Fitness Based Identification of a Robot Structure

Juan Cristobal Zagal[1], Javier Ruiz-del-Solar[2], Adrian Galo Palacios[1]

[1]Centro de Neurociencia de Valparaíso, Universidad de Valparaíso,
and Instituto de Sistemas Complejos de Valparaíso, Chile
[2]Departamento de Ingeniería Eléctrica, Universidad de Chile, Chile
jzagal@ing.uchile.cl

Abstract

Embodiment theory suggests that recurrent processes of sensorimotor activity give rise to cognitive structures. In the case of robots, internal sensorimotor activity generated with physics simulators can be exploited to expand the historical domain of action, however pre-engineered simulations are limited by the reality gap problem. Alternatively simulation might be inferred and self-constructed out of data collected during robot functioning. Fundamental to this line of research is defining a distance function to asses the potential of candidate robot simulations to reproduce real world activity. In this paper we study the characteristics of a distance function based on behavioral fitness measurements. We show how this function can be applied for the generation of behaviors using an algorithm that co-evolves a robot and its simulation. The experiments show how the monotonicity of the function increases with the number of behaviors being tested in reality and with the genotypic diversity of corresponding robot controllers. Moreover it allows for the accurate identification of behavior-relevant parameters contained in the simulation. The metric shows an advantage, when compared to other metrics, for assessing the quality of simulators over long time scales of robot behavioral evaluation.

Introduction

The long-term motivation of this investigation is exploring how real robots can generate more interesting (and practical) behaviors during their ontogeny. The experience (Hornby et al., 1999) shows that a large amount of time is required for obtaining such behaviors as the result of interaction with a real environment. The use of simulation has been explored in order to expand the historical domain of action that a robot undergoes (Ziemke, 2003; Jakobi, 1998). However, according to the reality gap problem, controllers generated in simulation fail to perform similarly once transferred to a real environment. Alternatively we consider that robot simulation must be grounded and self-constructed rather than as a result of a pure engineering design process.

Following this line of thought, we have developed a dynamic reconfigurable robot simulator (Zagal and Ruiz-del-Solar, 2004), together with an algorithm (Zagal et al., 2004; Zagal and Ruiz-del-Solar, 2007) that allows a robot to continuously construct and validate its simulation by co-evolving it with its controller. The method have shown the highest learning performance[1] when compared to other machine learning methods (Genetic Algorithms, Policy Gradient Reinforcement Learning, Evolutionary Hill Climbing With Line Search, Powell Direction Set) that have been applied to the task of gait generation with AIBO robots. It was also successfully applied to the automatic generation of unconstrained ball kick behaviors with AIBO robots (Zagal et al., 2004; Zagal and Ruiz-del-Solar, 2007).

Similarly in (Philipona et al., 2004) the question of whether *there is an algorithm linked to an unknown body that can infer by itself information about the body and the world it is in* was raised. According to experiments with a simulated head they concluded that *sensorimotor laws possess intrinsic properties related to the structure of the physical world in which an organism's body is embedded.* In (Bongard and Lipson, 2004) the *Estimation Exploration* algorithm is proposed as a way to co-evolve a robot and its simulator. They later applied this algorithm to real robots (Bongard et al., 2006). Converging approaches are presented in (Vaughan and Zuluaga, 2006) where self-simulation is proposed for robot planning, and (Ziemke et al., 2005) where internal simulation of robot perception is explored.

Central to this work is defining a distance function to asses the quality of candidate robot simulations. Different functions have been applied; the *rolling mean metric* (Bongard and Lipson, 2004) aims at comparing sensor time series resulting from a target robot and candidate robot simulations. However, according to their proponents *quantitatively comparing sensor data from two highly coupled, highly non linear machines,... is very difficult: slight differences between the two machines rapidly leads to uncorrelated signals.* In (Lungarella et al., 2005) it is proposed that sensorimotor activity can be characterized by looking at their statistical regularities. From this idea in (Mirza et al., 2007)

---

[1]Learning performance defined as $LP = \frac{IF \times d}{e}$, where $IF = \frac{f_{end} - f_0}{f_{max}}$ is the normalized fitness (f) improvement, d is the controller dimensionality and e is the number of evaluations performed in a real robot.

the *experience metric* over a temporal window of sensorimotor observation (statistical distance) was proposed, giving some insights on the relation of the horizon of experience and cycle time of interactions. Such statistically based metrics have the potential of overcoming the limitations of a time dependent comparison.

In this paper we explore a distance function based on the average fitness discrepancies of a set of robot behaviors tested in reality and in candidate robot simulations. The function has already been shown to be useful at reducing fitness discrepancies of real versus simulated robots during co-evolutionary experiments (Zagal et al., 2004). The intention of this paper is to address some questions that remain to be clarified:

1. We wonder if a minimization of this function does necessarily imply a better identification of aspects from reality such as structural robot variables contained in the simulation, or alternatively if it generates bizarre representations that are good just for reproducing behavior.

2. If the first is true we wonder if the function just allows for good approximations or whether it might allow a perfect match of measurable quantities to be achieved.

3. To aid genetic search over a space of candidate simulations the function should be monotonic with respect to the identification error, thus we wonder about how is this monotonicity being affected by the number of behaviors.

4. Another question is about how this monotonicity is affected by the genotypic diversity of these behaviors.

Answering these questions is fundamental towards generating robots with self-modeling capabilities. Since the task requires a reality that can be measurable at the hands of the experimenter we use a simulated reality defined by parameters that are to be uncovered by the methodology under inspection.

The remainder of this article is organized as follows: Definitions are presented in section II. The principle of operation of the function under analysis is presented in section III. Experimental results are presented in section IV. Comparisons with alternative metrics are presented in section V. Conclusions and projection of this work are given in section VI.

Definitions

Simulation: A robot simulation is defined by a vector $s = \{s_1, \ldots, s_{N_s}\}$ in the space S of possible simulations. The dimensions of this space might be defined by morphological aspects such as the length, width, shape or weight of robot components as well as their topological relation. It might also include aspects such as the friction among different elements, gravitational forces, motor parameters such as PID servo constants, etc. The experimenter defines the S boundaries of each parameter $s_i \in [min_i, max_i]$ with

$i = \{1, \ldots, N_s\}$. In the particular case in which reality is defined as a point $s_r \in S$ we will refer to a simulation as any point $s \neq s_r$ in S.

Reality: Reality is the target operational environment of the robot. We present experiments in which reality corresponds to a particular realization of the simulation $s_r \in S$. As it will be described, s_r is unknown for the robot and it should be determined by the algorithm by relying on behavioral comparisons.

Controller: A robot controller is defined by the vector $c = \{c_1, \ldots, c_{N_c}\}$ in a space C of possible robot controllers. The space C might include morphological descriptors of the robot besides controller-related parameters. However, we have not performed experiments for the evolution of robot morphologies. The experimenter defines the C boundaries of each parameter $c_i \in [min_i, max_i]$ with $i = \{1, \ldots, N_c\}$.

Behavior: It is the set of actions that a robot executes in response to the environment E. The characterization and qualification of a robot behavior necessarily depends on the observer. From a single viewpoint we can model behavior in discrete time $t_j = j \cdot \Delta t$ as a time series $B = \{X_0, \ldots, X_{N_b-1}\}$ of N_b vector states $X_j = \{x_1, \ldots, x_{N_d}\}$, each describing N_d dynamical parameters, such as position, rotation and velocity, of bodies composing the robot or interacting with it. If we assume a set of fixed initial conditions, a fixed reference system and fixed evaluation period $T_e = N_b \cdot \Delta t$, we can establish that the robot behavior B is a function of the robot controller c and the evaluation environment E. Thus we have $B = B(E, c)$. In this context E might be either a simulation defined by a point s in S or the reality itself. Clearly in the later case $''E = reality''$ is just an abstraction[2] for the sake of consistency.

Fitness: It is the behavioral evaluation provided by the experimenter. From a set of M robot controllers we note the fitness of robot controller c_k, with $k = \{1, \ldots, M\}$ that elicit behavior $B(E, c_k)$ as f_{Ek} with $E = r$ for reality and $E = s$ for simulation. For example, at the end of T_e it might be the distance traveled by the robot, the distance traveled by a ball that the robot kicks, the amount of consumed energy, etc.

Exploiting Behavioral Consistency

In this section we discuss the problem of how to construct a robot sensorimotor simulation from data collected during robot functioning. If the behavior elicited by robot c in simulation s is similar to the behavior observed in reality we write

---

[2]We intend to approximate relevant aspects of reality, but not to represent it.

$$B(s,c) \approx B(r,c) \qquad (1)$$

more precisely this means that at any time step j we also have

$$X_{sj} \approx X_{rj} \ \forall j = \{0, \ldots, N_b - 1\} \qquad (2)$$

where X_{sj} stands for the state vector of $B(s,c)$ at time step j, and X_{rj} is the state vector of $B(r,c)$ at time step j. In other words there should be a match along time of the robot state in both simulation and reality.

If we somehow measure the degree in which this match is obtained for each simulation s, we can derive a useful distance for adapting simulation to match behavior. Unfortunately the state X_{rj} is generally unobservable[3]. On the other hand, by definition of an Evolutionary Robotics problem the behavioral fitness obtained in simulation f_{sk} and reality f_{rk} are always available. We can thus define in terms of these measurements the behavioral fitness discrepancy elicited by robot c_k as

$$\delta_k = |f_{sk} - f_{rk}| \qquad (3)$$

If behavioral discrepancies, as expressed in (3), are reduced for various behaviors, then it is natural to expect that simulation approximates reality better in those characteristics that are relevant for the execution of these behaviors. We note the average behavioral differences $\Delta_{fitness}$ as:

$$\Delta_{fitness} = \frac{1}{m}\sum_{k=1}^{m} \delta_k = \frac{1}{m}\sum_{k=1}^{m} |f_{sk} - f_{rk}| \qquad (4)$$

Back-to-Reality algorithm

Using this distance we follow the Back-to-Reality (BTR) algorithm steps in order to construct simulations during robot ontogeny. Detailed descriptions of the algorithm are reported in (Zagal et al., 2004) and (Zagal and Ruiz-del-Solar, 2007). At each iteration i the algorithm co-evolves robot controllers with robot simulators by executing the following steps:

Step 1, *robot controller search under simulation:* Using the best known simulation s_{i-1} as environment, genetic search is conducted over the controller solution space C. A starting population of M controller individuals is obtained by performing bit changes with probability p_m over the solution c_{i-1} which is given as a seed. For the first iteration the population can be generated as random or biased by a known starting solution c_0.

The search is steered towards maximizing the fitness $f(B(s_{i-1}, c_i))$. The amount of generations during which

genetic search is conducted in this step should be small if the problem present a high tendency for drift[4].

Step 2, *selection, transfer and test:* A set of ($m < M$) controllers are selected in order of descending fitness and tested in reality. Corresponding fitness values f_{rk}, with $k = \{1 \ldots m\}$ are stored. If it is not possible to find m transferable individuals from the last generation, they will have to be taken, in descending order, from the previous generations obtained in Step 1.

Step 3, *simulation search:* The best existing simulator solution s_{i-1} is used in order to bias a population of L simulator individuals. In the case of the first iteration this population is generated as random or as biased by a known starting simulator solution s_0. A simulation s_i is obtained by steering the evolution towards minimizing $\Delta_{fitness}$.

The algorithm continues by taking the simulation obtained in step 3 as a new environment for step 1 in the next iteration. There is a genotypic similarity among the m controllers that triggers a phenotypic similarity among corresponding behaviors. A probability p_m per bit controls the rate of mutation for a population constructed from a given controller c_i.

Results

Experimental settings

A dynamic simulation of an ant-like robot (hexapod) was implemented using the UCHILSIM simulator (Zagal and Ruiz-del-Solar, 2004). Figure 1 (a) shows the configuration of the 15 rigid robot bodies. The alitrunk, petiole and gaster are represented by bodies b_0, b_1 and b_2. Bodies $\{b_3, \ldots, b_8\}$ correspond to the femur and $\{b_9, \ldots, b_{14}\}$ are the tibia and tarsus of each leg. The joint j_0 connect body b_0 and b_1, similarly the joint j_1 connect body b_1 with b_2. The joints $\{j_2, \ldots, j_7\}$ connect each femur with a corresponding alitrunk having vertical and frontal axis of motion as depicted on the figure. The joints $\{j_8, \ldots, j_{13}\}$ connect each femur and tibia with a frontal axis of motion. Each independent axis of motion i (a total of 18) is motorized and torque is applied according to the output of PID dynamic compensators that follow a motion reference signal $r_i = \theta_i + a_i sin(wt + \phi_i)$, where θ_i is a pre-defined central angle of oscillation for each motor i. Equal uniform mass density is given to all bodies.

The default robot posture (when $r_i = \theta_i \ \forall i$) is presented in Figure 1 (b). The behavior evaluation time T_e is set to 1.7 seconds. A physics integration step takes $\Delta_t = 5 \times 10^{-4}$ seconds, therefore the state vector $X_{\{s/r\}j}$ is computed $N_b = 3400$ times along a behavior evaluation. The frequency is selected to be $w = 60Hz$. The amplitude a_i and motion phase ϕ_i are defined by the robot controller vector

---

[3]In control theory a system is observable if, for any possible sequence of state and control vectors, the current state can be determined in finite time using only the outputs.

[4]A pathology of co-evolving systems in which the selection pressure of one population has no influence in the co-evolving population.

(a)
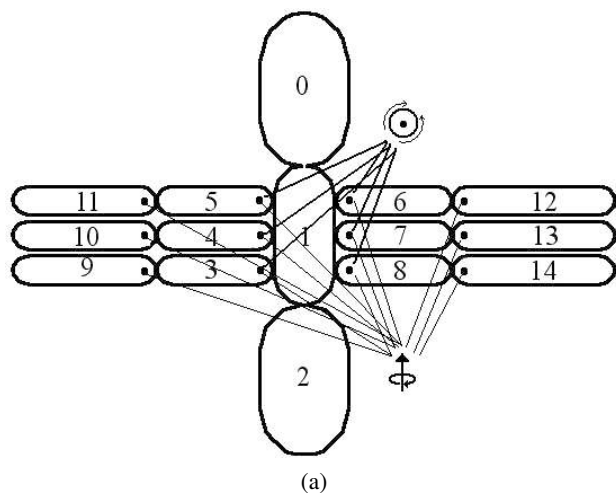
(b)
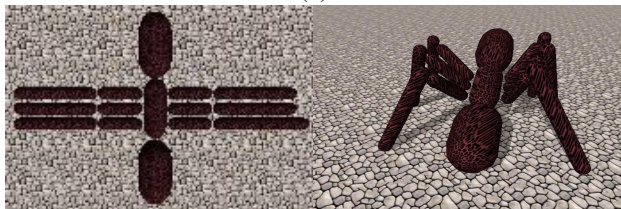
(c)   (d)

(e)   (f)

Figure 1: In (a) configuration of the 15 rigid bodies of the robot; motorized joints are connecting bodies $\{3,\ldots,8\}$ with body 1 in two orthogonal axis of motion. Bodies $\{9,\ldots,14\}$ are connected to the previous set constrained to one axis of motion. In (b) the robot under its default posture. In (c,d) reality defined such that body 14 length is $bl_{max} = 1.6$. In (e,f) example of other realization of the simulator for which body 14 length is $bl_{min} = 0.4$. Remaining body lengths are $\{bl_3,\ldots,bl_8\} = 0.64$ and $\{bl_9,\ldots,bl_{13}\} = 1.36$ in both cases.

under search, $c = \{a_1, \phi_1, \ldots, a_{18}, \phi_{18}\}$ of 36 elements. These parameters are in the range $a_i \in \{a_{min}, a_{max}\}$ and $\phi_i \in \{\phi_{min}, \phi_{max}\}$ for each joint. The robot simulator is defined as a vector s representing the actual lengths of the ant limbs $s = \{bl_3, \ldots, bl_{14}\}$ of 12 elements in the range $bl_i \in \{bl_{min}, bl_{max}\}$.

We have selected the following parameter range: $a_{min} = 0.12989$, $a_{max} = 0.314$, $\phi_{min} = 0$, $\phi_{max} = 3.1416$, $bl_{min} = 0.4$, $bl_{max} = 1.6$. It is important to notice that the selected parametrization might bring about radically different behaviors since no symmetry simplifications have been made for the leg motion pattern and the parameter range is sufficiently large to produce a variety of different behaviors (falling upside-down, walking in circles, backwards, sideways, forward, jumping, etc).

Fitness discrepancy and morphology

As result of applying BTR step 1 using a starting simulator s_0 and and robot controller c_0 we obtained a population of $m = 15$ robot controllers. Corresponding fitness was measured in reality according to BTR step 2, stored and ranked. Figure 2 (a) shows corresponding evolution of controller fitness. The best transferable controller c_1 achieved $11 m/s$ in simulation.

Using this population of controllers we scanned $\Delta_{fitness}$ as function of morphological variations (of simulated versus real robot). In this case simulation is defined by a vector s that differs only in one parameter with respect to reality s_r, and then we evaluate $\Delta_{fitness}$ while varying this parameter along its whole range. As a first test, let us define reality s_r in S such that the length of body 14 takes the value $bl_{14} = bl_{max} = 1.6$ while the remaining body lengths are set to $\{bl_3, \ldots, bl_8\} = 0.64$ and $\{bl_9, \ldots, bl_{13}\} = 1.36$ as shown in Figure 1 (c,d). In order to illustrate the range of search a point s_0 different from reality is shown in Figure 1 (e,f) such that $bl_{14} = bl_{min} = 0.4$.

Figure 2 (b) shows results of scanning $\Delta_{fitness}$ along the complete parameter range of bl_{14}. The partial components $\delta_k = |f_{rk} - f_{sk}|$ used for computing $\Delta_{fitness}$ are presented for all behaviors $k = \{1, \ldots, 15\}$ as functions of bl_{14}. Figure 2 (c) presents results from a second test in which the length of body 5, bl_5, is chosen as the varying parameter. In this case reality is such that the length of this body is $bl_5 = 0.88$, while the remaining body lengths are $\{bl_3, \ldots, bl_4\} = 0.64$, $\{bl_6, \ldots, bl_8\} = 0.64$ and $\{bl_9, \ldots, bl_{14}\} = 1.36$. Similarly as before $m = 15$ individuals are selected from a population of robot controllers.

In order to understand the influence of m on the monotonicity of $\Delta_{fitness}$ we generated $M = 45$ controllers by modifying c_1 genotype bits with a probability $p_m = 0.02$ per bit. Figure 3 (a) shows scans of δ_k computed separately around body $bl_5 = 0.88$, and in (b) scans of $\Delta_{fitness}$ computed as a function of different amounts m of behaviors. The same figure shows in two subplots corresponding fitness val-
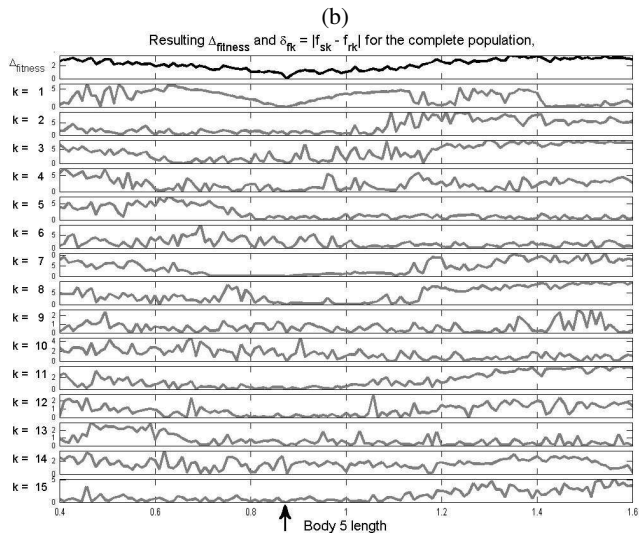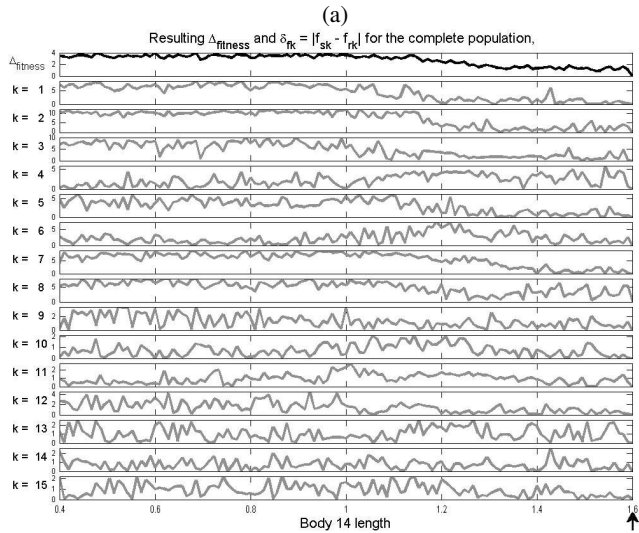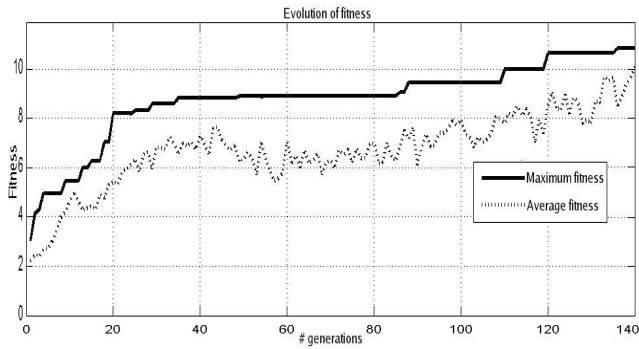
(a)

(b)

(c)

Figure 2: In (a) evolution of controller fitness. In (b,c): $\Delta_{fitness}$ together with its $m = 15$ behavioral components $\delta_k = |f_{rk} - f_{sk}|$ resulting from scanning body 14 (b) and body 5 (c) lengths along the full parameter stroke $bl_{14,5} = [0.4, \ldots, 1.6]$. It is particularly interesting to observe the decreasing monotonic behavior of $\Delta_{fitness}$ when approximating the real value of 1.6 (a) and 0.88 (b). Further observations are described in the main text.

ues resulting from each one of the 45 behaviors when measured in reality (left) and for the starting simulation (right). Figure 3 (c) shows 16 $\Delta_{fitness}$ curves computed for a fixed number of behaviors (15) but increasing the diversity parameter p_m with the darkness of curves.

Finally, following BTR step 3, $\Delta_{fitness}$ is used to identify a hidden simulation point defined by $bl_5 = 0.54$, $bl_6 = 0.74$, $bl_3 = 0.54$, $bl_8 = 0.74$. Figure 4 (a) shows the minimization of $\Delta_{fitness}$ along 100 generations. In (b) the resulting parametric convergence is shown. We have made the following observations from these experiments:

1. A first observation is that when simulation equals reality, this is when the varying parameters match corresponding real values ($bl_{14} = bl_{max} = 1.6$ in the first test and $bl_5 = 0.88$ in the second test) with the remaining parameters left equal, we verify for the $k = \{1, \ldots, m\}$ partial behavioral discrepancies that $\delta_k = 0 \ \forall k$. This is an experimental support for the following theorem:

   Theorem 1 $s = s_r \Rightarrow \delta_k = 0, \ \forall k \, / \, k = \{1, \ldots, m\}$.

   This theorem is trivial since we have assumed a set of conditions in order to guarantee that if $s = s_r$ we can reproduce behaviors and thus obtain the same fitness values.

2. Similarly we verify that when simulation equals reality $\Delta_{fitness} = 0$. This result is an experimental support of the following theorem:

   Theorem 2 $s = s_r \Rightarrow \Delta_{fitness} = 0$.

   This theorem is also trivial (but useful as well), from previous theorem and equation (4) we have

$$\Delta_{fitness} = \frac{1}{m}\sum_{k=1}^{m}\delta_k = \frac{1}{m}\sum_{k=1}^{m} 0 = 0 \qquad (5)$$

3. We also observe that $\Delta_{fitness} \geq 0 \ \forall s \, / \, s \neq s_r$ and that $\Delta_{fitness} = 0 \ \Leftrightarrow s = s_r$ for the observed range of s. This suggests the following two hypotheses:

   Hypothesis 1 $\Delta_{fitness} = 0 \Rightarrow$ *a subset of parameters in S that are relevant for the execution of m behaviors has been correctly identified by s from s_r.*

   Hypothesis 2 $\Delta_{fitness} = 0 \Rightarrow s = s_r$ *if and only if S contains only parameters which are relevant for the execution of m behaviors, with $m \to \infty$.*

4. Even though we observe that $\Delta_{fitness} = 0 \Leftrightarrow s = s_r$, the partial behavioral discrepancies δ_k might become zero at points such that $s \neq s_r$. In fact there is a likelihood that the fitness of a particular behavior is the same in reality
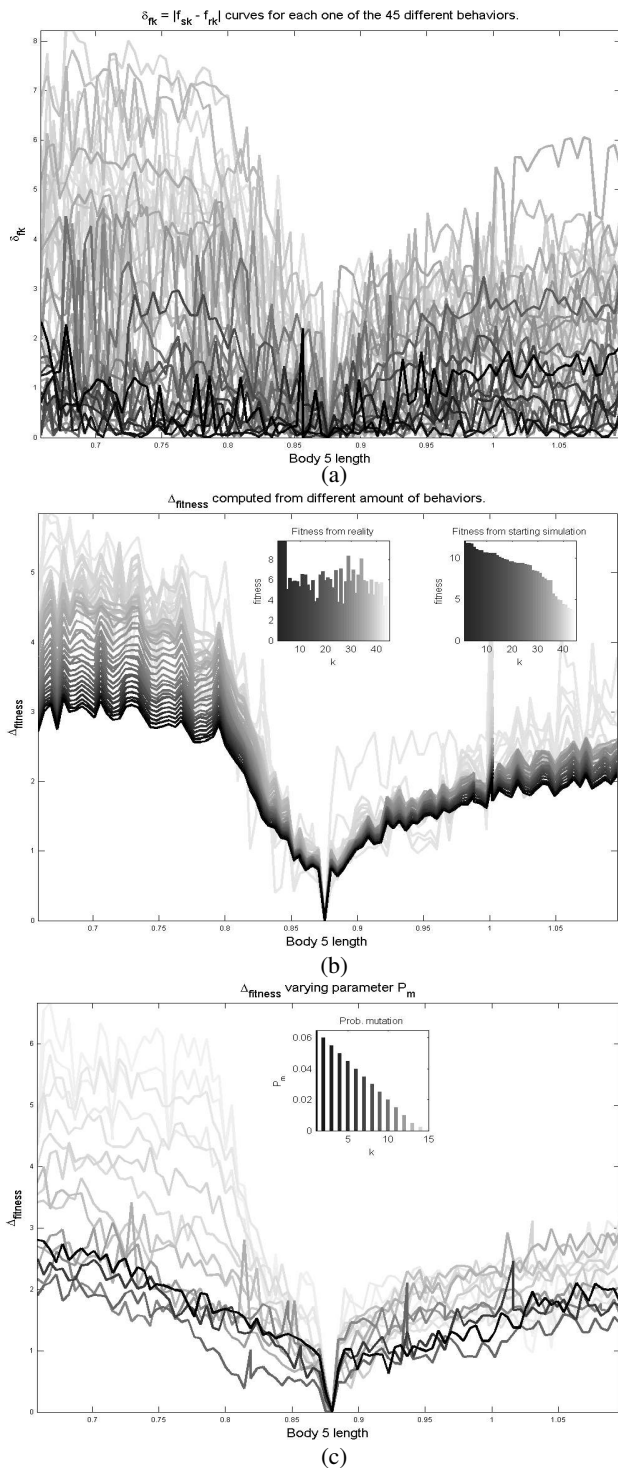
$\delta_{fk} = |f_{sk} - f_{rk}|$ curves for each one of the 45 different behaviors.

(a)

$\Delta_{fitness}$ computed from different amount of behaviors.

(b)

$\Delta_{fitness}$ varying parameter P_m

(c)

Figure 3: In (a) 45 δ_k curves computed for variations of bl_5 around $s = s_r$. In (b) 45 $\Delta_{fitness}$ curves computed for different number of behaviors. Darkness increases with the number of behaviors, the subplots shows fitness values for each one of the 45 controllers in reality (left) and the starting simulation (right). In (c) 16 $\Delta_{fitness}$ curves computed for a fixed number of behaviors (15) but increasing the parameter p_m which is depicted in the sub-plot. Curve darkness increases with P_m.

and simulation even for $s \neq s_r$, but the key point is that it is extremely unlikely that the coincidence happens at the same point s with another behavior. This is a great illustration of the reason several behavioral comparisons are averaged in order to achieve useful measurement.

5. It is possible to recognize a plateau on $\Delta_{fitness}$ for values $bl_{14} \leq 1.1$ in Figure 2 (b). This can be clearly understood when looking at Figure 1 (c,d), what happens is that for such values the corresponding leg does not actually touch the ground and therefore, as can be observed, it does not have an impact in behavior until it reaches values above 1.1. A similar observation can be made in Figure 2 (c), when $bl_5 \geq 1.3$ the whole extremity is lifted away from the ground and therefore the parameter changes beyond that value are not affecting behavior.

6. We can observe as well that $\Delta_{fitness}$ monotonically increases with the identification error, i.e. when s moves away from the real value s_r. The Figure 2 (a,b) shows a strong increasing monotonic behavior of $\Delta_{fitness}$ with the superposition of small fluctuations. On the other hand, the characteristic of a single behavioral discrepancy δ_k is not monotonic with the identification error. From Figure 3 (b) we observe how the smoothness of $\Delta_{fitness}$ increases with m. From these observations we make the following hypothesis

Hypothesis 3 *The smoothness of $\Delta_{fitness}$ as the function of a parameter of S increases with m over the range of behavioral influence of the parameter.*

7. We observe from Figure 3 (c) an increase of the smoothness and linearity of $\Delta_{fitness}$ when increasing the diversity factor p_m. Thus we make another hypothesis

Hypothesis 4 *The smoothness of $\Delta_{fitness}$ as the function of a parameter of S increases with p_m over the range of behavioral influence of the parameter.*

Comparisons With Other Metrics

Results from applying sensor based metrics such as the rolling mean metric (Bongard and Lipson, 2004) and the Euclidean difference of sensor time series of real versus candidate simulations over the described range of bl_5 are presented in figure 5. A central parameter of the rolling mean metric is the header length h which indicates how much of the starting sensor time series are going to be compared. Corresponding metric is computed for different values of this parameter, ranging from 1.25% up to 100% of the total evaluation time. As can be seen, comparing a very short starting period (1.25%) leads to an almost perfectly linear behavior of the metric, however when increasing the length of sensor data under comparison the monotonicity of the
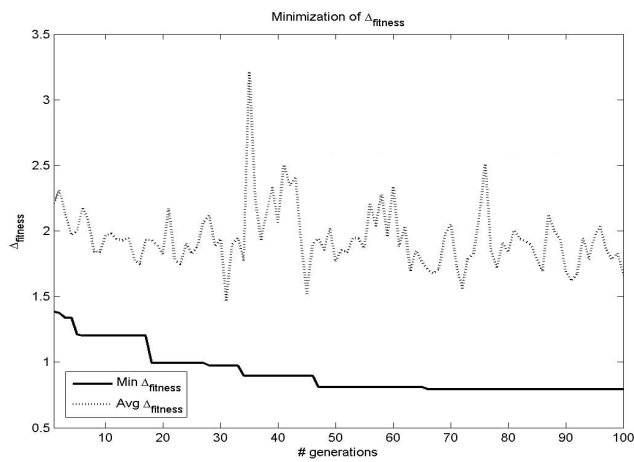
(a)
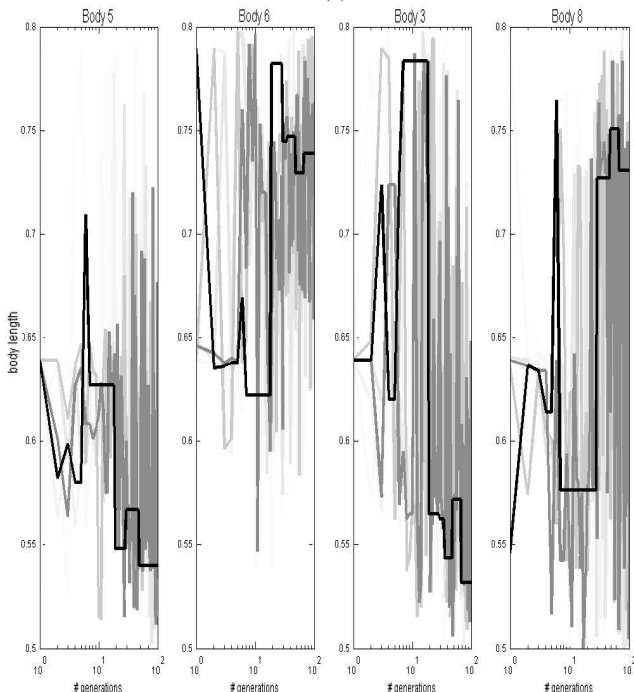
(b)

Figure 4: In (a) minimization of $\Delta_{fitness}$ carried along 100 generations. In (b) corresponding convergence to parameters defining reality is shown. Black lines represent parameters encoded by the individuals having minimum $\Delta_{fitness}$. Similarly the remaining individuals are represented by gray lines, where the darkness of curves is proportional to $\Delta_{fitness}$.

curve decreases reaching a non monotonic behavior. The metric behaves similarly as the Euclidean difference of corresponding time series when h is equal to the whole evaluation period; this is depicted with dashed line. Such behavior can be clearly understood since the decorrelation of sensor time series increases with time for slight differences of simulation and reality.
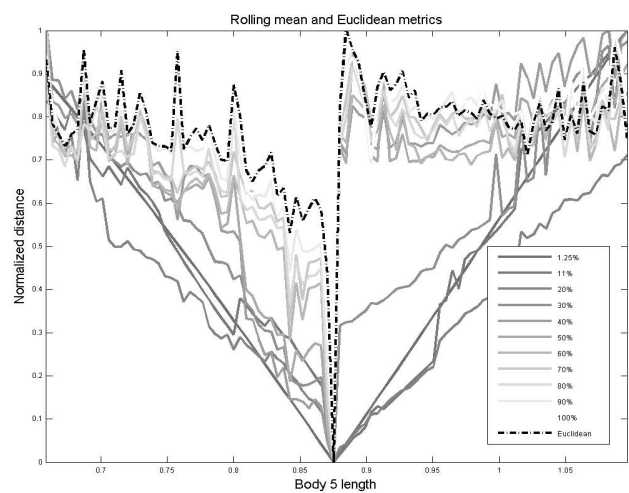

Figure 5: Rolling mean and Euclidean metrics. The rolling mean metric is computed for different values of the starting time header h, ranging from 1.25% up to 100% of the total evaluation time. In dashed line the Euclidean metric is presented considering the whole evaluation time. The figure illustrates how the monotonicity of the rolling mean metric is affected by the parameter h, behaving similar to the Euclidean metric when the whole time window is considered.

Conclusions

We have investigated the behavior of a distance function that can be used for comparing candidate simulators by measuring the average fitness of small variations of behavior. Before drawing conclusions we should remember that in the presented experiments *reality* was also simulated, having the same physical laws as the candidate simulations, but varying morphological aspects of the robots. In principle a negative effect of fitness measurement noise should be reduced with m given the linear construction of $\Delta_{fitness}$. However, performing several behavioral evaluations in a real environment is expensive.

Having this into account we can give the following answers to the main questions that motivated this work:

1. Indeed a minimization of $\Delta_{fitness}$ necessarily implies a better identification of aspects of reality. However these aspects must be related to the execution of behaviors considered under $\Delta_{fitness}$. Thus the methodology under analysis allows a robot to generate a self model of the behavior-relevant components of its interaction with the world. However the methodology would leave undetermined the value of parameters that are not relevant for the execution of behavior (like the color of the head!).

2. We have observed that the function allows a perfect match of those parameters that are relevant for the execution of behaviors to be achieved.

3. The monotonicity of $\Delta_{fitness}$ increases with the number m of behaviors that are tested in reality.

4. The monotonicity of $\Delta_{fitness}$ increases with the diversity of the controllers and it can be controlled with the parameter p_m.

We conclude that the latter two factors should be carefully considered when designing experiments on a fitness based identification of robot structures since the monotonicity of the distance function is a critical factor towards increasing the dimensionality and complexity of simulation search spaces.

Out from these experiments we observe that there is an advantage of using a sensor based time series metric (such as the rolling mean metric) when comparing small time portions of data collected during robot functioning. Moreover, since this apporach involves the collection of a higher amount of data (sensor time series versus fitness), it appears as the right move to experiments in reality given the reduction in hardware trials. However, when time increases, the sensor time series become highly decorrelated if simulation is dissimilar to reality. A fitness based comparison such as $\Delta_{fitness}$ allows us to assess the quality of candidate robot simulation over extended evaluation periods.

Simulation should allow us to reproduce real robot operation over all behavior-relevant time scales. We consider that comparisons cannot be restricted to the first instants of robot operation but must be extended until the outcome of behavior is obtained. An account of multi time scale behavioral comparisons is required.

References

Bongard, J. and Lipson, H. (2004). Once more unto the breach: Co-evolving a robot and its simulator. In *Proc. of the Ninth Int. Conf. on the Simulation and Synthesis of Living Systems (ALIFE9)*, pages 57–62.

Bongard, J., Zykov, V., and Lipson, H. (2006). Resilient Machines Through Continuous Self-Modeling. *Science*, 314(5802):1118–1121.

Hornby, G., Fujita, M., Takamura, S., Yamamoto, T., and Hanagata, O. (1999). Autonomous evolution of gaits with the Sony quadruped robot. *Proc. of the Genetic and Evolutionary Computation Conf.*, 2:1297–1304.

Jakobi, N. (1998). *Minimal Simulations for Evolutionary Robotics*. PhD thesis, University of Sussex.

Lungarella, M., Pegors, T., Bulwinkle, D., and Sporns, O. (2005). Methods for quantifying the informational structure of sensory and motor data. *Neuroinformatics*, 3(3):243–262.

Mirza, N., Nehaniv, C., Dautenhahn, K., and te Boekhorst, R. (2007). Grounded Sensorimotor Interaction Histories in an Information Theoretic Metric Space for Robot Ontogeny. *Adaptive Behavior*, 15(2):167.

Philipona, D., O'Regan, J., Nadal, J., and Coenen, O. (2004). Perception of the structure of the physical world using unknown multimodal sensors and effectors. *Advances in Neural Information Processing Systems*, 16:945–952.

Vaughan, R. T. and Zuluaga, M. (2006). Use your illusion: Sensorimotor self-simulation allows complex agents to plan with incomplete self-knowledge. In *From Animals to Animats IX, Proc. of 9th Int. Conf. on Simulation of Adaptive Behaviour (SAB06)*, Rome, Italy.

Zagal, J. C. and Ruiz-del-Solar, J. (2004). UCHILSIM: A dynamically and visually realistic simulator for the robocup four legged league. In *RoboCup 2004: Robot Soccer World Cup VII*, volume 3276 of *Lecture Notes in Computer Science*, pages 34–45. Springer.

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 *Proc. 5th IFAC Symp. on Intelligent Autonomous Vehicles (IAV '04)*, Lisbon, Portugal. Elsevier Science Publishers B.V.

Ziemke, T. (2003). On the role of robot simulations in embodied cognitive science. *Journal of Artificial Intelligence and the Simulation of Behavior*, 1(4):389–399.

Ziemke, T., Jirenhed, D., and Hesslow, G. (2005). Internal simulation of perception: a minimal neuro-robotic model. *Neurocomputing*, 68:85–104.