

Evolution of Station Keeping as a Response to Flows in an Aquatic Robot

Jared M. Moore
Department of Computer
Science and Engineering
Michigan State University
East Lansing, MI, USA
moore112@msu.edu

Anthony J. Clark
Department of Computer
Science and Engineering
Michigan State University
East Lansing, MI, USA
ajc@msu.edu

Philip K. McKinley
Department of Computer
Science and Engineering
Michigan State University
East Lansing, MI, USA
mckinley@cse.msu.edu

ABSTRACT

Developing complex behaviors for aquatic robots is a difficult engineering challenge due to the uncertainty of an underwater environment. Neuroevolution provides one method of dealing with this type of problem. Artificial neural networks discern different conditions by mapping sensory input to responses, and evolutionary computation provides a training algorithm suitable to the high dimensionality of the problem. In this paper, we present results of applying neuroevolution to an aquatic robot tasked with *station keeping*, that is, maintaining a given position despite surrounding water flow. The virtual device exposed to evolution is modeled after a physical counterpart that has been fabricated with a 3D printer and tested in physical environments. Evolved behaviors exhibit a variety of unexpected, complex fin/flipper movements that enable the robot to achieve and maintain station, despite water flow from different directions. Moreover, the results show that evolved controllers are able to effectively carry out this task using only information from a simulated accelerometer and gyroscope, matching the inertial measurement unit (IMU) on the actual robot.

Categories and Subject Descriptors

I.2.9 [Computing Methodologies]: Artificial Intelligence—*Robotics, Autonomous vehicles*

Keywords

Evolutionary Robotics, Neural Network, Application, Simulation, Station Keeping, Aquatic Robotics, Neuroevolution

1. INTRODUCTION

Increasingly, mobile robots with embedded microprocessors and electronic control systems aid humans in a variety of tasks. While many applications currently rely on remote-controlled units, increasing levels of autonomy are expected to produce systems that can assist in a growing list of difficult and dangerous applications. One domain in which autonomy is particularly important is aquatic sensing, where human oversight is often limited, if not impossible.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '13, July 6–10, 2013, Amsterdam, The Netherlands.
Copyright 2013 ACM 978-1-4503-1963-8/13/07 ...\$15.00.

Mobile aquatic sensors are likely to play a critical role in ecosystem management, tracking of hazardous wastes, and surveillance of harbors and coastal waterways. While some of these applications can employ propeller-based robots, in others fin-based locomotion potentially offers better maneuverability, less noise and less disruption of the environment. In such devices, often termed *robotic fish* [22], fin movements are typically achieved with either small motors [4, 6, 21, 23] or deformation of electroactive polymers [3]. For instance, Chen et al. [3] demonstrated that a carangiform (that is, propulsion primarily generated by a caudal fin) robotic fish can successfully navigate the surface of water with a single actuator. Other studies of biomimetic aquatic robots have yielded insight into the dynamics of fish locomotion [1, 12, 14] and collective behaviors [8, 16].

Despite these advances, aquatic robots still do not approach their biological counterparts in terms of maneuverability or autonomy; the materials, sensors and actuators that make up a robotic fish simply are not as effective as organic tissue. Unlike meter-sized autonomous underwater vehicles (AUVs) that can house sophisticated hardware for sensing, actuation and data processing, robotic fish are usually required to be small (8-30 cm in length) and relatively inexpensive. To that end, they are typically equipped with low-precision sensors for navigation (accelerometers, gyroscopes, GPS, and digital compasses) and relatively small batteries, making energy management a critical issue. Yet, these robots are required to negotiate aquatic environments characterized by uncertainties resulting from waves, currents and turbulence, as well as plant growth and other obstacles. While mathematical models of the hydrodynamic interactions help to evaluate structures and mechanisms prior to development, the design process remains a challenge due to the large number of parameters involved in producing effective locomotion under different conditions. Each combination of different materials and electromechanical constraints produces different performance and requires detailed knowledge of material properties. These factors directly affect not only low-level control, but also higher-level decision making on how to maneuver the robot to carry out complex tasks.

Evolutionary computation methods are well suited to such high-dimensional problems. By broadly sampling the solution space, evolutionary algorithms are able to test for and blend the beneficial aspects of individual solutions in order to produce effective results. In particular, neuroevolution has emerged as an important tool in developing controllers for systems that must cope with uncertainty and nonlinear dynamics [9, 11]. Evolved neural controllers have been shown to be capable of such tasks, including bipedal walking [15], quadruped gait control [5], navigation [2], and control of a finless rocket [10]. Typically, a simulation environment is uti-

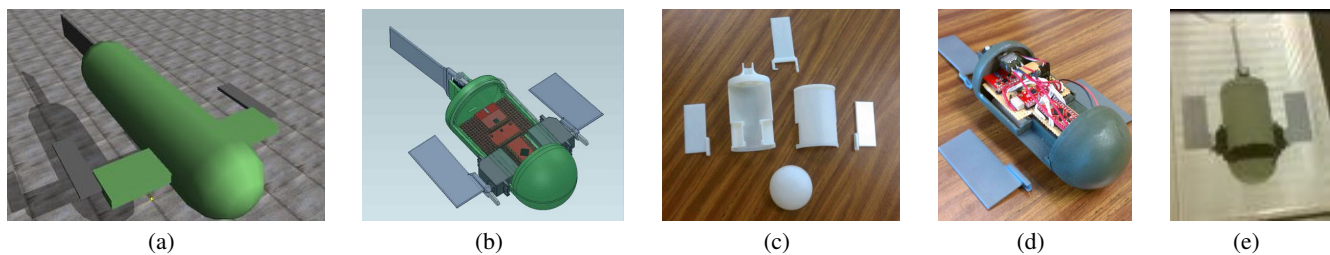


Figure 1: Modeling and fabrication of an aquatic robot. From left to right: (a) evolutionary experiment based on a simulation model; (b) corresponding SolidWorks model for prototype; (c) 3D-printed passive components of prototype; (d) integration of electronic components and battery into the prototype; (e) assembled, painted and waterproofed prototype in the flow tank. The physical prototype’s main body is 13cm long and 8cm in diameter with fins that are 8cm long and 2cm wide.

lized to accelerate the design process. Even a coarse approximation of the target environment can give insight into what constitutes a successful control strategy, although higher fidelity simulation can provide better results [13]. Moreover, by focusing on finding solutions that are *good enough*, as opposed to the global optimum, evolutionary computation is able to handle situations, such as highly dynamic aquatic environments, where exhaustive search of the controller space is infeasible.

Our research integrates evolutionary computation, efficient models of physical materials, and rapid prototyping in order to explore novel designs for robotic fish and other types of robots; see Figure 1. We conduct these studies on a testbed for evolutionary robotics research, termed Evolution Park. The testbed includes rack-mounted computer clusters and a collection of high-end graphics workstations enable evolution of both controllers and morphologies, along with high-precision interactive simulations of the resulting robots. A multi-material 3D printer enables rapid prototyping of robot models produced through computational evolution. The printer is capable of simultaneously jetting materials with different properties in a single build, enabling direct fabrication of complex structures, such as fins that comprise semi-rigid struts and more pliable inner membranes, as well as printing of molds for intricate metal parts. The printed forms are then coupled with electronic control systems, motors, and sensors to produce fully functional robots. To evaluate robots, experiments are conducted in a 4500-gallon tank with underwater and ceiling-mounted video cameras, or an elliptical flow tank in which the robots are exposed to currents and turbulence; see Figure 1. Outdoor experiments are conducted on ponds and lakes near the Michigan State University campus.

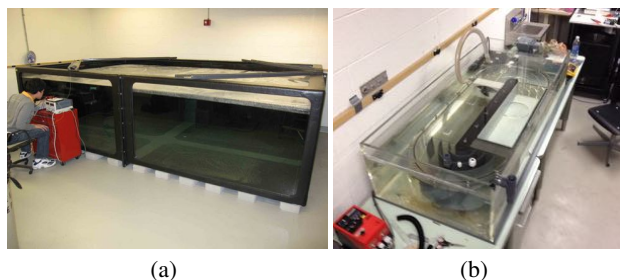


Figure 2: Components of the Evolution Park experimental environment: (a) 4500-gallon custom-built tank for robotic fish experiments; (b) elliptical flow tank for studying behaviors in the presence of water currents.

In this paper, we describe a study on the evolution of controllers for *station keeping*, whereby an aquatic robot is required to maintain a specified position despite surrounding water flow. A behavior exhibited by many species of fish, station keeping is important to robotic tasks such as identification of stationary objects and collection of water quality data at a specified location. Station keeping in aquatic robots is similar to hovering in flying insects, which has been studied as a component of morphological evolution in [18]. Here, we use the NEAT algorithm [20] to evolve controllers for the aquatic robotic platform, shown in Figure 1, that includes two actuated lateral “flippers,” an actuated caudal fin, and an inertial measurement unit (IMU). In this study, we address station keeping in the presence of external forces produced by laminar water flow. As opposed to a turbulent flow, which is characterized by eddies, a *laminar* flow occurs when the water flows at a constant rate in parallel layers, with no mixing between layers. To achieve station, a robotic controller must coordinate the actuation of all motors in an effort to locomote against external forces by interpreting inertial (i.e., linear and angular acceleration) data. To the best of our knowledge, aquatic station keeping has not previously been studied in evolutionary robotics.

The primary contributions of this paper are threefold. The first concerns input to the simulated controller. Typically, neural controllers work with feedback obtained from both sensory input and physical hardware responses. However, we found that decoupling the controller from direct motor feedback, and instead using only input from a simulated IMU (corresponding to the IMU chip we used in the physical prototype), produced effective behaviors. Second, we discovered that for this particular task, a cumulative fitness function was most effective, but that it had to ignore a significant “startup phase” in which the robot could lose fitness in the process of re-orienting itself. Third, we observed a number of interesting gaits and other movements that enabled the robot to achieve and maintain station in the presence of different laminar flows. Specifically, the evolved neural controllers in several trials were able to identify the direction of flow and correctly orient themselves through complex movements before facing their target station and then transitioning to simpler forward swimming. In ongoing work, we plan to integrate the evolved controllers into our physical prototype and evaluate it in the elliptical flow tank shown in Figure 1.

Section 2 presents an overview of the simulated robot, the simulation environment including hydrodynamic model, artificial neural networks and the station keeping task. Experiments and results are presented in Section 3, along with a description of specific evolved behaviors for different flows followed by conclusions in Section 4. Videos of the evolved behaviors are available through links provided throughout the paper.

2. METHODS

In this section, we describe the models and concepts relevant to evolving neural controllers for the aquatic robot.

2.1 Robot Model

The robotic model developed for this study was designed to emulate the form and function of a physical device, seen in Figure 1(e). As shown, the model consists of a static body, a caudal fin and two continuous rotation lateral flippers. This design has some resemblance to a biological fish, however, the functionality of the flippers is significantly different in both range of motion and possible behaviors. Specifically, the flippers are not limited to a defined range, instead exhibiting a 360° range of motion in both directions, while the caudal fin is limited to a $\pm 30^\circ$ symmetric range of motion. The fins used in this study are assumed to be rigid; other studies address flexible components [4, 17]. With these three actuated components, a wide range of three-dimensional maneuvers are possible, providing evolution with a broad slate to discover unique gait patterns for different environments and problems.

An important goal in this study is for the simulated model to mimic the physical prototype in form and sensing abilities. Despite the availability of servo encoders, which provide feedback on the state of the motors, the capabilities of the robot were kept minimal in order to examine how the evolved solutions perform with limited sensory information. This disconnect between mechanical positioning and the control signal sent to a motor creates a situation in which the neural controller is dependent only upon its perception of the surrounding environment, rather than feedback from a motor position, which may be incorrect over time due to hardware decay. Hence, the virtual robot does not have exact position or speed information for its servo motors. Instead, inertial data (i.e., linear and angular acceleration) is provided by a simulated inertial measurement unit (IMU), matching the hardware of the physical prototype. Thus, the robot's controller must interpret sensory data provided by the simulated IMU to determine how actuators change the body's state. In this study we do not consider error in the IMU data, which is relegated to future work.

2.2 Simulation Environment

The simulation environment used to evaluate candidate solutions is based on the Open Dynamics Engine (ODE) [7], a rigid body physics simulation engine. While ODE provides a method for resolving forces and torques into motion, it does not include fluid dynamics. For this study, we incorporated a model based on hydrodynamic drag [23]. This model evaluates hydrodynamic forces by examining each component (i.e., simulated rigid body) of the robot independently. A drag force is applied to each face of the component. Drag acts in opposition to linear velocity and is scaled by the area of a given face and a constant hydrodynamic drag coefficient. In this method, only the faces that oppose the direction of travel experience drag. Propulsion is the result of a net force generated from each individual face's force. Algorithm 1 outlines how hydrodynamic drag is calculated for all robot components. This simulation environment provides efficient computation of robot-fluid interactions while minimizing CPU time required to evaluate solutions. A more accurate fluid dynamics simulator might provide higher fidelity, but the overhead incurred by such a system would significantly limit the number of individuals and generations that we could evaluate in the same amount of time. Furthermore, in this initial phase of our investigation, we are most interested in the *general* behaviors that evolve.

Algorithm 1 Hydrodynamic model. Adapted from [19].

```
for each body do  
   $lin\_vel \leftarrow getLinearVelocity(body)$   
   $body\_rot \leftarrow getBodyRotation(body)$   
  for each face do  
     $area \leftarrow face\_area$   
     $norm \leftarrow (face\_normal * body\_rot)$   
     $force \leftarrow norm * lin\_vel * area * drag\_coeff$   
    if  $force > 0$  then  
       $addForce(force)$   
    end if  
  end for  
end for
```

2.3 Neural Controller

In this study, neural controllers are evolved with the NEAT algorithm, which evolves recurrent artificial neural networks (ANNs) with a modified genetic algorithm [20]. With NEAT, only the number of inputs and outputs must be specified, while the hidden layers and connectivity are modified through the evolutionary process. Relevant NEAT parameters used in this study were: a dropoff age of 200, survival threshold of 0.2, mutation only probability of 0.25 and a mate only probability of 0.2. For this study, we have nine inputs, three outputs, and the ANN is activated every 5ms of simulation time. Two outputs control speed of the flipper servos, and the third governs oscillation of the caudal fin servo. Three of the inputs are the previous ANN outputs, another three are the robot's current three-dimensional position (obtained through the simulated IMU data), and the final three inputs describe the difference between the current position and the target position. Inputs were chosen for their ability to model values that are available to a physical robot, which will hopefully simplify the transfer of evolved controllers from simulation to reality in future work. It is important to note that servo motor inputs for the ANN were based upon the previous outputs, and do not directly reflect the mechanical position of the simulated motors.

2.4 Evolution of Station Keeping

Station keeping is challenging in aquatic environments where environmental dynamics make it difficult for the robot to hold position for tasks such as monitoring and inspection. In our experimental setup, we consider station keeping to be the ability to hold the robot's center of mass at a desired location in space in the presence of a laminar flow. An example of a solution attempting to reach station can be seen in Figure 3. The station point is represented by the convergence of the three white lines and a white sphere) and the robot's center of mass is represented as the convergence of the three green lines. Accomplishing station keeping requires multiple steps for a robot's neural controller. First, the robot must interpret the direction and speed of flow using the inputs received from the environment. Next, the neural network is activated and produces a set of outputs based on the current inputs. Finally, motors are actuated to respond to the inputs and the control loop repeats to move toward and ultimately maintain station.

In our setup, an individual's evaluation involves an initial transient phase and an evaluation phase in which fitness is accumulated every 250ms. The transient period allows the robot to identify the direction of flow and then reorient itself using movements that might otherwise cause a decrease in fitness (i.e., moving away from the station point). We implemented this transient period after observing preliminary results where individuals were not capable of reorienting themselves to flows approaching from certain

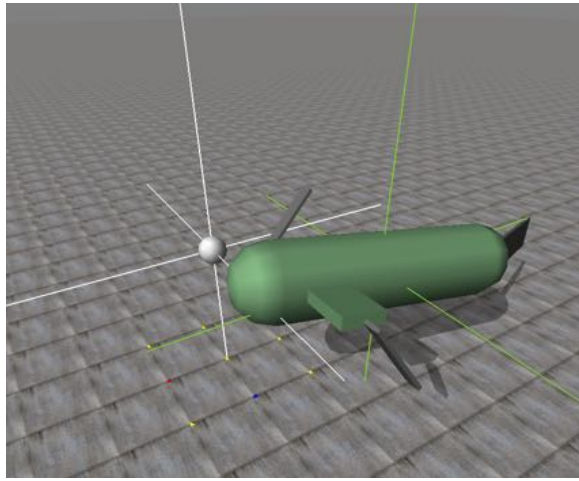


Figure 3: A demonstration of the station keeping task. The sphere and crossing lines indicate the desired station keeping point for the robot. Maximum fitness is accrued when the robot brings its center of mass, denoted by the green lines, to intersect with this station point.

directions. Instead, these solutions would attempt to hold station immediately using body orientations that were not very effective. This strategy led to the individuals drifting out of the fitness area without actually accomplishing the station keeping task. During the evaluation phase, fitness is accumulated periodically, creating an evolutionary pressure to hold station throughout an entire evaluation period, instead of just at the conclusion of the evaluation period. Fitness is calculated as follows:

$$distance = \left(\sqrt{(x_T - x_t)^2 + (y_T - y_t)^2 + (z_T - z_t)^2} \right)^3 \quad (1a)$$

$$fitness = \begin{cases} 10 - (distance) & \text{if } > 0 \\ 0 & \text{otherwise} \end{cases} \quad (1b)$$

where (x_T, y_T, z_T) is the desired station position and (x_t, y_t, z_t) is the current position. This fitness function creates a spherical zone in which individuals can accrue fitness. As an individual moves closer to the station point, its fitness score for that timestep increases. A cubic gradient encourages solutions to move toward and maintain station at the target location rather than just stay relatively close. Individuals accumulate no fitness when outside the sphere but are not explicitly penalized.

3. EXPERIMENTS AND RESULTS

Four separate trials, illustrated in Figure 4, were conducted to evolve station keeping for flows originating from different directions relative to the robot's initial orientation. Trial 1 simulates a flow from the front; Trial 2 from the rear; Trial 3 from the left; and Trial 4 from the right-front. Each trial consists of 25 replicate runs evolved for 2000 generations with a population of 100 individuals. Each individual was simulated for 120 seconds with only the second half being used to calculate the fitness. The four trials were conducted independently, with solutions being evolved to handle a specific flow situation. A time step of 5ms was used in the simulation environment, giving each individual solution a total of 24,000 neural controller updates during a run.

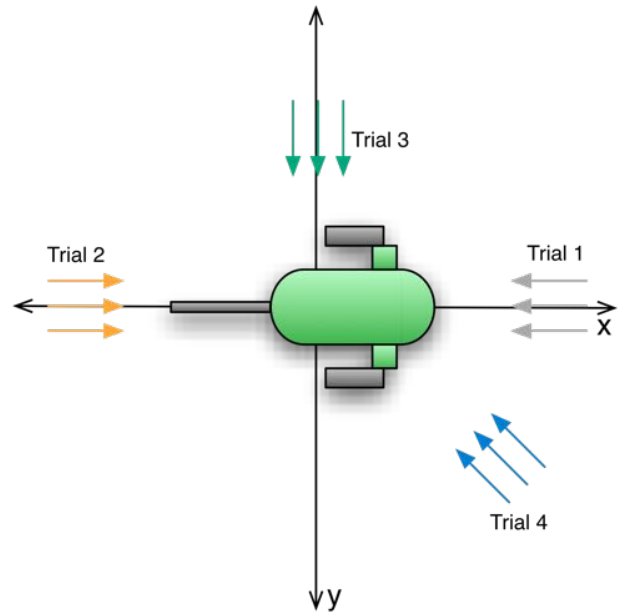


Figure 4: Direction of flow in each of the four trials. Trial 1 involved a flow coming from straight ahead. In trial 2, the flow comes directly from behind. For trial 3, the flow came from the side of the body. Finally, in trial 4 the flow came 45 degrees from straight ahead.

Evolved Behaviors.

Depending on the direction of the flow, evolved behaviors varied from simple forward swimming locomotion to complex acrobatic maneuvers that reorient the robot towards a flow. In Trial 1, a simulated laminar flow from *upstream* (i.e., from the front to the back of the robot) was applied. This configuration served as a benchmark to measure performance of the subsequent trials, as the robot did not have to reorient itself prior to maintaining station. Evolved gaits for this solution were reminiscent of natural fish locomotion, with both the caudal fin and flippers working in a coordinated oscillatory motion to swim against the flow. A video of this solution is available at http://y2u.be/fb_JzvKXfKU. Trial 2 simulated a flow pushing on the rear of the robot. Initially, we had expected to see solutions that used the flippers to maintain station while keeping the orientation relatively stable. However, evolved solutions instead developed a more effective maneuver in order to bring the caudal fin into an effective position to counteract the flow. An early individual and a late generation video can be seen in the following videos; early generation: <http://y2u.be/m8ka0Ay7wNo> and late generation: <http://y2u.be/UufbnEGFwV4>. As depicted in Figure 5, the robot flipped itself over, and then executed a forward motion similar to that seen in the first trial. Reorienting the body into an effective position for *forward* propulsion demonstrates the controller's ability to identify and counteract the force generated by a laminar flow.

In Trial 3, a simulated flow exerts force against the side of the robot. Of the four trials conducted, this proved to be the most challenging, apparently due to the difficulty of turning 90 degrees to the left in the allotted time. A video showing a middle generation evolved individual can be seen at <http://y2u.be/uszEpt-09n0>. For this trial, the expected behavior was to turn 90 degrees and face the flow without any need to rotate along another axis. However,

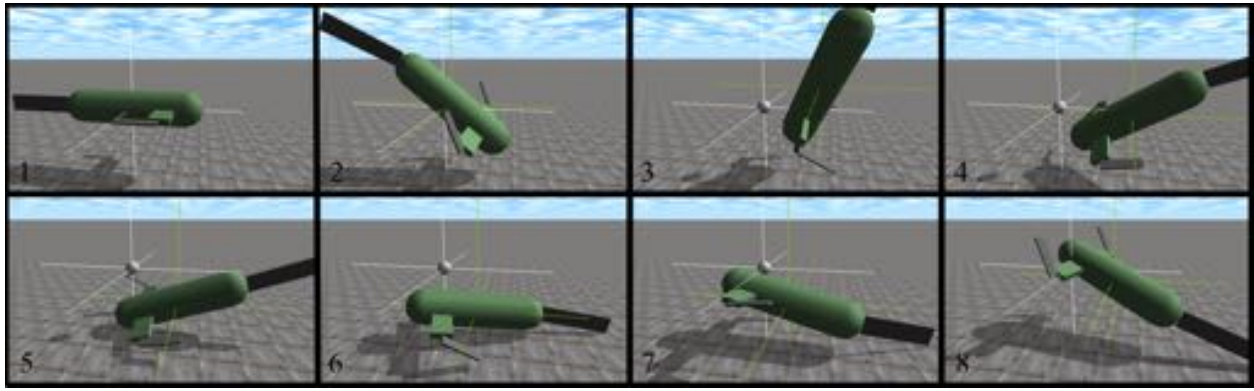


Figure 5: Behavior of an evolved solution in Trial 2. The first 60s, which is the transient phase, is utilized to reorient itself against a laminar flow pushing on the robot from left to right. The robot executes a 180 degree flip to bring the caudal fin into a position from which it can provide the greatest thrust against the flow. In this solution, the flippers are used to flip the robot as well as make minor adjustments once the robot is in an effective position. A video of this evolved solution is available at <http://y2u.be/UufbnEGFwV4>.

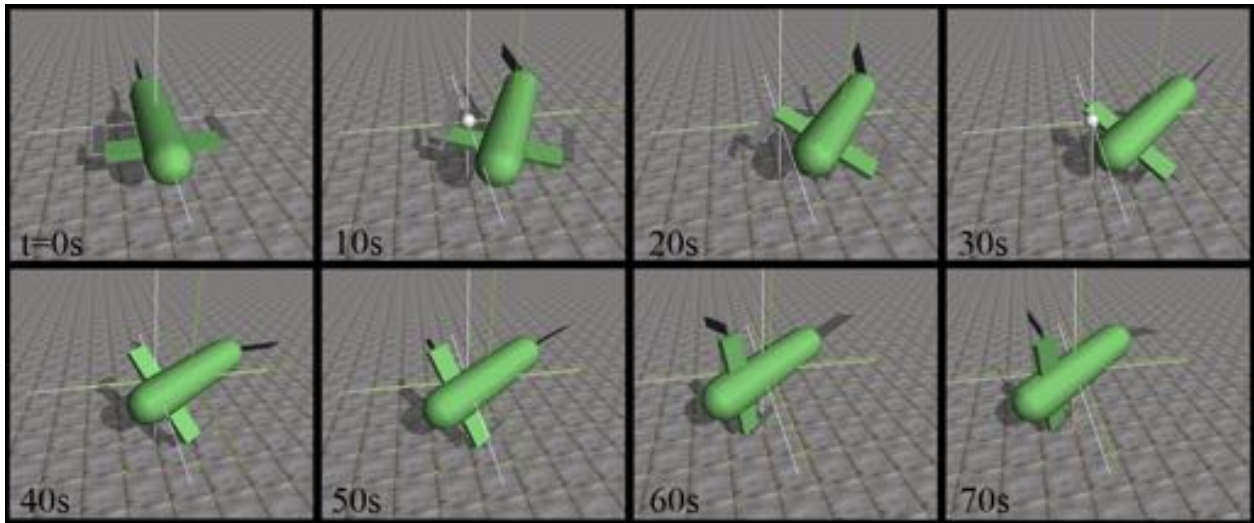


Figure 6: Action taken by an evolved solution in Trial 4. In this trial, an individual faces a laminar flow at a 45° angle to the robot's front. The robot spends the first 50s reorienting itself against the flow. After 50 seconds, the robot has achieved a relatively stable station and begins to accumulate high levels of fitness by using the flippers and fin in a coordinated effort to maintain its center over the station point. A video of this evolved behavior is available at <http://y2u.be/HIDHC3KG7Yw>.

evolved solutions instead exhibited a more complex maneuver, apparently because a 90 degree turn proved time intensive. Instead, the evolved behavior favored a combination of the flipping motion seen in Trial 2 with a roll to bring the body into an effective position for swimming against the flow. Trial 3 individuals had difficulty achieving station within the time allowed as the initial reorientation required a significant amount of the evaluation period.

For Trial 4, a simulated flow was applied at a 45 degree angle to the robot's initial right-front. As depicted in Figure 6, evolved individuals demonstrated the ability of the controller to respond to the direction of flow and attain station keeping during the course of an individual evaluation. Images in Figure 6 are taken at 10 second intervals over the first 70 seconds of simulation time. Initially, the robot is displaced from its station. The robot begins to react at approximately 10s when it starts to orient itself to the flow by using its flippers to rotate the body while the fin provides forward

propulsion. Fitness evaluation begins at 60s. By this time the robot has achieved, and can maintain, station by working to correct its position relative to the given station point. Videos of an early generation individual and the evolved solution described previously are available at: http://y2u.be/dF_5-3I6B14 and <http://y2u.be/HIDHC3KG7Yw>, respectively.

Fitness Evaluation.

Fitness results from the trials are shown in Figure 7 and 8. These results provide insight into the relative difficulty that each flow presented to the evolutionary process. Specifically, in Trial 1, where the robot directly faced the flow, solutions achieved near perfect results, where a fitness of 1 correlates to solutions that maintained station for the entire evaluation phase. Apparently, the lack of need to reorient the body helped to produce such high fitnesses. Figure 9 shows the final distribution of the best evolved individuals for

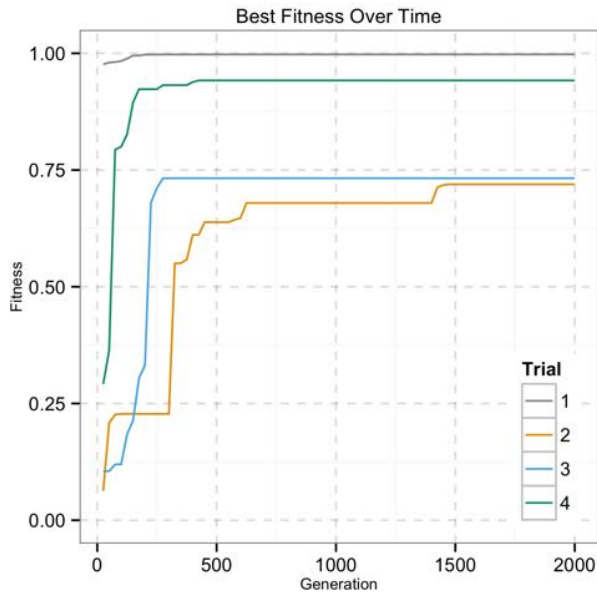


Figure 7: Fitness of the best evolved results from the trials. Trial 1 was able to achieve a near perfect fitness score as it did not have to reorient itself prior to holding station. The other trials had some success with achieving station although their fitness scores were lower than Trial 1. This was likely due to the movements required to reorient themselves to flows.

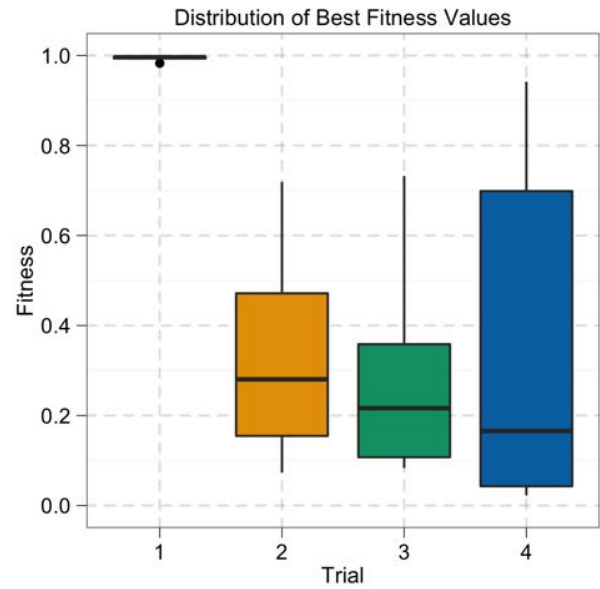


Figure 9: Box plot of the best fitness values for each replicate run in the four trials. Box indicates the upper and lower quartiles, median is represented as the center line in the boxes. Ends of the whiskers represent the maximum and minimum values, excluding outliers.

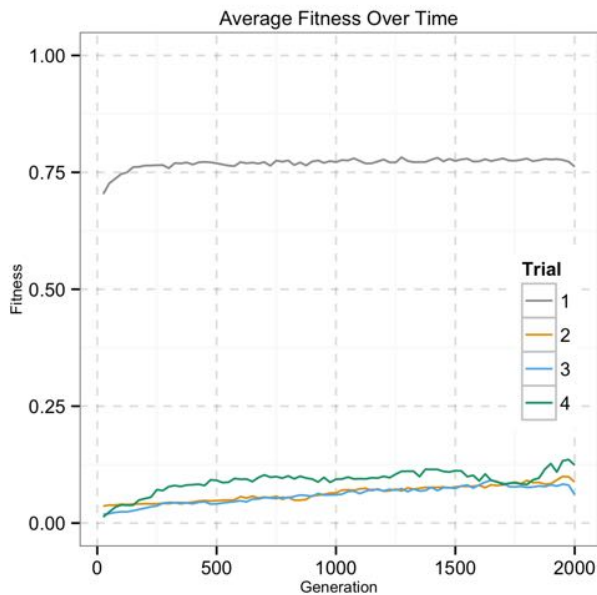


Figure 8: Average fitness of the population of evolved results from the trials. Trials 2, 3, and 4 had very low average fitnesses due to the difficulty of accomplishing station keeping. Many individuals were able to swim, but left the station area, accruing no fitness during the evaluation period.

each of the replicate runs. This plot illustrates the relative difficulty associated with the different flows. As can be seen in the results from Trials 2, 3 and 4, only a few solutions were able gain high fitness. The flows faced in these trials required the evolved individuals to both reorient themselves and develop effective swimming gaits. Typically, the evolved behaviors required more time for re-orientation and were still doing so at the beginning of the fitness evaluation phase. Average fitnesses in the final three trials were low as many individuals swam with no regard to the station keeping task. This resulted in them moving out of the fitness area prior to accumulating any measure of fitness.

Behavior Comparison.

Figure 10 presents a top down view of two evolved solutions from Trial 2 (Figure 5 presents snapshots of the final evolved individual's movement over time). In this plot, we see the final evolved solution drift outside of the fitness area before moving back in and ultimately holding station inside the reward zone. In contrast, the early generation individual, selected after 25 generations is continuously pushed farther away. This early individual started swimming only after one minute of simulation time and was not able to correctly identify the station keeping task. While the evolved solution exhibits behavior similar to that of the early individual, it executes its movements much earlier in the simulation. Videos of these two evolved solutions can be seen at <http://y2u.be/m8ka0Ay7wNo> and <http://y2u.be/UufbnEGFwV4>.

Figure 11 plots trajectory information for the final solution and an early generation solution in Trial 4 (Figure 6 presents the gait that the final evolved solution takes to achieve station). For the final evolved solution, the robot initially moves outside of the fitness gathering area. However, this maneuver occurs during the transient phase, allowing the evolved solution to move without losing

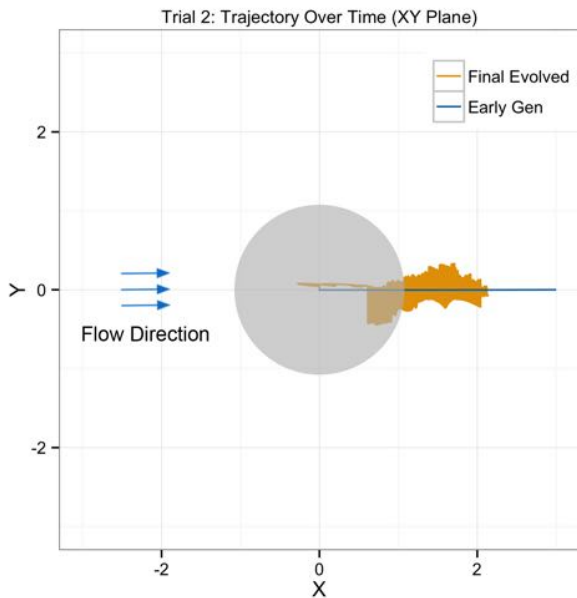


Figure 10: Two-dimensional, top-down trajectory plot for an evolved solution and an early generation candidate solution from Trial 2. The grey circle represents the area of fitness reward. The evolved solution is initially displaced by the flow before identifying the direction of flow, reacting to it and then achieving station. In the figure, the early generation individual drifts out of the plot area before moving the flippers at all. It eventually settles into a steady state of moving its flippers continuously well outside the fitness area.

fitness. The earlier controller fails to identify the flow and can be seen drifting away over time. Even though the robot does manage to swim with some coordination, it lacks the ability to identify the flow direction and coordinate its swimming to act against the force. Consequently, this individual ultimately is pushed out of the fitness reward area. Videos of these two solutions can be seen at http://y2u.be/dF_5-3I6B14 and <http://y2u.be/HIDHC3KG7Yw>.

Many evolved solutions exhibited swimming behaviors but were not able to coordinate those movements with the second task of identifying and holding against the flow. For example, one individual was able to effectively swim directly forward, however, the direction of flow in the trial (from the side) caused it to gradually drift out of the fitness reward zone, resulting in the individual gathering less than 1% of the available fitness.

4. CONCLUSIONS

One of the challenges facing any evolutionary robotics process is how to assign fitness. Specifically, station keeping required rewarding solutions for maintaining station at a desired point, while not penalizing solutions that used movements outside of the fitness area to reach more effective orientations. In preliminary tests, fitness was allowed to accumulate from the beginning of the run, creating an unnecessary pressure to perform well from the start. Such a fitness metric makes it difficult for an evolutionary algorithm to find strategies that sacrifice initial fitness for an overall better strategy. Given the robot's morphology in this study, the caudal fin produces the greatest propulsion. Therefore, an effective strategy

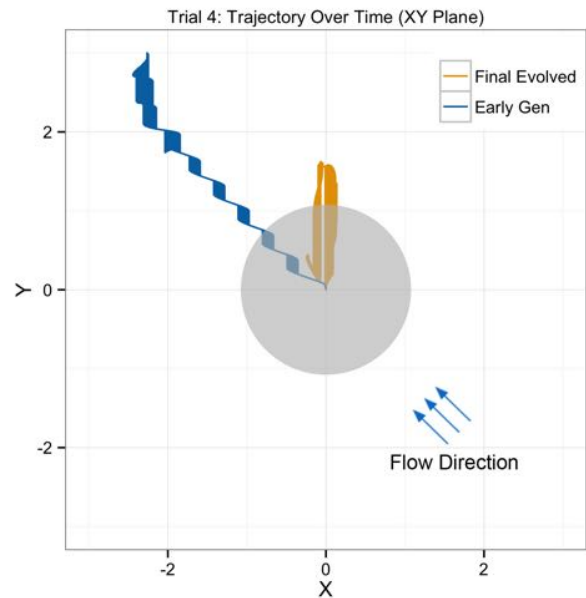


Figure 11: Two-dimensional, top-down trajectory plot for an evolved solution and an early generation candidate solution from Trial 4. The grey circle represents the area of fitness reward. The evolved solution is initially displaced by the flow before identifying the direction of flow, reacting to it and then achieving station.

is to first reorient the body towards the flow, and then focus on forward motion. After observing results of preliminary runs, we implemented a transient period. Rather than integrate additional metrics into the fitness function to account for movement, which might bias solutions, the transient period allows for a movement period without losing fitness. Moreover, the transient period allowed for movements we had not expected, such as in Trial 3 where the robot executed both a 180 degree flip *and* a 90 degree rotation. However, even with the addition of a transient phase, Trial 3 was still quite difficult. In the future, we will repeat the experiment with different length transient periods and eventually allow the evolutionary algorithm to adapt the transient and evaluation periods to appropriate durations. This might allow behaviors to evolve with a transient phase suitable for the given task.

A second observation concerns inputs to the controller. Physical motors can suffer from performance degradation over time as well as unexpected hardware failures. Predicting these inconsistencies and failures can be challenging, if not impossible, and incorporating these behaviors into simulation is equally difficult. Decoupling the controller from motor feedback can potentially alleviate some of the problems experienced with noisy motors misreporting their current state. In this approach, evolved results rely on the overall behavior of the robot and information obtained through sensors, regardless of actual motor performance. As demonstrated in this study, the neural controller is able to identify the environmental conditions through the simulated IMU sensor values and maintain a given station without the need for physical motor feedback.

Evolution of aquatic robots has been performed previously beginning with the work of Sims [19]. Recent studies have focused on optimizing the caudal fin of robotic fish [4]. In this study, solutions exhibited unexpected locomotion strategies that involve both

simple swimming gaits as well as more complex acrobatic maneuvers to reorient the robot to different laminar flows. We have shown that neuroevolution is capable of generating control strategies to address station keeping against a variety of different flow situations. Our ongoing work includes evaluating controllers in a physical robot placed in an elliptical flow tank. We also plan to pursue more generalized control strategies capable of holding station against dynamic flow situations. Future work will include developing more accurate simulations and addressing issues arising in transferring neural controllers from simulation to reality.

Acknowledgments

The authors gratefully acknowledge the contributions and feedback on the work provided by members of the Software Engineering and Network Systems Laboratory, the Smart Microsystems Laboratory, and the BEACON Center at Michigan State University. This work was supported in part by National Science Foundation grants CNS-1059373, CNS-0915855, DBI-0939454, CCF-0820220, and CNS-0751155.

5. REFERENCES

- [1] J. M. Anderson and N. K. Chhabra. Maneuvering and stability performance of a robotic tuna. *Integrative and Comparative Biology*, 42(1):118–126, 2002.
- [2] N. Bredeche, E. Haasdijk, and A. Eiben. On-line, on-board evolution of robot controllers. In P. Collet, N. Monmarché, P. Legrand, M. Schoenauer, and E. Lutton, editors, *Artificial Evolution*, volume 5975 of *Lecture Notes in Computer Science*, pages 110–121. Springer Berlin / Heidelberg, 2010.
- [3] Z. Chen, S. Shatarra, and X. Tan. Modeling of biomimetic robotic fish propelled by an ionic polymer metal composite caudal fin. *IEEE/ASME Transactions on Mechatronics*, 15(3):448–459, 2010.
- [4] A. J. Clark, J. M. Moore, J. Wang, X. Tan, and P. K. McKinley. Evolutionary design and experimental validation of a flexible caudal fin for robotic fish. In *Proceedings of the 13th International Conference on the Simulation and Synthesis of Living Systems*, East Lansing, Michigan, USA, 2012.
- [5] J. Clune, B. E. Beckmann, C. Ofria, and R. T. Pennock. Evolving coordinated quadruped gaits with the HyperNEAT generative encoding. In *Proceedings of the IEEE Congress on Evolutionary Computing*, pages 2764–2771, Trondheim, Norway, 2009.
- [6] M. Epstein, J. E. Colgate, and M. A. MacIver. Generating thrust with a biologically-inspired robotic ribbon fin. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2412–2417, Beijing, China, 2006.
- [7] R. S. et al. Open Dynamics Engine, <http://www.ode.org/>, 2012.
- [8] J. Faria, J. Dyer, R. Clément, I. Couzin, N. Holt, A. Ward, D. Waters, and J. Krause. A novel method for investigating the collective behaviour of fish: introducing ‘robofish’. *Behavioral Ecology and Sociobiology*, 64:1211–1218, 2010.
- [9] D. Floreano and L. Keller. Evolution of adaptive behaviour in robots by means of darwinian selection. *PLoS Biology*, 8(1), 2010.
- [10] F. Gomez and R. Miikkulainen. Active guidance for a finless rocket using neuroevolution. In *Proceedings of the 2003 Genetic and Evolutionary Computation Conference*, pages 2084–2095. Springer Berlin / Heidelberg, San Francisco, California, USA, 2003.
- [11] F. J. Gomez and R. Miikkulainen. Active guidance for a finless rocket using neuroevolution. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 2084–2095, San Francisco, 2003. Morgan Kaufmann.
- [12] Q. Hu, D. R. Hedgepeth, L. Xu, and X. Tan. A framework for modeling steady turning of robotic fish. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2669–2674, Kobe, Japan, 2009.
- [13] S. Koos, J. B. Mouret, and S. Doncieux. Crossing the reality gap in evolutionary robotics by promoting transferable controllers. In *Proceedings of the 2010 ACM Genetic and Evolutionary Computation Conference*, pages 119–126, New York, New York, USA, 2010. ACM.
- [14] G. V. Lauder and E. G. Drucker. Morphology and Experimental Hydrodynamics of Fish Fin Control Surfaces. *IEEE Journal of Oceanic Engineering*, 29(3):556–571, 2004.
- [15] J. Lehman and K. O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223, June 2011.
- [16] S. Marras and M. Porfiri. Fish and robots swimming together: attraction towards the robot demands biomimetic locomotion. *Journal of the Royal Society of Interface*, 2012.
- [17] J. M. Moore and P. K. McKinley. Evolving Flexible Joint Morphologies. In *Proceedings of the 2012 ACM Genetic and Evolutionary Computing Conference*, Philadelphia, Pennsylvania, USA, 2012. ACM.
- [18] C. Richter and H. Lipson. Untethered hovering flapping flight of a 3D-printed mechanical insect. *Artificial Life*, 17:73–86, 2011.
- [19] K. Sims. Evolving virtual creatures. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, pages 15–22, 1994.
- [20] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, June 2002.
- [21] X. Tan, M. Carpenter, J. Thon, and F. Alequin-Ramos. Analytical modeling and experimental studies of robotic fish turning. In *Proceedings of the 2010 IEEE International Conference on Robotics and Automation*, pages 102–108, Anchorage, Alaska, USA, 2010.
- [22] X. Tan, D. Kim, N. Usher, D. Laboy, J. Jackson, A. Kapetanovic, J. Rapai, B. Sabadus, and X. Zhou. An autonomous robotic fish for mobile sensing. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5424–5429, Beijing, China, 2006.
- [23] J. Wang, F. Alequin-Ramos, and X. Tan. Dynamic modeling of robotic fish and its experimental validation. In *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 588–594, San Francisco, California, USA, 2011.