Morphological Evolution for Pipe Inspection Using ROS

Ahmed Hallawa[1], Touhidur Rahman[1], Giovanni Iacca[2] and Gerd Ascheid[1]

[1] RWTH Aachen University, Kopernikusstraße 16, 52074 Aachen, Germany
[2] University of Trento, Via Sommarive 9, 38123 Trento, Italy
hallawa@ice.rwth-aachen.de, rahman@rwth-aachen.de, giovanni.iacca@unitn.it, ascheid@ice.rwth-aachen.de

Keywords: Morphological Evolution, Evolutionary Computation, Pipe Inspection

1. Overview

In many practical applications such as water/oil/gas distribution systems (e.g. those used in metallurgy plants), continuous monitoring of the status of the pipes is needed to prevent leaks and other kinds of malfunctioning of the system. However, pipe systems are typically difficult to access, and inspecting them might require a complete interruption of the operations of the pipes. An alternative solution is to use mm-sized passive sensor agents, which are able to inspect the pipes *while they are still operating*. However, these agents are typically kinetically passive due to their size. Consequently, controlling the agent behaviour in a particular environment is challenging due to the complexity of the mathematical representation of the problem at hand (e.g. agent-environment interaction). In this regard, black-box optimization techniques such as Evolutionary Algorithms (EAs) suite the problem. In this work, we present a methodology to evolve the morphological properties of an agent for a given behavioural objective in a pipe scenario. As a proof-of-concept, we set the motion behaviour objective to moving across a given pipe as fast as possible, for different insertion points in the pipe. As for the evolvable morphology aspect, we assume that the agent has eight compartments that can be filled with weights. Thus, our contribution is threefold: 1) Designing and developing an evolutionary scheme for evolving the agent's morphology to move across a pipe as fast as possible. This includes genotype-phenotype representation, reproduction operators, selection criteria and fitness function. 2) Developing a simulation framework for evaluating the proposed solution, using Robot Operating System (ROS), and 3) Testing the whole evolution scheme, including simulation, on two different pipe configurations. This proof-of-concept work is designed to take part in the evaluation process of the Phoenix project, which aims to explore unknown environments via evolvable sensor agents [1]. However, we envision this approach to be generally helpful in water/oil/gas distribution systems, where realistic objectives other than speed must be set.

2. Methodology

2.1. Genotype-phenotype representation

The representation of the genotype is an important factor in the evolutionary process. Here, we use a straightforward binary encoding to represent the morphology of the agent. In particular, we assume that an agent is a ball-shaped structure that contains up to eight compartments, which can be either filled with weight or not. The presence of each of these weights in the compartments is obtained by the EA. Therefore, the genotype representation describes the mass distribution inside the agent.

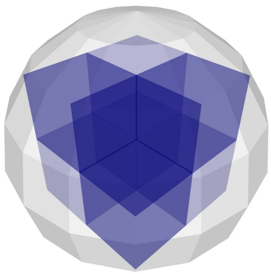

Figure 1: A graphical representation of the agent composed by multiple compartments

For representing the genotype, binary string 0 and 1 are used to show the presence of each of the compartments. So, an agent will have a genotype of 8-bit string, where each bit corresponds to the presence of a weight in a compartment. **Figure 1** shows the structure of the agent defined by a genotype of "11111111". The top four compartments are represented with the first 4 bits, and bottom half is represented with the next 4 bits. So, if the genotype is: "00101110" then the agent will have one filled compartment on the top and 3 compartments on the bottom. Every compartment is similar in structure, and has the same mass.

2.2. Simulation details

To test our ROS-based simulator, we carried out two pipe experiments. In the two simulation scenarios, agents were evolved in order to optimize their ability to travel a certain distance, as fast as possible, in a fixed amount of time (fixed simulation duration). It should be noted that the simulated agents have no actuators and cannot apply any kind of thrust to control their motion, that is only determined by the fluid in which the agents are placed. In both pipe scenarios, we considered a pipe with a certain level of flowing water inside. The difference between the two environments consists in having two different levels of water in each case. The developed simulation loop, which is graphically represented in **Figure 2**, consists of three main phases: 1) initializing the simulation environment with Gazebo; 2) setting up the agents in this environment along with all the computational ROS nodes related to each agent and UUV simulator [2]; and 3) running the simulation to measure the fitness of the agents. The details of the environments, agents parameters, and the definition of the fitness function are described as follows:

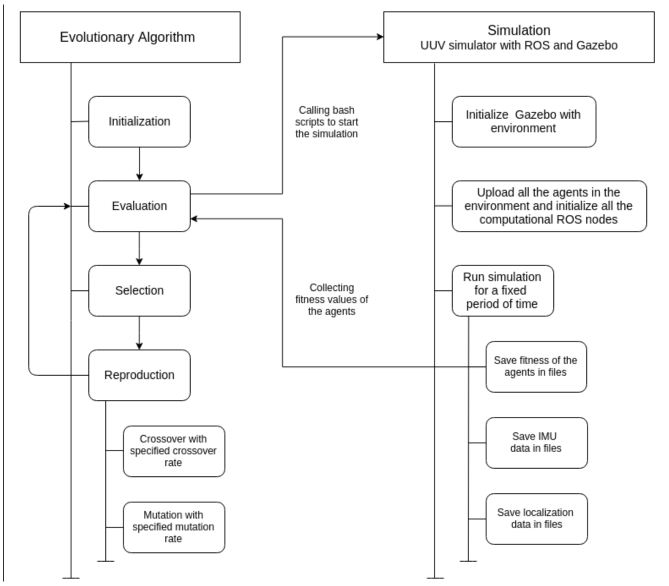

Figure 2: A scheme of the proposed EA-based simulation framework

Environmental setup: In the two pipe scenarios, the water level is 95% and 50%, i.e. filling almost entirely the pipe, or filling only half of it. At the beginning of the simulation, the agents are randomly placed at a different water height, with an offset of 0.5 meters w.r.t. the pipe entrance. The pipe is 10 meters long and has a radius of 1 meter. The water velocity is 1 ms^{-1}, its density is 1024 kgm^{-3}, and the simulation duration is set fixed to 10 seconds.

Agent parameters: The spherical structure of the agent has a radius of 0.05 meter and the each mass compartment is a cube with size 0.028 meters.. The mass of the spherical structure is 0.3 kilogram, and the mass of the compartment is 0.02 kilogram.

Fitness function: The fitness is the average velocity of an agent to travel through the pipe. The agent's goal is to reach the end of the pipe. As the agent is placed 0.5 meters away from pipe entrance and the length of the pipe is 10 meters, the agent is required to travel at least 9.5 meters to achieve the goal. So, the minimum required average velocity is 0.95 ms^{-1} in 10 seconds (simulation duration) to reach the goal. This sets an upper bound to the maximum fitness value. The agent's mass and density is then evolved in each generation to maximize this fitness, so to reach the highest possible speed.

3. Experimental results

Numerical results are shown in **Figure 3**, for both scenarios. The experiments show relative fast convergence towards the maximum possible velocity. It is worth mentioning that each experiment is repeated ten times due to the stochastic nature of EA.

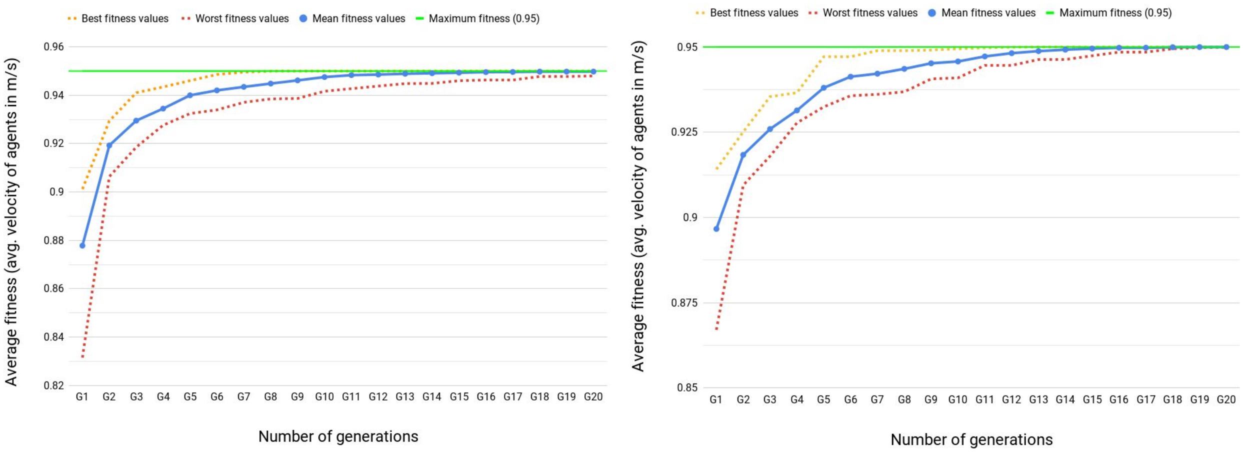

Figure 3: Numerical results in the two scenarios: fully-filled pipe (left) and half-filled pipe (right)

References

1. Hallawa A, Schlupkothen S, Iacca G, Ascheid G. Energy-efficient environment mapping via evolutionary algorithm optimized multi-agent localization. In: Proceedings of the Genetic and Evolutionary Computation Conference Companion 2017 Jul 15 (pp. 1721-1726). ACM.
2. Manhães MM, Scherer SA, Voss M, Douat LR, Rauschenbach T. UUV simulator: A gazebo-based package for underwater intervention and multi-robot simulation. In: OCEANS 2016 MTS/IEEE Monterey 2016 Sep 19 (pp. 1-8). IEEE.