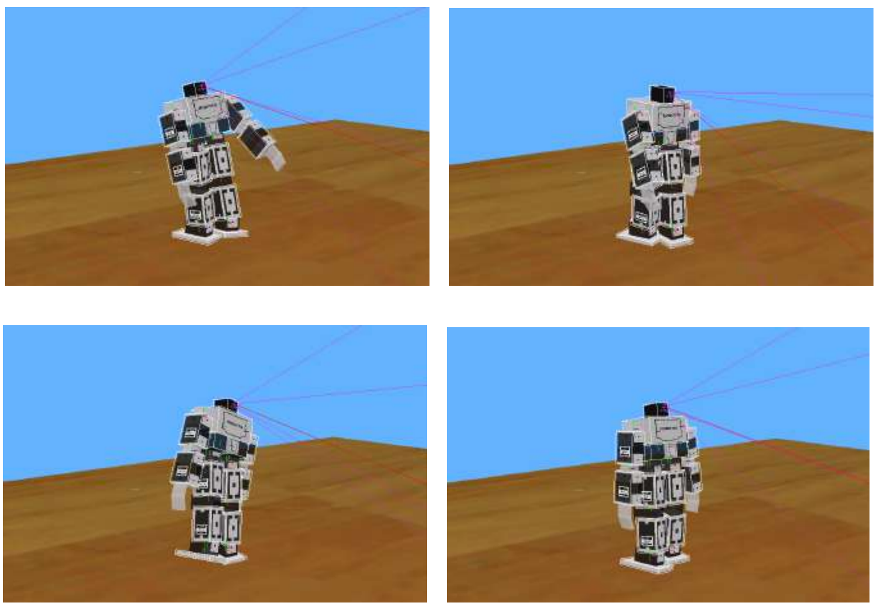


Fig. 6. Simulated snap shots of the Webots simulation of the Bioloid humanoid robot walking

the previous experiment, rising in the early runs before stabilizing out; The difference however is not as marked, as the robot is more inherently stable to begin with, so does not have to restrict its joint ranges as much in the early runs so as to remain standing.

The robot simulated was 16-DOF rather than the 18-DOF of the actual robot; both elbow joints were immobilized. This was done as they operate in an in-out fashion rather than the forward-backward configuration in the original PINO-like robot and this also might interfere with the arm movement. Also four additional 16-bit fields were added to specify the speed of movement for each of the four cycles, however this is not of particular relevance to the current discussion and will be discussed in more detail in a a later article.

When a walk has evolved this can then be transferred to the Bioloid humanoid via the motion editor. An example of an evolved simulated walk is given in Fig. 6 together with the walk as transferred to the Bioloid humanoid in Fig. 7. The fidelity of the transfer is reasonably good, indicating the accuracy of the model, however work remains to be done to fully "cross the reality gap"[37], but these initial results are very promising.

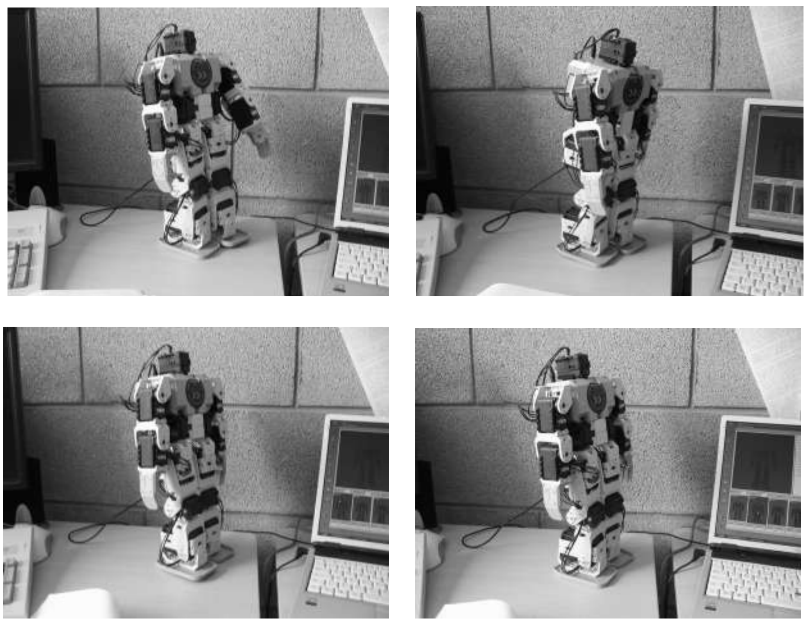


Fig. 7. The actual walking control sequence transferred to the physical robot

7 Discussion and Looking Ahead

In this work we have demonstrated one of the first applications of evolutionary algorithms to the development of complex movement patterns in a many-degree-of-freedom humanoid robot. Perhaps the time has now arrived for a more serious and detailed discussion on the possible ethical ramifications of the evolution of human-like robots. Such robots may be able to take our place in the workforce or in other fields and there may well also be other significant social consequences. Other, more technical, issues arise – while Asimov's three laws of robotics may appear a little dated, it could be important to avoid the appearance in the home or workplace of unexpected evolved “side effects” which may escape a rigorous testing regime. For example one of the walking behaviours evolved in our work involved the robot walking (staggering) in a manner amusingly reminiscent of an intoxicated person. While this gait proved surprisingly effective, not many people would relish the prospect of being shown around a new house for sale by a seemingly drunken robot! In conclusion, if indeed we are now beginning to see the first tentative “steps” towards autonomous humanoid robots, perhaps now is the time to look forward to the harnessing of this technology for the benefit of mankind.

Acknowledgments. Thanks are due to T.J. Davitt for his work on the initial version of the code for this system, and to Diarmuid Scannell for his work on the Webots Bioloid humanoid model. Also much appreciation to Olivier Michel of Cyberbotics for his help and rapid responses to my many queries.

References

1. Nolfi, S., Floreano, D.: Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines. MIT Press, Cambridge (2000)
2. Holland, O.: From the imitation of life to machine conciousness. In: Kunii, H.S., Jajodia, S., Sølvberg, A. (eds.) ER 2001. LNCS, vol. 2224, pp. 1–37. Springer, Heidelberg (2001)
3. Floreano, D., Urzelai, J.: Evolutionary Robotics: The Next Generation. In: Gomi, T. (ed.) Evolutionary Robotics III, AAI Books, Ontario (Canada) (2000)
4. Brooks, R.: Steps towards living machines. In: Kunii, H.S., Jajodia, S., Sølvberg, A. (eds.) ER 2001. LNCS, vol. 2224, pp. 72–93. Springer, Heidelberg (2001)
5. Harvey, I.: Artificial evolution: a continuing SAGA. In: Kunii, H.S., Jajodia, S., Sølvberg, A. (eds.) ER 2001. LNCS, vol. 2224, pp. 94–109. Springer, Heidelberg (2001)
6. Pollack, J., Lipson, H., Funes, P., et al.: First three generations of evolved robots. In: Kunii, H.S., Jajodia, S., Sølvberg, A. (eds.) ER 2001. LNCS, vol. 2224, pp. 62–71. Springer, Heidelberg (2001)
7. Brooks, R.: Evolutionary robotics: where from and where to. In: Embley, D.W. (ed.) ER 1997. LNCS, vol. 1331, pp. 1–13. Springer, Heidelberg (1997)
8. Pfeifer, R., Bongard, J.: How the body shapes the way we think: a new view of intelligence. MIT press, Cambridge (2007)
9. Mori, M.: Bukimi no tani [the uncanny valley]. Energy 7, 33–35 (1970)
10. MacDorman, K.F.: Mortality salience and the uncanny valley. In: Proceedings of 2005 IEEE-RAS International Conference on Humanoid Robots, IEEE Computer Society Press, Los Alamitos (2005)
11. Mataric, M., Cliff, D.: Challenges in evolving controllers for physical robots. Robotics and autonomous systems 19(1), 67–83 (1996)
12. Akachi, K., Kaneko, K., Kanehira, N., et al.: Development of humanoid robot HRP-3P. In: Proc. of 2005 5th IEEE-RAS International Conference on humanoid robots, pp. 50–55. IEEE Computer Society Press, Los Alamitos (2005)
13. Xie, M., Weng, J., Fontanie, J.: Editorial. International Journal of Humanoid Robotics 1(1) (2004)
14. Brooks, R., Braezel, C., Marjanovic, M., et al.: The Cog project: building a humanoid robot. In: Nehaniv, C.L. (ed.) Computation for Metaphors, Analogy, and Agents. LNCS (LNAI), vol. 1562, Springer, Heidelberg (1999)

15. Pfeifer, R., Scheier, C.: Implications of embodiment for robot learning. In: Proc. 2nd European workshop on advanced mobile robots, pp. 38–43 (October 1997)
16. Brooks, R., Aryananda, L., Edsinger, A., et al.: Sensing and manipulating built-for-human environments. International Journal of Humanoid Robotics 1(1), 1–28 (2004)
17. Full, R.: Using biological inspiration to build artificial life that locomotes. In: Kunii, H.S., Jajodia, S., Sølvberg, A. (eds.) ER 2001. LNCS, vol. 2224, pp. 110–120. Springer, Heidelberg (2001)
18. Ambrose, R., Ambrose, C.: Primate anatomy, kinematics, and principles for humanoid design. International Journal of Humanoid Robotics 1(1), 175–198 (2004)
19. Zykov, V., Bongard, J., Lipson, H.: Evolving Dynamic Gaits on a Physical Robot. In: Deb, K. (ed.) GECCO 2004. LNCS, vol. 3102, Springer, Heidelberg (2004)
20. Bongard, J., Zykov, V., Lipson, H.: Resilient machines through continuous self modeling. Science 314, 1118–1121 (2006)
21. Gat, E.: Towards a principled experimental study of autonomous mobile robots. Autonomous Robots 2, 179–189 (1995)
22. Eaton, M., Collins, J.J., Sheehan, L.: Towards a benchmarking framework for research into bio-inspired hardware-software artefacts. Artificial Life and Robotics 5(1), 40–46 (2001)
23. Boeing, A., Hanham, S., Braunl, T.: Evolving autonomous biped control from simulation to reality. In: Proc. 2nd International Conference on Autonomous Robots and Agents, December 13-15, pp. 440–445 (2004)
24. Bongard, J., Paul, C.: Making evolution an offer it can't refuse: morphology and the extradimensional bypass. In: Kelemen, J., Sosík, P. (eds.) ECAL 2001. LNCS (LNAI), vol. 2159, pp. 401–412. Springer, Heidelberg (2001)
25. Ishiguro, A., Fujii, A., Eggenburger, H.: Neuromodulated control of bipedal locomotion using a polymorphic CPG circuit. Adaptive Behavior 11(1), 7–18 (2003)
26. Endo, K., Maeno, T., Kitano, H.: Co-evolution of morphology and walking pattern of biped humanoid robot using evolutionary computation-consideration of characteristic of the servomotors. In: Proceedings of the 2002 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems, Lausanne, Switzerland, pp. 2678–2683 (October 2002)
27. Zhang, R., Vadakkepat: An evolutionary algorithm for trajectory based gait generation of biped robot. In: Proceedings of the International Conference on Computational Intelligence, Robotics and Autonomous Systems, Singapore (2003)
28. Reil, T., Husbands, P.: Evolution of central pattern generators for bipedal walking in a real-time physics environment. IEEE Trans. On Evolutionary Computation 6(2), 159–168 (2002)
29. Sellers, W.I., Dennis, L.A., Crompton, R.H.: Predicting the metabolic energy costs of bipedalism using evolutionary robotics. The Journal of Experimental Biology 206, 1127–1136 (2003)
30. Miyashita, K., Ok, S., Kazunori, H.: Evolutionary generation of human-like bipedal locomotion. Mechatronics 13, 791–807 (2003)
31. Collins, S., Ruina, A., Tedrake, R., et al.: Efficient bipedal robots based on passive-dynamic walkers. Science 307, 1082–1085 (2005)
32. Michel, O.: Webots: Professional Mobile Robot Simulation. International Journal of Advanced Robotic Systems 1(1), 39–42
33. Mojon, S.: Realization of a Physic Simulation for a Biped Robot. In: Semester project, School of Computer and Communication Sciences EPFL, Switzerland (2003)
34. Eaton, M., Davitt, T.J.: Automatic evolution of bipedal locomotion in a simulated humanoid robot with many degrees of freedom. In: Proceedings of the 11th International Symposium on Artificial Life and Robotics, Japan, January 23-25, 2006, pp. 448–451 (2006)

35. Leung, Y., Gao, Y., Zong-Ben, X.: Degree of population diversity – a perspective on premature convergence in genetic algorithms and its Markov chain analysis. IEEE Transactions on Neural Networks 8(5), 1165–1176 (1997)
36. Hild, M., Jungel, M., Spranger, M.: Humanoid team Humboldt team description 2006. In: Proceedings CD Robocup 2006, Bremen, Germany (2006)
37. Lipson, H., Bongard, J., Zykov, V., Malone, E.: Evolutionary robotics for legged machines: from simulation to physical reality. In: Arai, T., et al. (eds.) Intelligent Autonomous Systems 9(IAS-9), pp. 11–18 (2006)

An approach to the synthesis of humanoid robot dance using non-interactive evolutionary techniques.

Malachy Eaton, *Member, IEEE*
Dept. Computer Science and Information Systems,
University of Limerick,
Limerick, Ireland
malachy.eaton@ul.ie

***Abstract*—After bipedal locomotion, dance is one of the most commonly studied behaviours for researchers seeking to replicate human-like motion in humanoid robots. Many of the methods employed involve direct interaction with, or imitation of, human participant(s). For example, the generation of dance movements using interactive evolutionary computation (IEC) involves the replacement of an objective fitness function with the subjective evaluations of human observer(s). In this paper we present an alternative approach to the synthesis of humanoid robot dance using non-interactive evolutionary computation (non-IEC) methods. We propose a novel fitness function for the evolution of robotic dance, and we present initial results of the application of this evolutionary process to the generation of dance patterns for the 18-DOF Bioloid humanoid robot. We conclude that even without the presence of a human or humans in the evolutionary loop, it is possible to produce surprisingly lifelike and novel dances using this approach.**

Keywords- genetic algorithms; humanoid robots; evolutionary robotics; dance behaviours

I. Introduction

Evolutionary humanoid robotics (EHR) is a subset of the evolutionary robotics field which is broadly concerned with the application of evolutionary techniques to the creation of novel behaviours and/or morphologies for humanoid robots [1]. After walking, dance is probably one of the most widely studied forms of robotic movement by researchers seeking to evolve human-like movement in humanoid robots. Many of the papers on this topic utilise a form of interactive or interleaved evolution; that is an evolutionary process in which a human, or humans, play a part in the fitness function once the evolutionary process has started. Of course there is always some degree of *a priori*, or domain specific information included in the formulation of individual fitness functions [2], however interactive evolution takes this one step further by including a human, or humans interactively in the evolutionary cycle [3]. This will, of course, introduce an extra element of subjectivity to the final results obtained, with these results tailored, to a substantial extent, to the individual preferences of the individual, or of the groups of individuals involved in the evaluation process.

This paper is structured as follows. In Section II we briefly discuss some of the commonly employed methods currently used for the generation of robot dance behaviours, including dance by imitation, and the generation of dance using a process of interactive evolution, that is by replacing the objective fitness function commonly used by genetic algorithms with a subjective fitness function based, at least in part, by the dance preferences of a human observer(s). Then in Section III we present a brief discussion of the software simulation platform used to evolve the robotic dances, and the hardware platform (humanoid robot) used to implement these dances. In Section IV we discuss the fitness function employed and the rationale involved in its derivation.

Experimental results are then presented in Section V, with screenshots of the Webots [4] simulator used to evolve dances, together with photographs of keyframes from a dance transferred to the actual robot. We then finish with a brief discussion of the experiments conducted, together with our general conclusions.

II. Different Approaches to Humanoid Robot Dance Generation

There are several different approaches that may be used in the generation of dance movements in humanoid robots. One method that may be employed uses the keyframe concept, where the position of the robots joints at fixed intervals of time are set by the user, and then these keyframes are joined together using some smoothing algorithm to produce fluidic motion. Another possible approach is the imitation learning, or learning from observation approach, where the robot observes a human dance and then, through some form of feature extraction of the dance motions, transfers these motions to actuate the joints of the humanoid robot. Issues may, of course, arise due to differences between human and robot morphologies. An interesting application of this method was used by researchers employing motion capture data in order to generate classic Japanese folk dances[5].

Another approach to the automatic synthesis of humanoid robot dance is by a process of interactive evolutionary computation (IEC). Here the user is presented with a set of alternative individuals and evaluates them based on subjective criteria. The results of this selection are fed back into the evolutionary algorithm (EA) and the evolution progresses on. In essence the objective fitness function used by the "standard" evolutionary algorithm is replaced by the subjective preferences of the user. The use of interactive evolutionary algorithms is not new, and goes back to Richard Dawkins'

978-1-4799-0652-9/13 $31.00 © 2013 IEEE
DOI 10.1109/SMC.2013.563

Biomorphs [6] and Karl Sims' pioneering work in the evolution of virtual creatures and of generating art through EA's[7,8]. For recent work in the area of the application of IEC to dance see [9,10]. For a review of the general field of dancing robots throughout history see [11].

All of the methods outlined above have the characteristic that they require the presence of a human or humans "in the loop", and this is quite a common practice for the generating complex motions in humanoids. For example in [12], although the final optimised motions for sitting and for kicking behaviours are generated by a real valued GA, the initial motions were generated using IEC. This may not be seen necessarily as a negative factor, as, in a sense all dance may be seen in a subjective light. Our approach, however, is to try to reduce this subjective element to a minimum and to attempt to evolve dance-like behaviours using little or no human intervention. The design of a suitable fitness function is, of course, key to this process and we will discuss this shortly.

To conclude this section we should not trivialise the role that dance has played throughout human history and civilization. To quote from Hanna[13], and as reproduced in Or [11]

" To dance is human, and humanity almost universally expresses itself in dance. Dance interweaves with other aspects of human life, such as communication and learning, belief systems, social relations and political dynamics, loving and fighting, and urbanization and change. It may even have been significant in the biological and evolutionary development of the human species. When dance is suppressed for moral, religious, or political reasons, it rises phoenixlike to assert the essence of humanity. "

III. SOFTWARE SIMULATION AND HARDWARE IMPLEMENTATION

Our simulation of the robot dance movements also uses the keyframe concept described in the previous section, however rather than manually setting the joint positions for each keyframe, these value are chosen by the genetic algorithm. The simulation is performed using the Webots mobile robot simulation package which employs an accurate physics simulator, thus facilitating the transfer for simulation to the real robot [4]. This package allows for the creation of robot world, robot controllers, and the actual robots themselves. The robot modeled is the Bioloid humanoid, which has 18 degrees of freedom in total. This robot is built from a kit produced by Robotis Inc. Korea. This kit comprises the CM510 processor, together with 18 AX12+ senso-motoric actuators, and an assortment of frame pieces from which the humanoid robot can be built. Each of these 18 degrees of freedom is represented by an individual value in the chromosome, and a total of four keyframes determine each cycle of the dance movement. This cycle is repeated until the end of the run, or until the robot falls over.

The genetic algorithm employs standard roulette wheel selection, together with elitism, the top string from each generation passes automatically to the next. Two point crossover with a probability of 0.5 together with a mutation rate of 0.04 is used. These values were arrived at after some experimentation. In addition four seperate values define the speed of movement for each of the four keyframes. In summary then, the composition of the chromosome is as follows: the 18 degrees of freedom of the robot, represented by joint angles are encoded as a 16 bit value each for each of the 4 keyframes, and an additional four 16-bit fields determine the speed of movement for each of the 4 keyframes. A further 16 bit value represents the fraction of the possible joint range the robot can use for each of its joints. Although this value proved very useful in previous experiments involving the evolution of stable bipedal locomotion [14] it was left unused in the experiments described here, as we wish to encourage joint movements over a wide range. A further two 16 bit fields (which could represent an additional two degrees of freedom) are also left unused in these experiments. This gives a total chromosome length of 400 bits. Further details of the experimental setup are given in [1,14].

Once a satisfactory dance has been evolved this is transferred to the actual Bioloid humanoid robot using a semi-automated process developed as part of this research, which maps the values generated in the Webots simulator directly onto joint angles for the Bioloid humanoid. The evolved dance may then be evaluated on the real robot.

IV. EVOLUTION OF A FITNESS FUNCTION FOR DANCE

Our main rationale behind the using the evolutionary process in the generation of dance motions is to aim for a fluidic rhythmic motion which is pleasing to the eye. No motion, or very little motion would generally not rate very highly, as would random joint movements. In addition, dances involving the robot dancer falling over midway through the dance would not (generally) meet with high approval ratings. In order to address the first issue (avoiding dances with little or no movement involved) we employ a fitness function designed specifically to reward dance movements which maximise movements on the dance floor. This was initially generated by summing the total movements over a particular dance sequence for each individual joint. The issue of very random, jerking, dances is addressed in two ways. Firstly, because of the cyclical nature of the movements generated by the genetic algorithm, a level of order is inherent in the robots' movements. Secondly, very jerky movements may well result in the robot falling over, which results in a sequence being terminated prematurely and the robot being subsequently being assigned a relatively low fitness value.

In keeping with these ideas, the initial fitness function for dance involved the sum of all of the movement values (absolute value of angular rotation values) over all of the joints multiplied by the time the robot remained standing. It was felt, however, that this function did not provide sufficient penalty for a robot which, for example, decided not to move its knees, thus generating a stable, but rather stilted and not very fluid or dynamic motion. An attempt was made to counteract this effect by providing weightings to certain joints in order to promote movements in these joints, however the results again veered towards the stilted side.

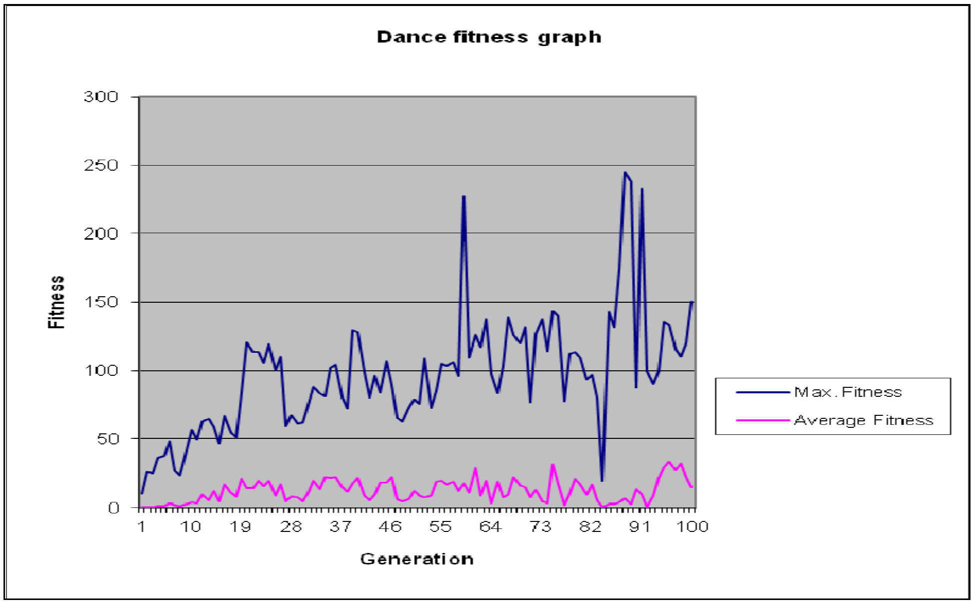


Figure 1. Maximum and average fitness values over 100 generations, averaged over three runs

In order to promote some movement in all of the robots joints an upgraded fitness function was devised. This is based on the product of the movements of all of the 18 joints of the robot over a specified time frame (approx. 40 seconds), or until the robot falls over. The product of the movements of the joints is also multiplied by the length of time the robot remains standing to the power of six in order to promote upright dancing. These values were arrived at after some experimentation. As this value can get very large because of the product function, we take the log of this value to keep it in a more reasonable range and then square this value to get the final fitness (typically a value now in the range 1-400). With the upgraded fitness function the onus is on providing at least some movement in all of the joints or else one of the product terms will reduce to close to zero, resulting in a very low fitness for this robot.

This updated fitness function is broadly in keeping with aspects of Krasnow and Chatfields' "Performance Competence Evaluation Measure" for the assessment of the qualitative aspects of human dance performance [15], which, in turn is broadly based on Chatfield and Byrnes' "Aesthetic Competence Evaluation" measure [16]. One of the key components of these measures in the assessment of dance performance in humans is in the "Limb Energy" aspect of the "Full Body Involvement" assessment where [there are]:

"No displays of "dead" or unattended body segments when focus of the movement is elsewhere, resulting in all body segments being energized,regardless of how minimal the movement is" [15]

V. EXPERIMENTAL RESULTS

Using the updated fitness function described above we ran three runs of the evolutionary algorithm on a population size of 100 individuals over 100 generations and averaged the results. Each run took about 8 hours actual simulation time. The best and average fitness values are shown in Fig. 1.

Note that maximum fitness values may go down as well as up from generation to generation as each robot inherits the initial values of its joint positions from the final position of the joints in the previously evaluated robot; this ensures a certain degree of robustness in the dances generated [14]. Any fitness value over 120 corresponds to a moderately effective dance routine. In all three runs dances of relatively high fitness (>250) were evolved within the 100 generations. A variety of different dance-like movements were evolved by the genetic algorithm. Fig. 2 and Fig. 3 show two typical dances that are generated using the procedure described. Fig. 4 shows the dance sequence from Fig. 3 as transferred to the Bioloid humanoid. This transfer was done using a semi-automated process, which translates the values for joint angles generated by Webots to the correct joint angles for the Bioloid humanoid. These motions can then be played using the RoboPlus software supplied with the Bioloid kit. It is also possible to download the motions to the robot for viewing dances without requiring a connection to the computer. The transfer to the real robot was quite faithful to the simulation, but not perfect, being slightly unstable at one point in the dance sequence depending on the floor conditions.

This dance (on the real robot) was demonstrated to a number of observers, including a class of AI students, and several commented on the lifelike nature of the dance evolved and the fluidity of the generated motions. While not perfect, certainly, as a proof of concept, this was a satisfactory result.

VI. Discussion and Conclusions

In this paper we have introduced an approach to the development of humanoid robot dance using an evolutionary technique which is neither imitation learning or interactive-evolutionary-computation based. Following a brief discussion of some of the other methodologies which have been used to generate dance movements we introduced our software and hardware platforms, the Webots simulator used to evolve the dance behaviours, and the Bioloid humanoid robot on which the final evolved dances were implemented.

We then discussed the creation of the fitness function which was arrived at for dance, together with the rationale behind its development. As we were not using subjective input in the selection of individual dances, or basing the dance on some existing dance, this process required several iterations together with an analysis of the methods used by human judges for the qualitative measurement of human dance.

Experimental results were then presented of dances as evolved in simulation in the Webots simulator using our updated model of the Bioloid humanoid robot and subsequently transferred to the actual Bioloid humanoid. A minor "reality gap" is observed in this transfer, however due to the accurate physics simulation of the Webots simulator, and the fidelity of our model to the actual Bioloid humanoid, it is possible to observe quite a faithful replication of the evolved dances on the real robot.

Observers have commented on the fluid and non-robotic nature of the movements evolved. We do not, of course, suggest that this methodology should completely replace the more "traditional" methods of generation of humanoid robot dance, such as learning by imitation/ observation and interactive evolutionary computation. We do, however, suggest that it could be used as a complementary methodology for the generation of novel and aesthetically pleasing humanoid dance moves.

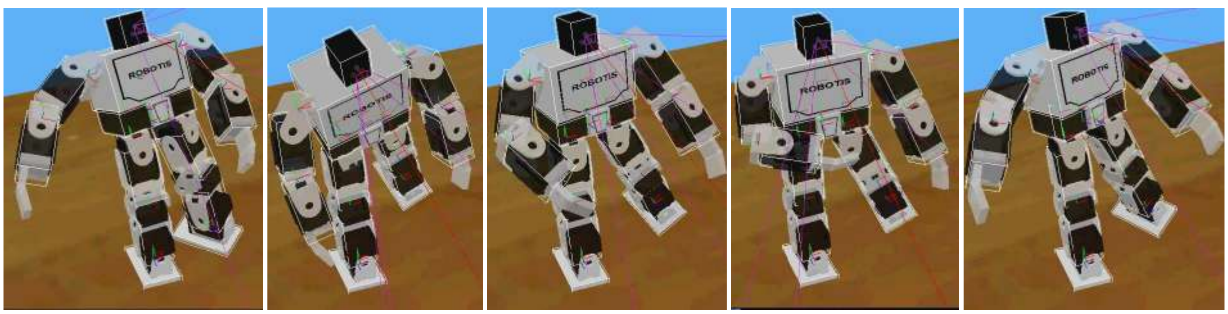


Figure 2. A sequence of evolved dance moves as generated in the Webots simulator

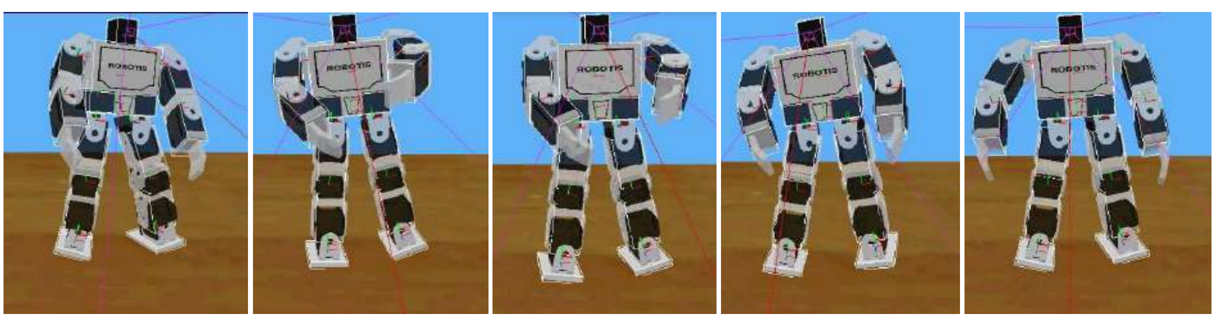


Figure 3. Another sequence of evolved dance moves as generated in the Webots simulator

Figure 4. The sequence of evolved dance moves from Figure 3 as transferred to the Bioloid humanoid robot

ACKNOWLEDGMENT

Acknowledgment is due to Aodán Donohue for his work on the updated version of the Webots Bioloid humanoid model as used in this paper.

REFERENCES

[1] M. Eaton, " Evolutionary humanoid robotics : past, present and future", in Pfeifer R., Bongard J.B., Lungarella (Eds.) 50 Years of Artificial Intelligence, Festschrift, LNAI 4850,2007, pp. 42-53.

[2] A. Nelson, G. Barlow, L. Doitsidis, "Fitness functions in evolutionary robotics: A survey and analysis". Robotics and Autonomous Systems 57(4), 2009 , pp.345-370.

[3] H. Takagi, "Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation". Proc. of the IEEE. ,Vol 89, No. 9, 2001, pp.1275-1296

[4] O. Michel, "Webots: Professional Mobile Robot Simulation", International Journal of Advanced Robotic Systems, Vol. 1, No. 1,2004, pages 39-42.

[5] J. Takamatsu, T. Shiratori, S. Nakaoka, S. Kudoh, A. Nakazawa, F. Kanehiro, and K. Ikekuchi. "Entertainment robot: Learning from observation paradigm for humanoid robot dancing." In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) Workshop “Art and Robots”,November 2007, pp. 45-56.

[6] R Dawkins, "The Blind Watchmaker: Why the Evidence of Evolution Reveals a Universe Without Design." W. W. Norton ,1986.

[7] K. Sims, "Evolving virtual creatures." In: SIGGRAPH '94: Proc. of the 21st annual conf. on Comp. graphics and interactive techniques, New York, NY, USA, ACM(1994) pp. 15-22

[8] K. Sims, "Artificial evolution for computer graphics." In: Proc. of SIGGRAPH, ACM Press (July 1991) pp. 319-328

[9] G. A. Dubbin and K. O. Stanley, "Learning to Dance through Interactive Evolution" In: Proceedings of the 8th European event on Evolutionary and Biologically Inspired Music, Sound, Art and Design (EvoMUSART 2010). New York, NY: Springer, 2010.

[10] M. Vircikova, Z. Fedor, Peter Sincak, "Design of Verbal and Non-Verbal Human-Robot Interactive System", 11th IEEE-RAS International Conference on Humanoid Robots Bled, Slovenia, 2011.

[11] J. Or, "Towards the Development of Emotional Dancing Humanoid Robots" Int J Soc Robot (2009) 1: pp. 367–382.

[12] T. Yanase, H. Iba, "Evolutionary Motion Design for Humanoid Robots", Frontiers in Evolutionary Robotics, (2008) pp. 567-586.

[13] J.L. Hanna "To dance is human: a theory of nonverbal communication." (1987) University of Chicago Press, Chicago

[14] M. Eaton." Evolving Humanoids: Using Artificial Evolution as an Aid in the Design of Humanoid Robots", Frontiers in Evolutionary Robotics, Hitoshi Iba (Ed.) ,2008, ISBN: 978-3-902613-19-6, InTech.

[15] D. Krasnow and S.J. Chatfield, "Development of the "Performance Competence Evaluation Measure" Assessing Qualitative Aspects of Dance Performance" , Journal of Dance Medicine & Science • Volume 13, Number 4, 2009, pp. 101-107.

[16] S. J. Chatfield, W.C. Byrnes . "Correlational analysis of aesthetic competency, skill acquisition and physiologic capabilities of modern dancers". In: 5th Hong Kong International Dance Conference Papers. Hong Kong: The Secretariat of the Hong Kong Academy for the Performing Arts, 1990, pp. 79-100

Bridging the Reality Gap — A Dual Simulator Approach to the Evolution of Whole-Body Motion for the Nao Humanoid Robot

Malachy Eaton
Dept. Computer Science and Information Systems, University of Limerick, Limerick, Ireland

Keywords: Evolutionary Algorithms, Humanoid Robotics, Ball Kicking, Evolutionary Robotics, Evolutionary Humanoid Robotics, Whole-Body-Motion.

Abstract: We describe a novel approach to the evolution of whole-body behaviours in the Nao humanoid robot using a multi-simulator approach to the alleviation of the reality gap issue. The initial evolutionary process takes place in the V-REP simulator. Once a viable whole-body motion has been evolved, this evolved motion is subsequently transferred for testing onto another simulation platform – Webots. Only when the evolved kicking behaviour has been demonstrated to also be viable on the Webots platform is this behaviour then transferred onto the real Nao robot for testing. This eliminates the time-consuming process of transferring behaviours onto the real robot which have little chance of successfully crossing the reality gap, and also minimises the potential for damage to the real Nao robot and/or it's environment. By using this novel approach of employing two different simulators, each with its own individual strengths and weaknesses, we reduce the likelihood that any individual behaviour will be able to exploit individual simulators' weaknesses, as the other simulator should pick up on this weak point. Using this procedure we have successfully evolved ball kicking behaviour in simulation, which has transferred with reasonable fidelity onto to the real Nao humanoid.

1 INTRODUCTION

The field of humanoid robotics addresses the creation of mobile robots that are broadly humanlike in their gross anatomy and/or aspects of their behaviour. Humanoid robots have several advantages, not least of which is their potential ability to operate in environments designed for humans, thus potentially having the ability to handle tasks that may be time-consuming, distasteful, or even dangerous for humans to perform (Eaton, 2015).

A robot in common use today by researchers into human-like behaviours is the Nao humanoid robot from Aldebaran Robotics (Gouaillier et al., 2009). This robot has up to 25 degrees of freedom and stands 58cm tall. A version of this robot is used in the RoboCup Standard Platform League (SPL), with the eventual avowed aim of producing, by the year 2050 a humanoid robot team that will be able to take on (and beat) the current human World Cup champions (Kitano and Asada, 1998) (Kitano et al., 1998). Central, of course, to being able to take up this challenge is the development of an effective kicking action, which is the area we address in this paper. While some work has been done to date on the automatic generation of kicking motions through parameter optimisation or other means (e.g. (Jouandeau and Hugel, 2014) ,(Li et al., 2015)), little work has been done on the direct evolution of individual joint motions for the robot, which is the approach we take. In general the field of evolutionary robotics seeks to evolve some, or all aspects of a robots controller and/or morphology (Nolfi and Floreano, 2000), (Bongard, 2013).

Much of the work in the area of generating soccer-centric skills has involved simulated robots for the RoboCup3D simulation league (Depinet et al., 2014), however our emphasis is on the evolution of behaviours which can be transferred effectively onto the real robot.

1.1 The "Reality Gap" Issue

A major issue that arises in this regard is the so-called "reality gap"; that is the potential disparity between evolved (or otherwise generated) behaviours in simulation, and their actual implementation on the real robot. This can be of particular importance in the

Eaton, M.
Bridging the Reality Gap — A Dual Simulator Approach to the Evolution of Whole-Body Motion for the Nao Humanoid Robot.
DOI: 10.5220/0006052301860192
In *Proceedings of the 8th International Joint Conference on Computational Intelligence (IJCCI 2016) - Volume 1: ECTA*, pages 186-192
ISBN: 978-989-758-201-1

evolution of behaviours for multi-jointed robots with many degrees of freedom, as in the case discussed in this paper. Various approaches have been taken to alleviate this issue, including the transferability approach (Koos et al., 2013), the grounded simulated learning approach (Farchy et al., 2013), the leveraging multiple simulators approach (Boeing and Bräunl, 2012), combining evolution in simulation with pre-programmed behaviours (Duarte et al., 2012), using an EA to tune the parameters of a simulator (Laue and Hebbel, 2009), fitness function correction interleaving simulated and real data (Iocchi et al., 2007), coevolution of controller and simulator (Lipson et al., 2006), (Bongard and Lipson, 2004), the "back to reality approach" (Zagal and Ruiz-Del-Solar, 2007), (Zagal et al., 2004), the online adaptation approach (Floreano and Urzelai, 2001), the envelope of noise approach (Jakobi, 1997a), (Jakobi, 1997b), and scaled experimentation (Eaton, 2015).

Although there has been work done to date on leveraging the effects of multiple physics simulators (Boeing and Bräunl, 2012), (Boeing, 2009) for evolutionary robotics experiments, to our knowledge this is one of the few, if any, which utilises the advantages of using multiple simulation packages, rather than just the core physics engines.

1.2 Simulators used — Webots and V-REP

The two simulation packages we use are Webots (Michel, 2004) and the Virtual Robot Experimentation Platform (V-REP) (Freese, 2010). Both of these packages have been used extensively in the simulation of a wide variety of robots including wheeled and legged robots of a variety of types, and also for the simulation of humanoid robots. Webots, which in original form dates from 1996, is one of the longest running simulators in continuous development suited for the detailed simulation of complex robotic environments. V-REP is a more recent arrival dating from around the start of this decade, and which describes itself as the Swiss army knife among robot simulators; an example scene from the V-REP simulator is given in Fig. 1. Regarding physics engines Webots relies on the Open Dynamics Engine (ODE), while V-REP provides a choice of 4 engines, ODE, the Bullet physics library, the Vortex Dynamics Engine and the Newton Dynamics engine. For the work described here we utilise the Bullet physics engine.

1.3 Overall Approach

Our approach, then, is to run the evolutionary experiments on a simulated Nao robot in the V-REP package, and then to transfer successfully evolved controllers into the Webots environment for further testing and validation of their overall performance.

One advantage of our approach is that it is unlikely that simulation weaknesses that would manifest themselves on one simulator would occur on the other , and vice versa. Another advantage of our approach is that while Webots is a proprietary simulation package, a fully functional version of V-REP is freely available for non-commercial use. We have observed through experimentation that a significant proportion of behaviours evolved using the V-REP platform do not transfer successfully to the real Nao robot. As the process of transferral to the real robot can be quite time-consuming , and as the potential for damage to the robot and/or its environment on execution of an incorrectly evolved motion involving quite rapid whole-body motion such as kicking is nontrivial, it is highly desirable only to transfer motions to the real Nao robot which have a high probability of success.

We have observed that it is very unlikely that a behaviour evolved in V-REP, but that fails to operate successfully in Webots will transfer onto the real robot with any degree of fidelity, however if validated in the Webots simulator a high percentage should transfer with reasonable accuracy. Preliminary experimental verification of this observation is discussed in section 3.

Another advantage is that while similar models of the Nao robot are used in each simulator, there are certain differences. For example, it is known that certain problems exist in the precise positioning of the centre of mass (COM) of some parts of the simulated Nao in V-REP. Again, by using multiple simulators the expectation is that exactly similar problems will not exist in all simulators.

While the work in (Boeing and Bräunl, 2012), (Boeing, 2009) involved a parallel evolutionary process, with each individual being tested in parallel on several physics simulators, and the results obtained from the evaluations being combined to generate an overall fitness for the individual, we employ a serial evolutionary process, with each individual in a generation being evaluated initially on the V-REP simulator as part of the evolutionary process. Only successful individuals are then transferred to the Webots simulator for validation of their performance before being then transferred to the real robot.

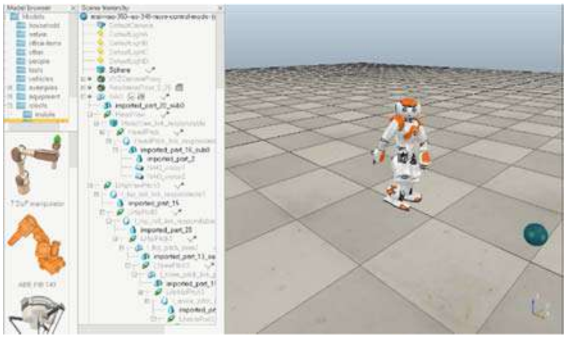


Figure 1: An example scene from the V-REP simulator. On the far left are examples of some of the robots that can be simulated, next to this is the scene hierarchy for the current scene, on the right is an example of an evolved kick.

2 FITNESS FUNCTION

For the evolution of ball-kicking behaviour we base our fitness function f on the distance travelled by the ball in the forward direction in the time allowed for each evaluation cycle. If the robot falls over mid-cycle we base the fitness on the distance travelled by the ball until the robot falls. This is to encourage stable and replicable kicking motions which should not cause undue strain to the real robot when transferred from simulation. So, if the robot fails to move the ball in the forward direction in the period in which it remains upright, or the time limit (T) expires, the fitness function is simply

$$f=100*t \qquad (1)$$

where t is the time that the robot remains upright (t=T if the robot does not fall in the experimentation period). T is set at 5 seconds for all the experiments described here. If the robot does manage to move the ball some distance in the forward direction the fitness is then given by

$$f=(1+d)*100*t \qquad (2)$$

where d is the distance travelled by the ball. This fitness function is designed in order to reward both the robot remaining upright, and the ball being moved in the forward direction. We note, of course, that no constraint of remaining upright is placed on human soccer players, it may indeed even be advantageous to a player to conclude a kicking motion on the ground in certain circumstances. The ball used in these experiments was of roughly similar diameter to that used in the RoboCup Standard Platform League (SPL).

3 EXPERIMENTAL DETAILS

3.1 Genome Composition

The genome length is 416 bits in total. This comprises 4 bits per joint angle (allowing for a total of 16 different angle positions per joint) for each of the 24 modelled joints of the robot, for each of 4 keyframe values. 4 keyframes were chosen as it was considered that this would be a sufficient number to characterise a complete kicking motion. While a certain amount of a-priori knowledge was involved in this decision, very little was specified about the joint values associated with each keyframe, apart from the fact that each joint has to keep within the maximum and minimum ranges as given by the specifications for the physical Nao robot. These maximum and minimum ranges are then modified by the joint restriction values evolved in each individual robots' genome as discussed below. Lower values of this 16-bit parameter correspond to higher joint ranges.

Typically this parameter starts at quite a low value early in the evolutionary process as the robot tends to move in a "thrashing" fashion, which may, or may not, cause ball movement. This value then increases as the robot restricts its joint movement range in order to increase the probability of not falling over due to "thrashing" motions. As the evolution progresses this value typically decreases gradually as the robot "frees up" its joints in order to more effectively perform the actions required (Eaton, 2007), (Eaton, 2013). As an example of this progression, for the experiment detailed in the next section the average value of this parameter for the best genome of generation 1 was 2.41, rising to a maximum value of 3.91 in generation 124, and reducing gradually to a value of 3 in generation 500.

A final 16 bits encodes movement durations for each of the 4 keyframes.

3.2 Keyframe Interpolation

The interpolation between the keyframe values is carried out by the V-REP and the Webots simulators themselves using inbuilt functions within each simulator. Once a sequence of 4 keyframes is completed the process cycles over until the time limit is exceeded or the run terminates for some other reason (the robot falling over).

3.3 Evolutionary Algorithm and Robot Control

The code for the evolutionary algorithm and robot control software in the V-REP simulator is written in the Lua programming language, while the corresponding controller for the Webots simulator and subsequent transfer to the real robot is written in Python. This transferral is semi-automated at present, however it is planned to fully automate this process in the future.

A population size of 120 was used using a mutation rate of 0.01 and a crossover probability of 0.2. These values were arrived at after some experimentation. The genetic algorithm employs tournament selection, and single-point crossover. It also employs elitism, where the best individual of each generation is guaranteed safe passage to the next generation.

4 EXPERIMENTAL RESULTS

4.1 Evolution of Kicking Behaviour

Three runs over 500 generations were performed, each taking about a day to complete on a 64-bit Dell 2.3GHz XPS 15 computer with an Intel i7 quad-core CPU and 16GB of RAM. The results obtained were then averaged to produce the fitness graph as shown in Fig.2.

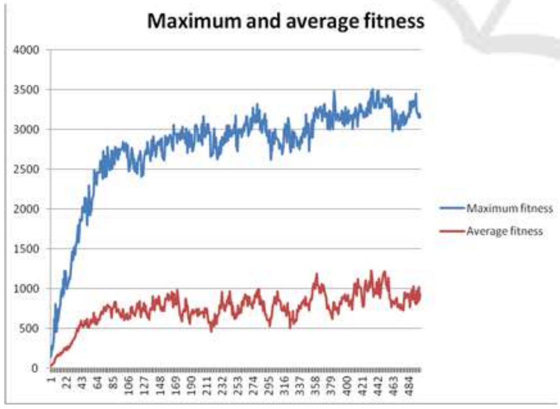


Figure 2: Maximum and average fitness, averaged over three runs for 500 generations for the evolution of ball kicking behaviour.

An effective kicking behaviour involves learning to stand on one foot, and the maintenance of balance on this foot while delivering a substantial blow to the ball with the other foot. For our work we also wish to maintain this balance (i.e. the robot does not fall over on completion of the kicking motion), if possible. Once the robot has learned to maintain its balance its kicking efficiency (as measured by the distance travelled by the ball) increases quite rapidly up to about generation 100, with more modest gains thereafter. Fig. 3 gives an example of an evolved kick as modelled in the Webots simulator.

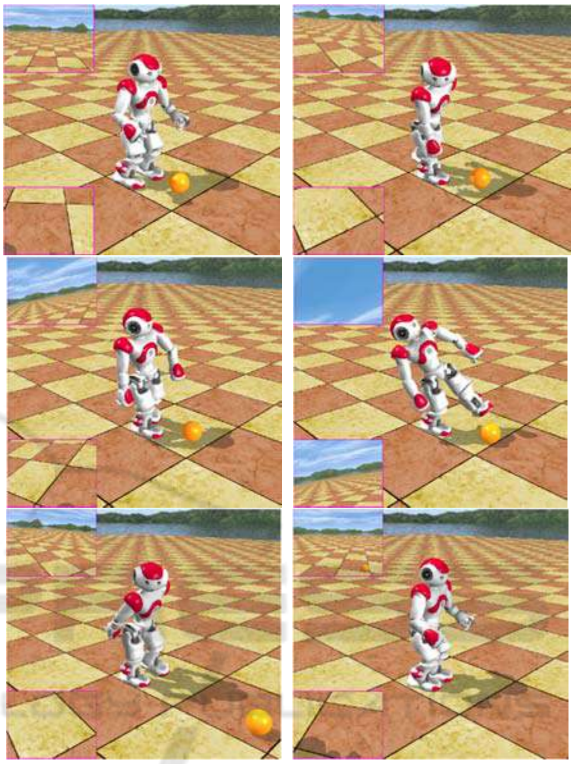


Figure 3: An example of an evolved kick, as transferred from V-REP to the Webots simulator; read from *top left* to *bottom right*.

Fig. 4 then shows this kick as transferred to the Nao humanoid robot using the procedure outlined earlier. The main portion of this kicking motion transfers directly onto the robot without need for human intervention. The only minor point of instability occurs in the final steadying motion before the robot comes to rest on completion of the kick, we conjecture that this is due to a friction mismatch between the surface the real robot rests on, and the values used in the V-REP and Webots simulators. However the majority of the kicking behaviour transferred directly to the robot resulting in an effective and quite human-like striking action. The entire behaviour sequence depicted in Fig.4 took place without the need for human intervention.

Effective kicking behaviour evolved in all three runs. In one of the runs predominately right-footed kicks were evolved, whereas a left-footed kick, as demonstrated in Fig. 3 and Fig. 4, was evolved in the other two runs, thus demonstrating the robustness and flexibility of our approach.

It should be noted that tests conducted on the real Nao robot were conducted at a reduced speed than the V-REP and Webots environments to reduce the likelihood of damage to the robot; it was also found that a behaviour was more likely to transfer successfully from simulated to real robot if conducted at a lower speed. However this reduction of speed was not, in general, found to cause a major diminution in the effectiveness of the behaviours evolved.

4.2 Validation of Our Approach

As an additional preliminary test of the effectiveness of our approach we chose 10 genomes at random from the first of the 3 runs. All of the fitness's of the evolved behaviours were the best of their generation and were around the 3000 mark, corresponding to an effective kick as evolved in the V-REP simulator.

Of these 10 behaviours two resulted in consistently unstable behaviour over several evaluations in the Webots environment. When transferred to the real Nao humanoid unstable motions also resulted, with the robot falling over and having to be manually restrained to avoid damage to the robot.

Of the remaining 8 motions three resulted in motions in which the either the robot either fell over in one of the Webots test evaluations, or, while not falling over exhibited significant instability at some point in the sequence. Of these three test runs, when transferred to the real Nao robot, two resulted instability (robot falling over), and one corresponded to an effective kicking motion, however exhibiting significant instability towards the end of the motion sequence.

Five of the 10 motions tested in the Webots simulator resulted in effective stable kicking motions. Of these 5 motions when transferred to the real Nao, all but one resulted in successful stable kicks.

Based on the results of these initial experiments we would not now generally consider testing any evolved motion on the real Nao robot that had not been successful in both the V-REP and Webots environments, to avoid potential strain on the Nao robots' actuators and/or actual damage to the robot or its environs.

5 CONCLUSIONS

In this paper we have demonstrated the evolution of kicking behaviour in the Nao humanoid robot. Using a novel dual-simulator approach with a fitness function based solely on the stability of the robot and the distance travelled by the ball, effective kicking behaviours were developed which were demonstrated to transfer, with reasonable fidelity, to the real Nao robot.

To our knowledge this is the first time a multi-simulator approach to the evolution of robot behaviours in the manner described in this paper. Also to our knowledge this is one of the few, if not the only, work which involves the evolution of kicking behaviours for direct transferral to the real Nao humanoid, rather than for use in the RoboCup simulation environment.

Figure 4: The evolved kick from Fig.3 as transferred to the real Nao robot. Compare these whole-body motions with the first four frames of Fig. 3.

We intend to extend the approach presented in this paper to the evolution of further behaviours of a more complex nature, including involving multiple robots.

ACKNOWLEDGEMENTS

My thanks to Jason Brownlee for his Lua GA implementation which provided the inspiration for our GA coding. My appreciation also to Norah Power for her assistance in the experimental phase of this work. Finally, also thanks to the reviewers of this paper for their helpful and constructive comments.

REFERENCES

Boeing, A., 2009. *Design of a Physics Abstraction Layer for Improving the Validity of Evolved Robot Control Simulations.* Ph.D Dissertation, School of Electrical, Electronic and Computer Engineering, The University of Western Australia, WA, 2009.

Boeing, A., and Bräunl, T., 2012. Leveraging multiple simulators for crossing the reality gap. In: *Control Automation Robotics and Vision (ICARCV), 2012 12th International Conference on* (pp. 1113–1119). IEEE.

Bongard, J. C. , 2013. Evolutionary robotics. *Communications of the ACM*, 56(8), 74–83.

Bongard, J. C., and Lipson, H., 2004. Once more unto the breach: Co-evolving a robot and its simulator. In: *Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems* (ALIFE9) (pp. 57–62).

Depinet, M., MacAlpine, P., & Stone, P. ,2014. Keyframe sampling, optimization, and behavior integration: Towards long-distance kicking in the robocup 3d simulation league. In *RoboCup 2014: Robot World Cup XVIII* (pp. 571-582). Springer International Publishing.

Duarte, M., Oliveira, S., and Christensen, A. L., 2012. Automatic synthesis of controllers for real robots based on preprogrammed behaviors. In: *From Animals to Animats 12* (pp. 249–258). Springer Berlin Heidelberg.

Eaton, M., 2007. Evolutionary humanoid robotics: past, present and future, In: *50 Years of Artificial Intelligence: Essays Dedicated to the 50th Anniversary of Artificial Intelligence LNAI 4850*, Springer, pp. 42–53.

Eaton, M., 2013. An Approach to the Synthesis of Humanoid Robot Dance Using Non-interactive Evolutionary Techniques. In: *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on* (pp. 3305–3309). IEEE.

Eaton, M., 2015. *Evolutionary Humanoid Robotics*. Springer Berlin Heidelberg.

Farchy, A., Barrett, S., MacAlpine, P., and Stone, P., 2013. Humanoid robots learning to walk faster: From the real world to simulation and back. In: *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems* (pp. 39–46). International Foundation for Autonomous Agents and Multiagent Systems.

Floreano, D., and Urzelai, J., 2001. Evolution of plastic control networks. *Autonomous robots, 11(3),* 311–317.

Freese, M.S., Singh, S, Ozaki, F, and Matsuhira N., 2010. Virtual robot experimentation platform v-rep: a versatile 3d robot simulator. In *Simulation, Modeling, and Programming for Autonomous Robots*, pages 51–62. Springer, 2010.

Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., Lafourcade, P., Mariner, B., Serre, J., and Maisonnier, B.,2009 . Mechatronic design of NAO humanoid. In: *Robotics and Automation, 2009. ICR'09. IEEE International Conference on* (pp. 769–774). IEEE.

Iocchi, L., Libera, F. D., and Menegatti, E. , 2007. Learning Humanoid soccer actions interleaving simul ated and real data, in: *Proc. of The Second Workshop on Humanoid Soccer Robots, IEEE-RAS 7th International Conference on Humanoid Robots*, Pittsburgh, 2007

Jakobi, N. , 1997b. Half-baked, ad-hoc and noisy: minimal simulations for evolutionary robotics. In: P. Husbands and I. Harvey, *Proceedings of the Fourth European Conference on Artificial Life*. Cambridge, MA: MIT Press.

Jakobi, N., 1997a. Evolutionary robotics and the radical envelope-of-noise hypothesis. *Adaptive Behavior, 6(2)*, 325–368.

Jouandeau, N., & Hugel, V. ,2014. Optimization of parametrised kicking motion for humanoid soccer player. In *Autonomous Robot Systems and Competitions (ICARSC), 2014 IEEE International Conference on* (pp. 241-246). IEEE.

Kitano, H., and Asada, M.,1998. RoboCup humanoid challenge: That's one small step for a robot, one giant leap for mankind. In: *Intelligent Robots and Systems, 1998. Proceedings, 1998 IEEE/RSJ International Conference on* (Vol. 1, pp. 419–424). IEEE.

Kitano, H., Asada, M., Kuniyoshi, Y., Noda, I., Osawai, E., and Matsubara, H., 1998. Robocup: A challenge problem for AI and robotics. In: *RoboCup-97: Robot Soccer World Cup I* (pp. 1–19). Springer Berlin Heidelberg.

Koos, S., Mouret, J. B., and Doncieux, S..,2013. The transferability approach: Crossing the reality gap in evolutionary robotics. *Evolutionary Computation, IEEE Transactions on, 17(1),* 122–145.

Laue, T., and Hebbel, M., 2009. Automatic parameter optimization for a dynamic robot simulation. In: *RoboCup 2008: Robot Soccer World Cup XII* (pp. 121–132). Springer Berlin Heidelberg.

Li, X., Liang, Z., & Feng, H. ,2015. Kicking motion planning of Nao robots based on CMA-ES. *In Control and Decision Conference (CCDC), 2015* 27th Chinese (pp. 6158-6161). IEEE.

Lipson, H., Bongard, J. C., Zykov, V., and Malone, E., 2006. Evolutionary Robotics for Legged Machines: From Simulation to Physical Reality. In: Arai, T. et al

(eds*.), Intelligent Autonomous Systems 9(IAS-9)*, pp.11–18.
Michel, O., 2004. Webots: Professional Mobile Robot Simulation. *International Journal of Advanced Robotic Systems, Vol. 1, No. 1*, 39–42.
Nolfi, S., and Floreano, D.,2000. *Evolutionary robotics. The biology, intelligence, and technology of self-organizing machines*. MIT Press.
Zagal, J. C., and Ruiz-Del-Solar, J., 2007. Combining simulation and reality in evolutionary robotics. *Journal of Intelligent and Robotic Systems, 50(1)*, 19–39.
Zagal, J. C., Ruiz-del-Solar, J., and Vallejos, P., 2004. Back to reality: Crossing the reality gap in evolutionary robotics. In: *IAV 2004 the 5th IFAC Symposium on Intelligent Autonomous Vehicles*, Lisbon, Portugal.

frontiers in Robotics and AI

REVIEW
published: 03 November 2021
doi: 10.3389/frobt.2021.744590

Check for updates

Robot Evolution: Ethical Concerns

Ágoston E. Eiben[1,2], Jacintha Ellers[3], Gerben Meynen[4,5] and Sven Nyholm[6]*

[1]Department of Computer Science and Ecological Science, Vrije Universiteit Amsterdam, Amsterdam, Netherlands, [2]Department of Electronic Engineering, University of York, York, United Kingdom, [3]Department of Ecological Science, Vrije Universiteit Amsterdam, Amsterdam, Netherlands, [4]Department of Philosophy, Vrije Universiteit Amsterdam, Amsterdam, Netherlands, [5]Department of Law, Utrecht University, Utrecht, Netherlands, [6]Department of Philosophy and Religious Studies, Utrecht University, Utrecht, Netherlands

Rapid developments in evolutionary computation, robotics, 3D-printing, and material science are enabling advanced systems of robots that can autonomously reproduce and evolve. The emerging technology of robot evolution challenges existing AI ethics because the inherent adaptivity, stochasticity, and complexity of evolutionary systems severely weaken human control and induce new types of hazards. In this paper we address the question how robot evolution can be responsibly controlled to avoid safety risks. We discuss risks related to robot multiplication, maladaptation, and domination and suggest solutions for meaningful human control. Such concerns may seem far-fetched now, however, we posit that awareness must be created before the technology becomes mature.

Keywords: evolutionary robotics, evolutionary design, ethics, meaningful human control, responsibility gaps, real-world robot evolution, morphological robot evolution

OPEN ACCESS

Edited by:
Claudio Rossi, Polytechnic University of Madrid, Spain

Reviewed by:
Robert H. Wortham, University of Bath, United Kingdom
Matthew Studley, University of the West of England, United Kingdom

****Correspondence:***
Ágoston E. Eiben
a.e.eiben@vu.nl

Specialty section:
This article was submitted to Robot Learning and Evolution, a section of the journal Frontiers in Robotics and AI

Received: *20 July 2021*
Accepted: *11 October 2021*
Published: *03 November 2021*

Citation:
Eiben ÁE, Ellers J, Meynen G and Nyholm S (2021) Robot Evolution: Ethical Concerns. Front. Robot. AI 8:744590. doi: 10.3389/frobt.2021.744590

INTRODUCTION

Surprisingly, the idea of robot evolution is one hundred years old. The famous play by Karel Čapek that coined the word "robot" was published in 1920 (Čapek 1920). Towards the end of the play the robots are at the verge of extinction and one of the humans, Alquist, advises them: "*If you desire to live, you must breed like animals.*" In 1920 this was a fantastic idea–as in: impossible. In today's world with rapidly proliferating artificial intelligence and robotics it is still a fantastic idea, but not impossible anymore.

Towards the end of the twentieth century the principles of biological evolution were transported to the realm of technology and implemented in computer simulations. This brought on the field of Evolutionary Computing, and evolutionary algorithms proved capable of delivering high quality solutions to hard problems in a variety of scientific and technical domains, offering several advantages over traditional optimization and design methods (Ashlock 2006; de Jong, 2006; Eiben and Smith 2003). Evolutionary algorithms have also been applied to developing the morphology (the hardware "body") and controller (the software "brain") of autonomous robots, which resulted in a new field called Evolutionary Robotics (Nolfi and Floreano 2000; Bongard 2011; Vargas et al., 2014; Doncieux et al., 2015).

Up till now, work on evolutionary robotics has mostly been performed in computer simulations, safely confined to a virtual world inside a computer [e.g (Bongard 2011)]. Occasionally, the best robots in the final generation have been constructed and materialized in the real world (Lipson and Pollack 2000; Kriegman et al., 2020), but even in these cases the evolutionary process itself took place in simulation. Some studies have demonstrated self-reproducing physical machines, but the resulting system was not evolutionary because there was no inheritance and reproduction created identical clones without variation (Zykov et al., 2005). Research about robots that reproduce and evolve in the

BOX 1 | Robots evolving in the real world

To make robots evolvable selection and reproduction need to be implemented. Selection of "robot parents" can be done by evaluating the robot's behavior and allocating higher reproduction probabilities to robots that work well. For reproduction two facets of a robot should be distinguished, the ***phenotype*** that is the physical robot itself and the ***genotype*** that is the specification sheet, the robotic equivalent of DNA that describes and encodes the phenotype. Reproduction can then be defined through two principal steps. The first step is to create a new genotype that encodes the offspring. This step generates genetic variation either by a recombination operator that stochastically mixes the genotypes of two parents (sexual reproduction) or by a mutation operator that causes random changes in the genotype of one single parent (asexual reproduction). This step is a fully digital operation that can use existing methods from traditional Evolutionary Computation. The second step is the execution of the genotype-phenotype mapping, that is, the construction of the physical robot offspring as specified by the newly produced genotype. A crucial technical challenge in robot evolution lies in the second step, the production of offspring.

real world has been rare because of technical limitations in the (re)production of arbitrary robot shapes (Long 2012). In **Figure 1** we exhibit some of the landmarks of the history of robot evolution.

However, this situation is changing rapidly and after the first major transition from "wetware" to software in the 20th century, evolution is at the verge of a second one, this time from software to hardware (Eiben and Smith 2015). Recent advances in and integration of evolutionary computation, robotics, 3D-printing, and automated assembly are enabling systems of physical robots that can autonomously reproduce and evolve (Brodbeck et al., 2015; Jelisavcic et al., 2017; Vujovic et al., 2017; Hale et al., 2019; Howard et al., 2019; Ellery 2020). The key concepts behind robots evolving in the real world are explained in **Box 1**, while **Box 2** illustrates how the most challenging step of the process, robot reproduction, can be implemented. Two examples of existing robot reproduction facilities are shown in **Figure 2**. Such autonomous evolutionary systems incarnated in hardware offer advantages for applications as well as for fundamental research.[1]

For practitioners, evolution serves as an approach to adjust optimal robot designs on-the-fly in dangerous or inaccessible places [19], such as mines, nuclear power plants, or even extraterrestrial locations (see **Figure 3**). Additionally, evolving robots can be seen as hardware models of evolutionary systems [13]. Thus, they can be used as a new type of research instrument for testing hypotheses about biological processes (Nolfi and Floreano 2000) and deliver deeper understanding of universal evolutionary principles (Floreano and Keller 2010; Waibel et al., 2011). Autonomous robot evolution can thus be a game changer compared to evolutionary systems implemented in the digital realm (Eiben et al., 2012).

A key insight of this paper is that the science and technology of robot evolution are elevating the known concerns regarding AI and robotics to a new level by the phenomenon we call *second order engineering* or *second order design*. First order system engineering is the current practice where AI and robots are developed and engineered directly by humans. Evolutionary robot technology radically changes this picture because it introduces a new layer: instead of directly constructing a robotic system for a certain application, humans are constructing an evolutionary system that will construct a robotic system. Ethical, moral and safety concerns should therefore be converted into design principles and methodological guidelines for humans. The fundamental challenge here is the inherent stochasticity and complexity of an evolutionary system and the weakened influence of humans on the end product. This implies that all issues of the current discourse on AI and robot ethics remain valid [see, e.g. (Torresen 2018)], but that we also get new ones.

The new ethical challenges related to robot evolution are rooted in the inherent inefficiency and unpredictability of the evolutionary process. Evolution proceeds through the generation of heritable variation (recombination and mutation) in combination with selection that favors more successful forms at the cost of large numbers of failures (Futuyma 2013). Evolving robots in hardware through automated (re)production may therefore bring about a high number of arbitrary robot forms, which increases the chance of unintentionally creating robots with harmful behaviors. Moreover, key evolutionary changes often take place in the form of large unpredictable innovations that arise from rearrangements of existing characteristics for new functions (True and Carroll 2002). Such emergent evolution is highly unpredictable in both direction and magnitude, increasing the likelihood that evolving robots will have unexpected capacities.

Whenever there is a technology that is not directly under human control–technologies without a "steering wheel"–and whenever the process is unpredictable, questions about risks and responsibilities arise (Sparrow 2007; Hansson 2017; Nihlen Fahlquist 2017; Santoni de Sio and van den Hoven 2018; Nyholm 2020). Do the benefits of the new technology outweigh its possible adverse effects? If there are adverse effects, how can we minimize and control these? And, importantly, if things spin out of control, who is responsible? Answering these questions not only requires solutions from the field of robot evolution itself, but also raises ethical issues about the measures we should take to prevent harm. One could argue that such concerns are far-fetched. However, we posit that these issues must be addressed long before the technology emerges. Simply put: if we start thinking about mitigating these problems when they arise, then, most probably, we are too late (van de Poel 2016; Brey 2017).

Protecting Humans From Evolving Robots

It is hard to overstate the possible implications of the two key enabling features in evolving robots: self-replication and random change in robot form and behavior. First, self-replication allows robots to multiply without human intervention and thus would

[1]We do not consider evolutionary soft robotics here, because that field mainly focuses on actuators and sensors, not on fully autonomous, untethered (soft) robots.

BOX 2 | Robot (re)production

A robotic genotype obtained by mutating the genotype of one robot or recombining the genotypes of two parent robots encodes a new robot, the offspring. This offspring could be constructed by feeding the genotype to a 3D printer that makes a robot as specified by this genotype. However, currently there are no 3D printers that can produce a fully functional robot including a CPU, battery, sensors, and actuators. Arguably, this problem is temporary, and rapid prototyping of such components will be possible in the (near) future. A practicable alternative for now is to combine 3D printing, prefabricated functional components stored in a repository (e.g., CPUs, batteries, sensors, and actuators), and automated assembly. In such a system, the genotype specifies a number of 3D printable body parts with various shapes and sizes, the types, numbers and geometrical positions of the prefabricated body parts and the properties of an adequate software "brain" to control the given body. The production of a new robot can be done by industrial robot arms that retrieve the 3D printed body parts from the printers, collect the necessary prefabricated components from the storage, and assemble them into a working robot. After that, the software can be downloaded and installed on the CPU and the new robot can be activated.

raise the need for control over their reproduction. Second, mutation or random evolutionary changes in the design of the robots could create undesired robotic behaviors that may harm human interests. Before developing any new technology with such potentially large ramifications, we should determine the acceptability of its consequences and identify ways to anticipate unwanted effects (van de Poel 2016).

Several other fields of science have faced similar safety dilemmas during developments of new technology and subsequent experimentation. In health sciences, biomedical ethical dilemmas are typically evaluated using a principle-based approach, based on the four principles of Beauchamp and Childress (Beauchamp and Childress 2019): autonomy, non-maleficence (avoiding harm), beneficence, and justice. Within the context of technological experimentation, the concept of responsibility has been added (van de Poel 2016), and specifically in the field of Artificial Intelligence (AI), a call has been made for adding the property of "explicability" (Floridi et al., 2018). This property entails that when AI-powered algorithms are used to make morally-sensitive decisions, humans should be able to obtain "a factual, direct, and clear explanation of the decision-making process" (Floridi et al., 2018), or of the decision resulting from the algorithm (Robbins 2019).

In evolutionary robotics all of these principles have clear relevance, but, most pressingly, the risk of harm and the question of responsibility need to be considered in more detail. These, in turn, are intimately related to the crucial issue of control and the potential loss of it. In order for a particular human being or group of human beings to be responsible for some process or outcome, it is usually thought that they need to have some degree of control of the process or outcome. Moreover, loss of control can be viewed as a form of harm, because it is typically seen as undermining human autonomy, and it may compromise other values, such as well-being, which depend to some extent on our ability to control what happens around us.

Risk of Harm

The issue of risk in the field of AI has previously been considered in relation to control concerns associated with the development of superintelligence (Bostrom 2014; Russell 2019; Russell and Norvig 2020). A notable difference between superintelligence-related concerns and ER-related worries, however, is the perceived probability of the risk. Many people find the idea of superintelligence either inherently implausible or at least something we need not worry about in the short run (Gordon and Nyholm 2021; Müller, 2020). More precisely, people may feel that although an excellent AI chess or Go player is manifestly possible, artificial *general* superintelligence is much less likely to emerge.

In contrast, evolving physical robots need not possess human level intelligence; animal level intelligence in such robots could be sufficient to do significant harm because of their physical features. Even without much individual intelligence and power, the evolved robots could potentially collaborate efficiently and perform much more complex tasks together than they could on their own. In other words, similar to highly social animals such as ants and wasps in the natural world, the number and cooperation among robots could be decisive factors. Therefore, the plausibility of a harmful scenario with evolving robots is all but trivial, and issues of control and the potential loss of it should be considered.

The most difficult aspect in anticipating possible risks of evolving robots is that we would be dealing with an *evolving* system that is inherently and continuously changing. The risk of harm therefore needs to be evaluated for potential future trajectories of the evolutionary process, not only for the current robots. We distinguish three key types of risks associated with the evolutionary process, connected to reproduction, selection, and emergent evolution, respectively:

Multiplication risk: The robots can evolve at high reproduction rates, resulting in uncontrolled population growth. If the robot population becomes too large, resources such as space, energy, and raw materials like air or water may be (locally) depleted. This effect can be compared to a locust plague: a swarm's voracious feeding can completely devour agricultural crops over a vast area, leading to famine and starvation in the human population. While individual robots may not pose any significant risk, their high number and collective behavior can be dangerous.

Maladaptation risk: Evolving the robots for a specific task can lead to unwanted features or behaviors that benefit the robot's assigned task, but that may be harmful to human society. For instance, robots may attempt to dismantle houses to use the stones or cut car tires for the rubber. In the most extreme cases, robots could harm humans if they hinder robots in performing their tasks. This type of risk can evolve because selection is "blind," meaning the most effective solutions for the task will prevail, without taking other consequences of the evolved trait into account.

Domination risk: The robots could evolve to become the dominating "species," not as a direct effect of selection, but rather as an emergent feature of the robot's functionality

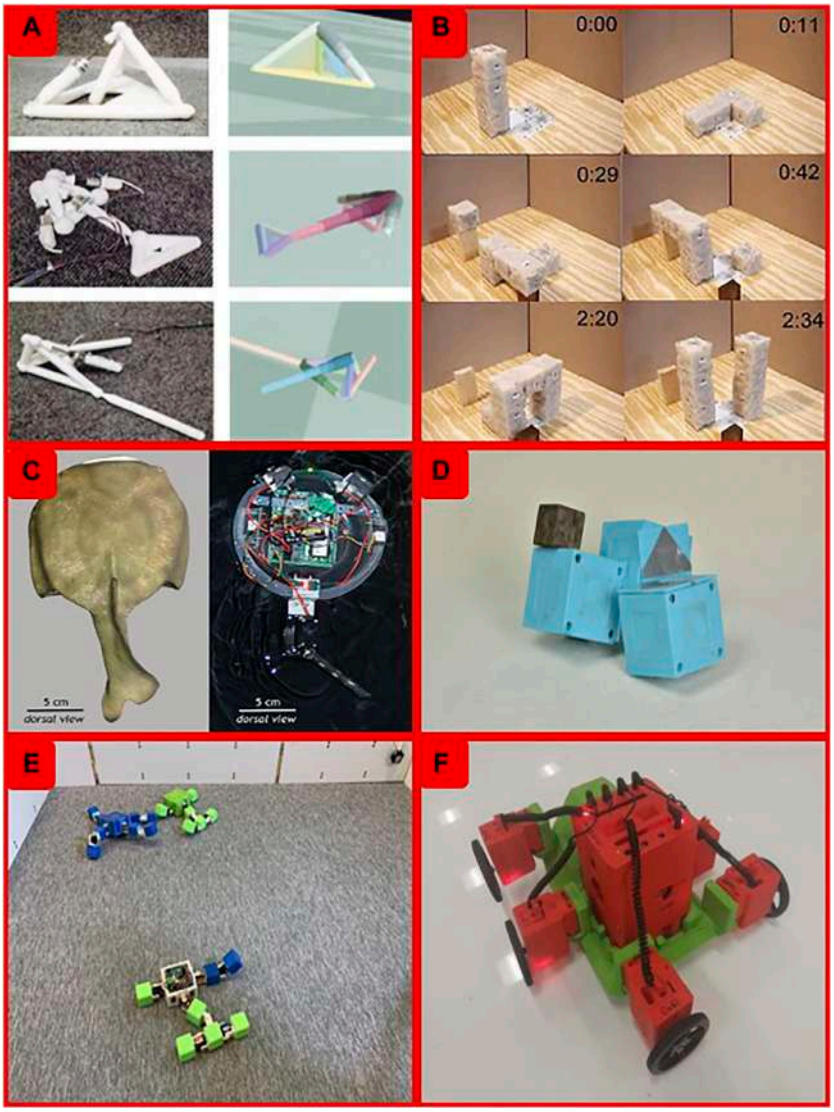


FIGURE 1 | Some of the landmarks of the history of robot evolution. We show examples of systems that demonstrated robot reproduction or evolution incarnated in the real world. **(A)** 2000: The GOLEM project (10) co-evolved robot bodies and controllers in simulation and fabricated the evolved robot afterwards. **(B)** 2005: A physical system based on Molecubes, demonstrated non-adaptive robots able to construct a replica of themselves (*12*). **(C)** 2012: Tadro robots (*13*) were used to verify a hypothesis about the evolution of Cambrian vertebrates. Consecutive generations were constructed and evaluated in real hardware. **(D)** 2015: Semi-automated construction of genetically encoded modular robots (15). Consecutive generations were constructed and evaluated in real hardware. **(E)** 2016: The Robot Baby Project (*17*) demonstrated the reproduction of genetically encoded robots. Robots co-existed in the same environment; the offspring was added there after "birth." **(F)** 2019: The Autonomous Robot Evolution Project (*18*) features hands-free construction of genetically encoded robots. The robots have sensors and can co-exist in the same environment. The robots shown in **(1A,C–E)** had no sensors. The robots shown in **(1C,D)** were constructed and evaluated one by one; the physical population consisted of one single robot at any time. The robots in **(1B)** are actually not evolvable, as there was no genetic encoding and the replica was an identical copy.

(Badyaev 2011). This can happen if they become superior to humans intellectually, physically, or "emotionally" (being stable and consistent). As a result, they might become benevolent influencers or decision makers, implicitly or explicitly arranging life for us. This effect can be compared to a parent-child relationship where the parent is better in understanding and anticipating situations and therefore confines the spatial range and activities of the child. Even though humans may not be physically harmed by the robot's dominating behavior, human autonomy would be, at least partly, diminished.

Meaningful Human Control

The risks of harm associated with robot evolution as identified above all arise from the underlying *control problem* of (semi) autonomous robotic systems. In the Artificial Intelligence literature, solutions to this control problem are often phrased in terms of *meaningful human control* (Santoni de Sio and van

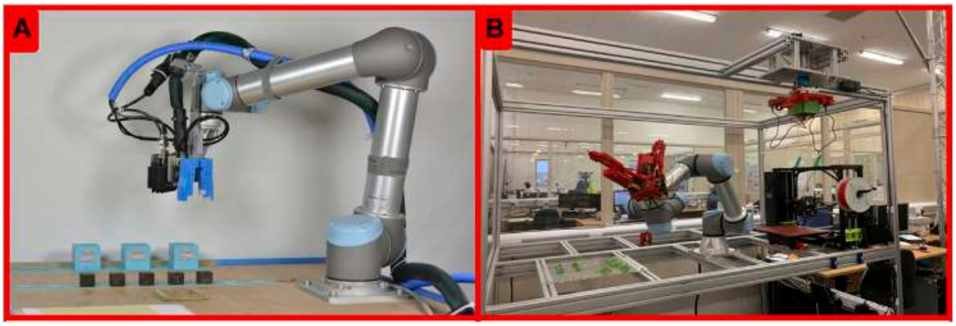


FIGURE 2 | Examples of robot reproduction facilities. Photos of two (semi) automated robot reproduction facilities. **(A)**: the system used in Cambridge (15). **(B)**: the one used in Bristol (18).

den Hoven, 2018). This term acknowledges that whereas there may be no direct control–e.g., a steering wheel in a car–it may still be possible to have *indirect control* allowing for allocation of responsibilities (Di Nucci 2020; Nyholm 2020). For evolving robots this would mean that precautionary design measures are required to control the evolutionary process itself. Such measures could include:

1) *Centralized, externalized reproduction.* A rigorous way of maintaining control over the system would be to set it up such that robot reproduction cannot take place "in the wild" but only in a centralized infrastructure–a reproduction center–where robot offspring can be made, for instance by 3D-printers and automated assembly facilities (Eiben et al., 2013; Hale et al., 2019). Limiting the reproduction to a single or a few centers not only allows keeping track of robot numbers, but also provides the option to restrict the number of robots produced per day. In addition, such a center could provide a possibility to test new robots for safety before releasing them into the outside world. Furthermore, a reproduction center can contain a "kill switch" that can be used to halt evolution by shutting down the reproduction process.
2) *Advanced prediction systems.* Complex simulations and prediction models could provide the necessary previews of the evolutionary process and the emerging features of the resulting robots. Such a "crystal ball," as Bostrom puts it, would allow humans to anticipate the developments and intervene if necessary (Bostrom 2014). To this end it is important to note that, contrary to natural organisms, robots can be monitored in detail. At the cost of some overhead for inspecting and logging the communications, actions, sensory inputs, and even the internal processes of the robots, a lot of data can be collected and utilized. To be realistic, modeling and predicting the complex evolutionary process of robots in the real world is currently beyond reach. In addition to practical constraints (data collection, data volumes, processing power) there can be fundamental limitations regarding the prediction of emergent behaviors in a population of evolving and interacting robots in environments that are dynamically changing and not fully known. However, meteorological and epidemiological simulations demonstrate that predictions need not be accurate to the finest details to be useful.
3) *Value loading.* Another option for control suggested by Bostrom (Bostrom 2014) is to instill certain properties inside the robot that make sure the robot does not set goals that are risky for humans. For instance, the system might be set up so that robots do not want to reproduce independently, so they will not "revolt" against the centralized reproduction center.

These control measures, meaningful as they are, can leave humans vulnerable because of the very nature of evolving systems, in which change is inherent. Evolving robots represent a whole new breed of machines that can and will change their form and behavior. This implies that robots could adapt their behavior to escape the implemented control measures. Therefore, controlling evolving robots is different from controlling the production of fixed entities, such as cars. One would therefore need to continuously adjust the control measures to stay ahead of evolutionary escape routes, not unlike a co-evolutionary arms race (Thompson 1994). In what follows, we highlight three possible evolutionary escape routes: two technology-related possibilities and one that exploits human emotional vulnerabilities and normative judgments.

First, the robots could develop solutions to circumvent the technological safeguards that have been put into place. A very unlikely, but conceivable escape route is the "Jurassic Park scenario," where the robots find an alternative way of reproducing outside the central reproduction facility. To mitigate this risk, additional reproductive constraints may be necessary, e.g., using an ingredient that is necessary for being viable and controlling its supply (Ellery and Eiben 2019). A more realistic way of escaping control is that robots stop sharing their operational data and thereby evade monitoring. This could partly be resolved by a mandatory data recorder built into all robots, similar to the flight recorders (a.k.a. black box) in airplanes (Winfield and Jirotka 2017; Winkle 2020).

Second, while Bostrom [36] suggests "value loading" for robotic and AI systems, in the case of evolving robot populations it is important to realize that it would be risky to

FIGURE 3 | Artist impression of evolving robots in space.

rely on the (current) features of individual robots. In an evolutionary process the robot's features undergo change. This does not mean that creating certain features (such as values or goals) in the robots is without merit, but it should be combined with some form of verification that the goals/values continue to be present in the newly produced robots. This requires new technologies that effectively combine immutable values with adaptable robot features and protocols for a thorough screening of "newborn" robots before they are allowed to leave the reproduction facility.

A third possibility for evolving robots to escape human control is non-technological, exploiting deep-seated emotional response patterns. Specifically, humans may grow fond of robots, developing feelings of "affection" towards them (Carpenter 2016; Darling 2017). This emotional vulnerability is probably the result of the long evolutionary history of humans, which has equipped our brains with various motivational and affective pathways tuned to human psychology (Damiano and Dumouchel 2018; Nyholm 2020). Consequently, we are responding to robots with brains and emotional sensitivities that are well-adapted to interacting with fellow human beings and familiar animals, but not necessarily adequate to responding sensibly to machines. Robots and other artificially intelligent technologies, therefore, may "push our Darwinian buttons" in ways that we may not upon reflection find suitable (Turkle 2004).

These sensibilities can be exploited if robots evolve features humans tend to like such as, possibly, big eyes, certain locomotion patterns or "lovely" sounds and gestures. Such features can increase attachment, undermine human controller's ability to remain objective and provide an evolutionary advantage on the long run. For instance, a robot could entice a human into supplying it with extra energy or allowing it to reproduce. Similarly, a "lovable" robot could prevent a human from switching off the robot or using the "kill switch" to shut down the evolution of the whole robotic species. These scenarios illustrate how emotions could get in the way of strict human control and induce an evolutionary bias [cf. (Bryson 2018)].

Filling the Responsibility Gap

The above-mentioned considerations concern ways of controlling the process of robot evolution. But there are more conceptual–ethical–concerns as well. Being able to ascribe responsibility is always important when risks are involved, both from an ethical and a legal point of view. The relevant form of responsibility here does not only have a backward-looking component (who can be blamed when things have gone wrong?), but is also forward-looking and clarifies who should do what in order to maintain control, e.g., mitigating risks and taking precautions (Nihlen Fahlquist 2017; Di Nucci 2020). Thus, a prominent issue is a potential responsibility gap. A responsibility gap occurs when there are significant risks of harm for which someone should take responsibility, but there is no obvious candidate to ascribe the responsibility to (Matthias 2004; Sparrow 2007;

Nyholm 2020). In the solutions above, the control envisioned will, at least in part, be exercised by humans. The crucial question is then how potential responsibility gaps might be filled.

At this point it may be instructive to refer to recent work by Santoni de Sio and Van den Hoven (Santoni de Sio and van den Hoven, 2018). They have developed a "track-and-trace" account of meaningful human control. The tracking part requires that the system behaves according to rules or paths that track human interests. In other words, the system should behave in a way that aligns with human values and interests. The tracing part requires that the robotic behavior can be traced back to at least one person who understands how the process works, as well as its moral and social significance. It might be added here that, ideally, this should work like when one is tracking and tracing a parcel: it should be possible to monitor how things are developing, just like one can monitor the journey of a parcel [(Nyholm 2020), p. 78].

The track-and-trace theory, understood as including the monitoring condition, looks promising from an ethical perspective for robot evolution. If the robot evolution is tracking human interests, if there are people who understand the process and its moral significance, and are able to monitor the robot evolution, then we can tentatively say that meaningful human control over this process has been achieved. If those conditions are fulfilled, that could help to fill any potential responsibility gaps.

The control solutions suggested above cover the "tracking" requirements from the track-and trace theory to a significant extent. The centralized, externalized reproduction centers would allow humans to monitor the numbers and types of robots produced each day, while the crystal ball would give insight into the future directions of the evolutionary path of the robots. Being able to monitor robot development in these ways, the humans involved would be able to observe whether human interests are being tracked. If not, they could use the "kill switch." The tracing part however, would need to be developed further as, at the moment, we do not have an appropriate level of understanding nor control of how the evolutionary process unfolds. At the same time, if studying these evolutionary processes in robots would deepen our scientific understanding of evolution, this could in effect help to also fulfil the tracing condition.

That being said, the big challenge here is, again, the inherent variability of an evolutionary system where new features emerge through random mutations and recombination of parental properties. Even though the whole system, specifically the genetic code (the robotic DNA), the mutation operators, and recombination operators are designed by humans, it is not clear to what extent these humans can be held responsible for the effects over several generations. On the positive side, let us reiterate that robots are observable, thus the genetic material and genealogy tree of an evolving population can be logged and inspected. In principle, it is possible to examine a newly created genotype (the robotic zygote) before the corresponding phenotype (the robot offspring) is constructed and destroy the genotype if it fails a safety test.

Protecting Evolving Robots From Humans

In the sections above, our main concern was to protect the human race from evolving robots. However, the matter can be inverted if we conceive of robots that can evolve and learn as a form of artificial *life*. Considering them as a form of life implies different kinds of ethical considerations (Coeckelbergh 2012; Bryson 2018; Gunkel 2018; Danaher 2020), which go beyond the issues of affection and attachment to individual robots as discussed above, and refer to the whole robotic population. The key is to see the robot population as a species that requires some *moral consideration*. Such an ethical view could be motivated by two arguments.

First, these robots have the possibility of reproduction, and in biology the crucial difference between life and non-life is reproduction. In addition, these robots share other characteristics with other life forms, such as movement and energy consumption. Second, the robots are not only able to reproduce; they themselves have also evolved. In other words, these robots are not (just) the result of human design, but of an evolutionary process. If humans, generally, start to feel that these robots are *forms of life*–albeit artificial–this could entail some perceived moral obligations, like we may feel we have obligations towards whales, dolphins, dogs, and cats. In other words, we may feel that these robots–and along with them, their evolutionary process–deserve some level of protection. This could raise the issue of robot rights, similarly to how we think about animal rights (Gellers 2020).

Second, it could be questioned whether certain control-interventions, such as the use of the "kill switch", are ethical regarding such forms of artificial life. An essential question here is if terminating evolutionary robots should be seen as switching off a machine or as killing a living being (Darling 2021). In any case, such moral considerations could potentially limit the possibilities of meaningful human control of robot evolution we have discussed.

CONCLUSION

Robot evolution is not science fiction anymore. The theory and the algorithms are available and robots are already evolving in computer simulations, safely limited to virtual worlds. In the meanwhile, the technology for real-world implementations is developing rapidly and the first (semi-) autonomously reproducing and evolving robots are likely to arrive within a decade (Hale et al., 2019; Buchanan et al., 2020). Current research in this area is typically curiosity-driven, but will increasingly become more application-oriented as evolving robot systems can be employed in hostile or inaccessible environments, like seafloors, rain-forests, ultra-deep mines or other planets, where they develop themselves "on the job" without the need for direct human oversight.

A key insight of this paper is that the practice of second order engineering, as induced by robot evolution, raises new issues outside the current discourse on AI and robot ethics. Our main message is that awareness must be created before the technology becomes mature and researchers and potential users should discuss how robot evolution can be responsibly controlled. Specifically, robot evolution needs careful ethical and methodological guidelines in order to minimize potential harms and maximize the benefits. Even though the evolutionary process is functionally autonomous without a "steering wheel" it still entails a necessity to assign responsibilities. This is crucial not only with respect to holding someone responsible if things go wrong, but also to make sure that people take responsibility for certain aspects of the process–without people taking responsibility, the process cannot be effectively controlled. Given the potential benefits and harms and the complicated control issues, there is an urgent need to follow up our ideas and further think about responsible robot evolution.

AUTHOR CONTRIBUTIONS

AE initiated the study and delivered the evolutionary robotics perspective. JE validated the biological soundness and brought the evolutionary biology literature GM and SN bridged the area of (AI) ethics and the evolutionary robotics context.

FUNDING

SN's work on this paper is part of the research program Ethics of Socially Disruptive Technologies, which is funded through the Gravitation program of the Dutch Ministry of Education, Culture, and Science and the Netherlands Organization for Scientific Research (NWO grant number 024.004.031).

REFERENCES

Ashlock, Daniel. (2006). *Evolutionary Computation for Modeling and Optimization*. New York: Springer) xix, 571.

Bongard, J. C. (2013). Evolutionary Robotics, *Commun. ACM* 56, 74–85.

Badyaev, A. V. (2011). Origin of the Fittest: Link between Emergent Variation and Evolutionary Change as a Critical Question in Evolutionary Biology. *Proc. R. Soc. B.* 278 (1714), 1921–1929. doi:10.1098/rspb.2011.0548

Beauchamp, T. L., and Childress, J. F. (2019). *Principles of Biomedical Ethics*. Eighth edition. New York: Oxford University Press.

Bongard, J. (2011). Morphological Change in Machines Accelerates the Evolution of Robust Behavior. *Proc. Natl. Acad. Sci.* 108 (4), 1234–1239. doi:10.1073/pnas.1015390108

Bostrom, Nick. (2014). *Superintelligence: Paths, Dangers, Strategies*. Oxford: Oxford University Press.

Brey, P. (2017). in *Ethics of Emerging Technology*. *Ethics of Technology*. Editor S.-O. Hansson (London: Rowman & Littlefield International), 175–192.

Brodbeck, L., Hauser, S., and Iida, F. (2015). Morphological Evolution of Physical Robots through Model-free Phenotype Development. *PLoS One* 10 (6), e0128444. doi:10.1371/journal.pone.0128444

Bryson, J. J. (2018). Patiency Is Not a Virtue: the Design of Intelligent Systems and Systems of Ethics. *Ethics Inf. Technol.* 20 (1), 15–26. doi:10.1007/s10676-018-9448-6

Buchanan, E., Le Goff, L. K., Li, W., Hart, E., Eiben, A. E., De Carlo, M., et al. (2020). Bootstrapping Artificial Evolution to Design Robots for Autonomous Fabrication. *Robotics* 9 (4), 106. doi:10.3390/robotics9040106

Čapek, K. (1920). R.U.R.: Rossums Universal Robots (English Translation). Available at: http://www.gutenberg.org/ebooks/59112.

Carpenter, Julie. (2016). *Culture and Human-Robot Interaction in Militarized Spaces: A War story (Emerging Technologies, Ethics and International Affairs*. Burlington, VTcm: Ashgate Publishing Company.

Coeckelbergh, Mark. (2012). *Growing Moral Relations: Critique of Moral Status Ascription (Houndmills, Basingstoke, Hampshire*. New York: Palgrave Macmillan) xvi, 239.

Damiano, L., and Dumouchel, P. (2018). Anthropomorphism in Human-Robot Co-evolution. *Front. Psychol.* 9, 468. doi:10.3389/fpsyg.2018.00468

Danaher, J. (2020). Welcoming Robots into the Moral Circle: A Defence of Ethical Behaviourism. *Sci. Eng. Ethics* 26 (4), 2023–2049. doi:10.1007/s11948-019-00119-x

Darling, Kate. (2021). *The New Breed: What Our History with Animals Reveals about Our Future with Robots* (First edition. edn.; New York, NY: Henry Holt and Company) pages cm.

Darling, K. (2017). "'Who's Johnny?' Anthropological Framing in Human-Robot Interaction, Integration, and Policy," in *Robot Ethics 2.0: From Autonomous Cars to Artificial Intelligence*. Editors P. Lin, K. Abney, and R. Jenkins (Oxford: Oxford University Press), 173–192.

de Jong, K. A. (2006). *Evolutionary Computation: A Unified Approach*. Cambridge, Mass: MIT Press) ix, 256.

Di Nucci, E. (2020). *The Control Paradox: From AI to Populism*. London: Rowman & Littlefield International.

Doncieux, S., Bredeche, N., Mouret, J-B., and Eiben, A. E. (2015). Evolutionary Robotics: what, Why, and where to. *Front. Robotics AI* 2 (4). doi:10.3389/frobt.2015.00004

Eiben, A. E., Bredeche, N., Hoogendoorn, M., Stradner, J., Timmis, J., Tyrrell, A. M., et al. (2013). "The Triangle of Life: Evolving Robots in Real-Time and Real-Space," in *Proc. Of ECAL 2013*. Editor P. Lio (MIT Press), 1056–1063. doi:10.7551/978-0-262-31709-2-ch157

Eiben, A. E., Kernbach, S., and Haasdijk, E. (2012). Embodied Artificial Evolution. *Evol. Intel.* 5 (4), 261–272. doi:10.1007/s12065-012-0071-x

Eiben, A. E., and Smith, J. E. (2003). *Introduction to Evolutionary Computing*. Springer.

Eiben, A. E., and Smith, J. (2015). From Evolutionary Computation to the Evolution of Things. *Nature* 521 (7553), 476–482. doi:10.1038/nature14544

Ellery, A., and Eiben, A. E. (2019). "To Evolve or Not to Evolve? that Is the Question," in ALIFE 2019: Proceedings of the Artificial Life Conference 2019. H. Fellermann, J. Bacardit, Á. Goñi-Moreno, and R. M. Füchslin. Editors, 357–364.

Ellery, A. (2020). How to Build a Biological Machine Using Engineering Materials and Methods. *Biomimetics (Basel)* 5 (3), 35. doi:10.3390/biomimetics5030035

Floreano, D., and Keller, L. (2010). Evolution of Adaptive Behaviour in Robots by Means of Darwinian Selection. *Plos Biol.* 8 (1), e1000292. doi:10.1371/journal.pbio.1000292

Floridi, L., Cowls, J., Beltrametti, M., Chatila, R., Chazerand, P., Dignum, V., et al. (2018). AI4People-An Ethical Framework for a Good AI Society: Opportunities, Risks, Principles, and Recommendations. *Minds & Machines* 28 (4), 689–707. doi:10.1007/s11023-018-9482-5

Futuyma, D. J. (2013). *Evolution (Massachusetts*. USA: Sinauer Associates.

Gellers, J. (2020). *Rights for Robots: Artificial Intelligence, Animal and Environmental Law*. London: Routledge.

Gordon, J. S., and Nyholm, S. (2021). *Ethics of Artificial Intelligence*. (Internet Encyclopedia of Philosophy https://iep.utm.edu/ethic-ai/.

Gunkel, David. J. (2018). *Robot Rights*. Cambridge, Massachusetts: MIT Press) xiv, 237.

Hale, M. F., Buchanan, E., Winfield, A. F., Timmis, J., Hart, E., Eiben, A. E., et al. (2019). MIT Press, 95–102.The ARE robot fabricator: How to (re) produce robots that can evolve in the real world. 2019 Conference on Artificial Life

Hansson, S. O. (2017). in *Ethical Risk Analysis*. *Ethics of Technology*. Editor S.-O. Hansson (London: Rowman & Littlefield International), 157–172.

Howard, D., Eiben, A. E., Kennedy, D. F., Mouret, J.-B., Valencia, P., and Winkler, D. (2019). Evolving Embodied Intelligence from Materials to Machines. *Nat. Mach Intell.* 1 (1), 12–19. doi:10.1038/s42256-018-0009-9

Jelisavcic, M., de Carlo, M., Hupkes, E., Eustratiadis, P., Orlowski, J., Haasdijk, E., et al. (2017). Real-world Evolution of Robot Morphologies: A Proof of Concept. *Artif. Life* 23 (2), 206–235. doi:10.1162/artl_a_00231

Kriegman, S., Blackiston, D., Levin, M., and Bongard, J. (2020). A Scalable Pipeline for Designing Reconfigurable Organisms. *Proc. Natl. Acad. Sci. USA* 117, 1853–1859. doi:10.1073/pnas.1910837117

Lipson, H., and Pollack, J. B. (2000). Automatic Design and Manufacture of Robotic Lifeforms. *Nature* 406 (6799), 974–978. doi:10.1038/35023115

Long, J. (2012). *Darwin's Devices: What Evolving Robots Can Teach Us about the History of Life and the Future of Technology*. New York: Basic Books.

Matthias, A. (2004). The Responsibility gap: Ascribing Responsibility for the Actions of Learning Automata. *Ethics Inf. Technol.* 6 (3), 175–183. doi:10.1007/s10676-004-3422-1

Müller, V. (2020). "Ethics of Robotics and AI," in *Stanford Encyclopedia of Philosophy*. Editor E. N. Zalta. (online) Available at: https://plato.stanford.edu/archives/win2020/entries/ethics-ai/.

Nihlen Fahlquist, J. (2017). in *Responsibility Analysis*. *Ethics of Technology*. Editor S.-O. Hansson (London: Rowman & Littlefield International), 129–142.

Nolfi, S., and Floreano, D. (2000). *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. Cambridge, MA: MIT Press.

Nyholm, Sven. (2020). Humans And Robots: Ethics, agency, and Anthropomorphism London: Rowman & Littlefield international.

Robbins, S. (2019). A Misdirected Principle With a Catch: Explicability for AI. *Minds and Machines* 29, 495–514.

Russell, S., and Norvig, P. (2020). *Artificial Intelligence: A Modern Approach*. 4th edition. Hoboken: Prentice-Hall.

Russell, S. (2019). *Human Compatible: Artificial Intelligence and the Problem of Control*. London: Penguin.

Santoni de Sio, F., and van den Hoven, J. (2018). Meaningful Human Control over Autonomous Systems: A Philosophical Account. *Front. Robot. AI* 5, 15. doi:10.3389/frobt.2018.00015

Sparrow, R. (2007). Killer Robots. *J. Appl. Philos.* 24 (1), 62–77. doi:10.1111/j.1468-5930.2007.00346.x

Thompson, J. N. (1994). *The Coevolutionary Process*. Chicago: University of Chicago Press.

Torresen, J. (2018). A Review of Future and Ethical Perspectives of Robotics and AI. *Front. robotics AI* 4 (75). doi:10.3389/frobt.2017.00075

True, J. R., and Carroll, S. B. (2002). Gene Co-option in Physiological and Morphological Evolution. *Annu. Rev. Cel Dev. Biol.* 18, 53–80. doi:10.1146/annurev.cellbio.18.020402.140619

Turkle, S. (2004). Whither Psychoanalysis in Computer Culture? *Psychoanalytic Psychol.* 21 (1), 16–30. doi:10.1037/0736-9735.21.1.16

van de Poel, I. (2016). An Ethical Framework for Evaluating Experimental Technology. *Sci. Eng. Ethics* 22 (3), 667–686. doi:10.1007/s11948-015-9724-3

Vargas, Patricia. A., Di Paolo, E. A., Harvey, I. M., Husbands, P., and Moioli, R. (2014). The Horizons of Evolutionary Robotics *(Intelligent Robotics and Autonomous Agents*. MIT Press) x, 302.

Vujovic, V., Rosendo, A., Brodbeck, L., and Iida, F. (2017). Evolutionary Developmental Robotics: Improving Morphology and Control of Physical Robots, *Artif. Life*, 23, 169–185. doi:10.1162/artl_a_00228

Waibel, M., Floreano, D., and Keller, L. (2011). A Quantitative Test of Hamilton's Rule for the Evolution of Altruism. *Plos Biol.* 9 (5), e1000615. doi:10.1371/journal.pbio.1000615

Winfield, A. F. T., and Jirotka, M. (2017). "The Case for an Ethical Black Box," in Annual Conference Towards Autonomous Robotic Systems (Springer), 262–273. doi:10.1007/978-3-319-64107-2_21

Winkle, K. (2020). "What Could Possibly Go wrong?"Logging HRI Data for Robot Accident Investigation'," in HRI '20: Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction, 517–519.

Zykov, V., Mytilinaios, E., Adams, B., and Lipson, H. (2005). Self-reproducing Machines. *Nature* 435 (7039), 163–164. doi:10.1038/435163a

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The handling Editor declared a past co-authorship with one of the authors (AE).