Fish Robot: Design, Control and Evolution of Undulatory Swimming Gaits

Technical Report

Ivan Tanev
Department of Information System Design
Faculty of Science and Engineering,
Doshisha University, Kyoto, Japan
itanev@mail.doshisha.ac.jp
http://isd-si.doshisha.ac.jp/itanev/

July 16, 2021

Original location of the report: http://isd-si.doshisha.ac.jp/itanev/TR/TR_20210716_Fishbot.pdf

Movie about the presented system: http://isd-si.doshisha.ac.jp/itanev/movies/Fishbot_Summary_mp4_LR.mp4

Abstract: We report on designing, controlling, and evolving the undulatory swimming gaits of a fish robot. Also, we present the experimental results obtained from the evolution – via genetic algorithms – of the gaits that result in maximum thrust of propulsion exerted by the swimming robot. We view these results as a proof of soundness of the concepts that we have relied on in the development of the robot and its evolutionary framework.

Keywords: fish robot, undulation, central pattern generator, evolutionary computation, genetic algorithms

Glossary

abstract layer of control	*12-17*	*operational reliability*	*7*
alleles	*20*	*optical tracker*	*25, 28*
binary tournament selection	*21*	*pectoral fins (movable)*	*26, 27*
block diagram (of entire system)	*29*	*physical layer of control*	*12, 18*
buoyancy	*4*	*QueryPerformanceCounter (function)*	*15*
central pattern generator (CPG)	*12*	*radius of gyration*	*8-10*
centre of buoyancy	*5*	*repositioning*	*26-27*
centre of gravity	*5*	*Reynolds number*	*4*
cooldown	*17*	*rotational momentum*	*5*
crossover	*21*	*sampling interval (of CPG)*	*16, 28*
degrees of freedom	*3*	*servo motors*	*3, 10*
delta function	*17*	*Strouhal number*	*32*
fitness evaluation	*21, 23-25*	*termination criterion*	*21*
fitness convergence characteristics	*30*	*torque (of servo motor)*	*8, 9*
GA Manager	*20-23, 28*	*transient acceleration*	*17*
genetic representation	*20*	*warmup*	*17*
heursitical estimation (of fitness)	*28*	*water sealing*	*5*
jitter (of servo motor)	*15*	*weight distribution*	*5*
momentum of inertia	*3, 6*	*yaw maneuverability*	*3, 6*
mutation	*21*	*yaw momentum*	*6*
omega-shaped spring	*16*		

1. Introduction

Vortices significantly influence the hydrodynamics of locomotion of natural fishes [2, 16], and similarly – of fish robots. However, the software simulation of vortices is far from perfect, and an inevitable gap between the simulation and the reality would compromise the credibility of the results, obtained from an eventual software model of the robot [12]. Therefore, we assume that a real, physical robot (rather than a software simulation) should be used in order to obtain realistic result about the fish locomotion [9, 13, 17, 18, 19, 21, 22]. The *objectives* of the research, presented in this document are (i) to build a physical fish robot, and (ii) to develop a software system that would allow us optimize its swimming gaits. In the development of the robot we mimicked the rainbow trout fish (*Oncorhynchus mykiss)* in nature. The locomotion of the fish belongs to the group of sub-carangiform, implying that the propulsion is achieved solely by undulating the caudal (tail) fin and the rear part of the body. Therefore, we relied on the two-joint morphology of the robot, where each joint is being moved by a dedicated servo motor. Regarding the second objective, we adopted genetic algorithm (GA) in order to evolve the values of the key parameters, pertinent to the locomotion of the swimming fish robot, that result in maximum thrust. GA, as a heuristic approach requires little domain-specific knowledge, and consequently, a precise understanding of the physics of vortices is not required [3, 5, 10, 11]. As an application domain, we view the monitoring and management of ecological systems as main application domain.

The remainder of this documents is organized as follows. In section 2 we explain the design of the fish robot. Section 3 presents the central pattern generator that is used to control the undulatory swimming gaits of the robot. Section 4 elaborates on the evolutionary framework, used to evolve the values of parameters of undulation that result in maximum thrust generated by the robot. In section 5 we describe the experimental setup and explain the three main components of the developed software system. Section 6 discusses the obtained experimental result on the evolution of the gaits that result in maximum thrust generated by the swimming robot. Finally, section 7 draws a conclusion.

2. Design

In this section we will present the rationale of the adopted morphology of the fish robot. In addition, we will elaborate on the design considerations that we took into account in the engineering of the robot.

2.1 Morphology of the Fish Robot

Both the shape and size of the fish robot are intended to mimic – as realistically as possible – the rainbow trout fish (*Oncorhynchus mykiss)* in nature. The primary motivation to consider such locomotion is that its efficiency heavily depends on the generation- and utilization of vortices. The effects of the vortices on the dynamics of the fish could be faithfully investigated only in a physical model (rather than in a software simulation) of the fish and its environment, which, in turn, vindicates our decision to build a real (physical) robot instead of its software simulator.

In the adopted morphology, the robot consist of three segments – front-, rear-, and the tail (caudal fin), connected via two hinge joints. The first (front) joint connects the front- and the rear segments, while the second joint connects the rear- and the tail segments of the robot. We choose the location of the front joint – to be close, and slightly behind the centre of mass of the robot – with an intention to achieve both (i) a low *moment of inertia* of segments and (ii) an improved *yaw maneuverability*, as we elaborate later in the subsection 2.2 Design Considerations: Favorable Weight Distribution.

The relative angular movement of each of these two joints is accomplished via two respective *servo motors*. The first (front) servo motor is mounted on the front segment and its push rod is attached to- and moves the rear segment, while the second (rear) servo motor, being mounted on rear segment of the robot, moves the tail via a corresponding push rod. The number of *degrees of freedom* of such a design is two. The side view – with the fairings removed – and the kinematic diagram of the fish robot are depicted in Figure 1. The view of the robot submerged in water is shown in Figure 2.

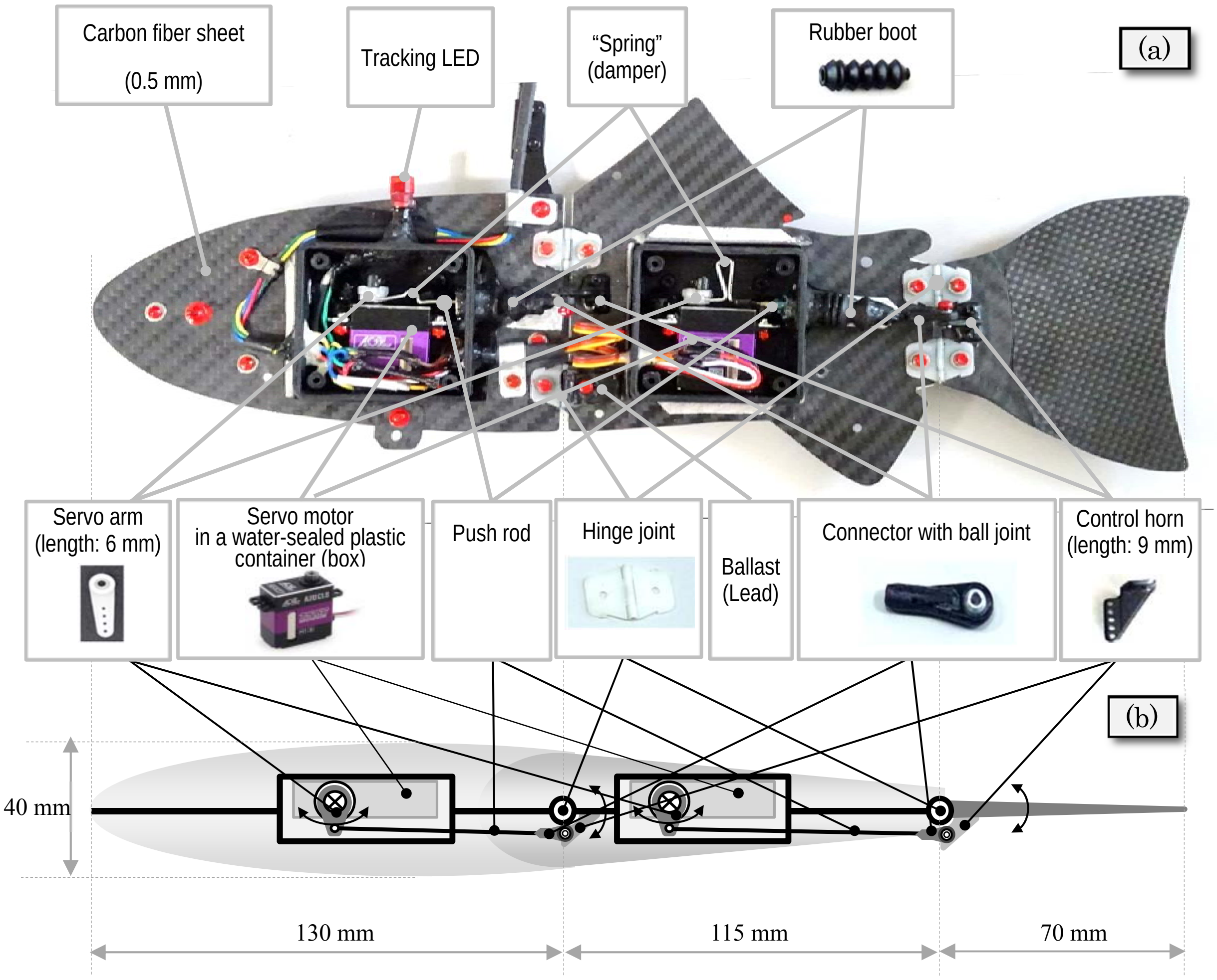

Figure 1. Side view of the fish robot without fairings (a) and its kinematic diagram (b).

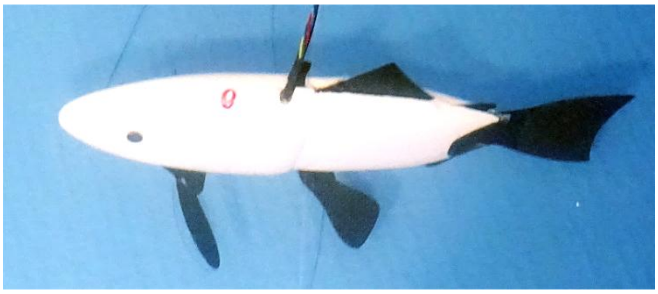

Figure 2. View of the fish robot submerged in water.

2.2 Design Considerations

In the design of the robot we attempted to satisfy the following engineering requirements:

- Lightweight design with good structural integrity,
- Water sealing of the components,
- Favorable weight distribution, and
- Operational reliability.

Below we shall elaborate on the solutions we adopted in order to satisfy the above mentioned requirements.

Lightweight Design with Good Structural Integrity

The design of the robot should fulfill the somehow contradicting requirements of being a lightweight, yet robust enough to sustain the significant forces that act on the undulating segments in water. The lightweight is required in order to allow a near neutral (yet slightly negative) *buoyancy* of the robot. An eventual too heavy robot will feature too strong negative buoyancy, which, in turn will compromise the plausibility and the realism of the robot, which, due to the significantly different *Reynolds number*, might widen the gap between the features of the obtained optimal swimming gaits of the robot and the corresponding features of its natural counterpart. In addition, an eventual too heavy segments of the robot would increase the wear and tear of the moving components (servo motors, hinge joints between the segments, ball joints of the push rods, etc.), which, in turn would compromise the operational reliability of the robot.

To achieve a lightweight, yet strong design, we built body of the robot from two thin (0.5 mm each) sandwiched lightweight sheets – a carbon fiber-, and aluminum one, glued together via an epoxy resin. Such a composite design offers a combination of the benefits of the strength, stress resistance, flexibility, and vibration damping of the carbon fiber sheet with the plasticity (indispensable for the design of housing of the water-sealed plastic boxes containing the servo motors) of the aluminum sheet. In addition, both the carbon fiber- and the aluminum sheets are corrosion resistant.

Sealing of the Components

The servo motors are placed in water-sealed plastic containers (boxes). An alternative solution to the sealing of the containers of servo motors would have been to adopt waterproof servo motors instead. However, most of the commercially available waterproof servo motors require that the latter should not be soaked in water (and, let alone, subjected to external water pressure if submerged) during the operation.

The *water sealing* of each of the two containers of servo motors is achieved by sealing of the following components:

- Two bolts (M2.6) of the mounting of the container to the body of the robot, sealed by improvised O-rings. These improvised O-rings are formed as circular cavities between the mounting bolts and the walls of the container, and are filled with liquid rubber waterproof sealant.
- Push rod, sealed by plastic rubber boot. Both ends of the boot are sealed by improvised O-rings filled with waterproof liquid rubber sealant.
- Caps of the containers, with edges sealed by improvised O-ring filled with waterproof liquid rubber sealant.
- Four screws that fasten the mounting of the cap of the container, sealed by improvised O-ring filled with waterproof liquid rubber sealant, and
- Electrical wires (providing the power supply and the control signal to the servo motor), sealed by waterproof epoxy resin.

Favorable Weight Distribution

We considered the *weight distribution* from two aspects: the alignment of the points where the relevant forces appear to act on a submerged fish robot, and (ii) the *rotational momentum* of the moving segments.

Regarding the first aspect, there are five major – balancing each other – forces that act on the swimming robot: propulsion, drag, gravity, buoyancy, and suspension. The former two act on a horizontal-, while the latter three – on the vertical plane. Because the points of application of the former two forces depend solely on the shape of the body of the robot, which we built with the intention to mimic the body of the rainbow trout as closely as possible, we will focus on the acting points of the latter three balanced forces, namely, gravity (F_G), buoyancy (F_B), and suspension (F_S). In order to guarantee that the robot will be in upright position in the water (i.e., with zero roll and pitch angle), and that position will be stable regardless of any external disturbances, the points of application of these three forces – *centre of gravity* (COG), *centre of buoyancy* (COB) and suspension point (SP), respectively – should be aligned on the vertical line. Moreover, in order to allow the forces to create a stabilizing torque (around both roll- and pitch axes) in case of external disturbance to the special position of robot, COB should be located above the COG, and SP – above both of these points, as illustrated in Figure 3. The actual values of all vertical forces are shown in Table 1.

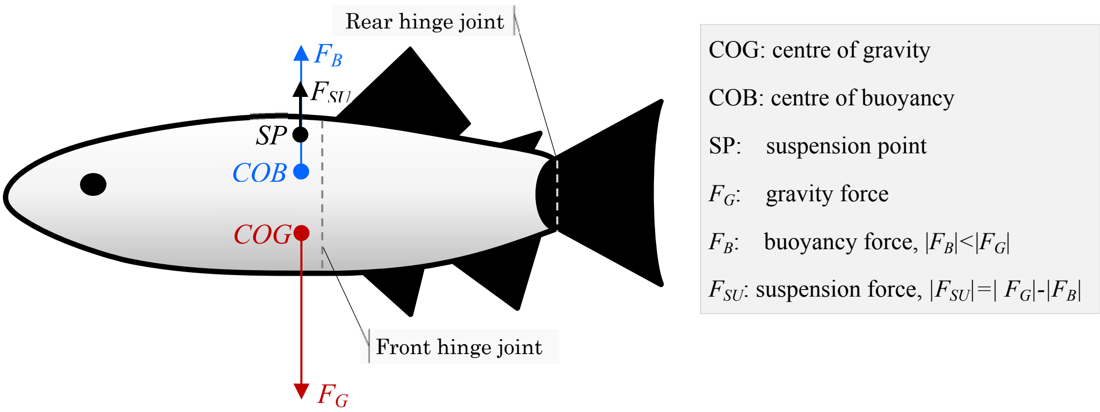

Figure 3. Location of the centre of gravity, centre of buoyancy, and suspension point of the fish robot.

Table 1. Actual values of vertical forces acting upon the submerged fish robot

Parameter	Value
Mass of the robot *m*, kg	0.432
Gravity force (weight), F_G, kg	0.432
Volume of displaced water, V_W, l	0.334
Buoyancy force, F_B, kg	0.334
Buoyancy type	Negative, $\|F_G\| > \|F_B\|$
"Sinking" force, $F_{SI} = F_G - F_B$, kg	0.098
Suspension force, $\|F_{SU}\| = \|F_S\|$, kg	0.098

The second aspect of the weight distribution implies a low *momentum of inertia* of the moving segments. This could be achieved by keeping the center of the moving masses (i.e., the mass of the "dry" segments plus the mass of the water trapped in the cavities between the fairings and the segments) as close as possible to the pivot point, i.e., the hinge joint between the front and rear segments. Low momentum of inertia of segments results in lower inertial forces acting upon the undulating segments of the robot, and consequently, lower wear-and-tear of its moving components (hinge joints, connectors with ball joints, and especially – the servo motors).

Moreover, because the centre of the mass of the robot is close to the joint between the front and rear segments, the lower moment of inertia of segments would apparently result in lower *yaw momentum* of inertia of the robot as a whole. This would improve the *yaw (turning) maneuverability* of the robot, which would be beneficial in case the robot has to navigate confined environments. The improved yaw agility was our primary motivation in choosing the location of the front joint (Figure 1) – to be close to (and little behind of) the middle of the robot and consequently – to the cetre of its mass.

In the following subsection *Operational Reliability* we shall discuss the implications of location of center of mass of segments on the torque that servo motors apply while undulating these segments.

We achieved both the favorable alignment of the vertical forces, and the low rotational momentum of the segments by handcrafting the fairings from Styrofoam. Because the Styrofoam features about 20 times lower density than that of water (0.05 g/cm^3 vs. 1 g/cm^3), the fairings attached to the robot create significant buoyancy forces, yielding a positive overall buoyancy of the robot. This, in turn, provided us with the opportunity to attach lead ballast (in order to achieve the intended slightly negative buoyancy) at the most beneficial locations – near the center of the desired centre of gravity, as lower as possible, and as close as possible to the joint (pivot point) between the front and rear segments. Moreover, in order to reduce the rotational momentum of the water trapped in the cavities between the fairing and the body of the robot, we varied the thickness of the internal walls of the fairings – the walls near the tip and the tail of the robot are much thicker than those close to the pivot point. Similarly, in order to elevate the centre of buoyancy as much as possible above the centre of gravity, the walls of the upper half of fairings are much thicker than those of the lower half of fairings.

Operational Reliability

Operational reliability implies that the robot should be function reliably for extended period of time under the intended operational conditions. Considering the fact that the servo motor is the component that suffers from most wear-and-tear, the design of reliable robot involves solving the following two main tasks:

- Task #1: Defining the operational conditions relevant to the wear-and-tear of the servo motors,
- Task #2: Choosing a proper servo motor with technical characteristics that are adequate for the given operational conditions.

The remaining of this subsection discussed how we solved these two main tasks.

Task #1: Defining the operational conditions

The failures of servo motors are usually attributed to the damage of one of the following internal components:

- IC controller of DC motor: the IC controller might suffer from burn-out due to overheating, which, in turn, is caused by too strong currents supplied to the DC motor for prolonged period of time. The primary reason for such strong currents is that the servo exerts a torque that exceeds the maximum torque specified in technical documentation of the servo.
- Reduction gears: Gears might be damaged as a result of servo exerting a torque (or being subjected to an external torque) that exceeds the designated one.
- DC motor: the brushes and collectors of DC motor might suffer from premature wear due to the excessive sparking, caused by the excessive electrical currents consumed by the DC motor. Again, the primary reason is that the servo is subjected to a torque (for extended period of time) that exceeds the designated one.

Hence, the common reason for all these failures of the servo motor is that the latter is subjected to a torque (for extended period of time) that is higher than the designated one. Therefore, the task of defining the operational conditions of the servo could be paraphrased as defining the *maximum value of the torque* that servo has to apply while undulating the segments of the robot.

Ignoring the forces of the surrounding water that act on the undulating segment, and focusing only on the inertial forces acting on the undulating segments of the robot, we could approximate the maximum *torque* τ_{MAX} that servo motor has to exert in order to overcame these forces as:

$$|\tau_{MAX}| = |F_{MAX}|.R.L_{SA}/L_{CH} = m_S.|a_{L\_MAX}|.R.L_{SA}/L_{CH} \quad (1)$$

where:

F_{MAX} is the maximum value of inertial force acting on the segment of the robot,

m_S is the overall mass of the segment (i.e., "dry" mass plus the mass of the water trapped in the cavity between the fairings and the body of the robot),

a_{L_MAX} is the maximum linear acceleration of the centre of the mass of the segment, and

R is the radius of gyration of the segment – the distance between the centre of the mass of the segment and the pivot point (point of rotation of the joint),

L_{SA} is the length of the servo arm, and

L_{CH} is the length of the control horn (Figure 4),

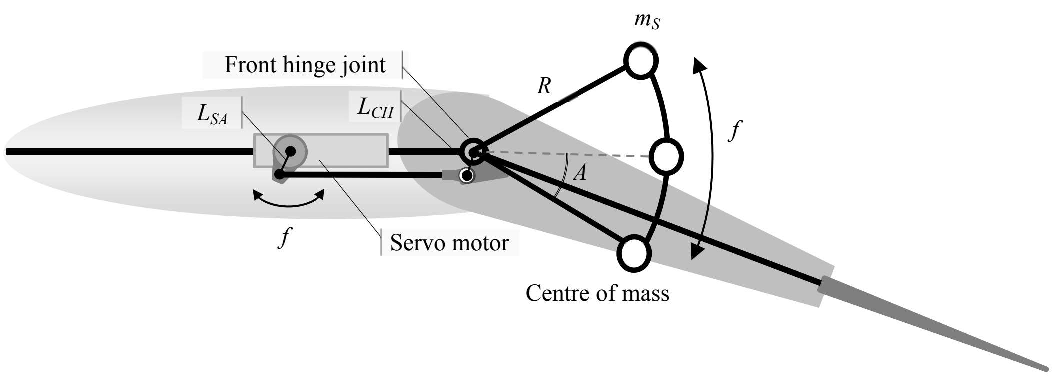

Figure 4. Undulation of the rear segment of the fish robot.

The value of maximum linear acceleration a_{L_MAX} of the centre of the mass could be calculated from the maximum angular acceleration a_{A_MAX} and the *radius of gyration* R:

$$a_{L\_MAX}=R.\ a_{A\_MAX} \quad (2)$$

The angular acceleration a_A is a derivative of the angular speed ω, which, in turn – is a derivative of the angular displacement α:

$$\alpha = A\ .\ sin(2.\pi.f.t) \quad (3)$$

where:

A is the angular amplitude of oscillation of the segment,

f is the frequency of oscillation, and

t is the time

Then, we could obtain the angular speed (ω) and angular acceleration (a_A) of the segment as follows:

$$\omega = d(\alpha)/dt = A \,.\, 2.\pi.f.\cos(2.\pi.f.t) \quad (4)$$

$$a_A = d(\omega)/dt = - A \,.\, (2.\pi.f)^2.\sin(2.\pi.f.t) \quad (5)$$

The maximum value of the angular acceleration of the segment a_{A_MAX} would be achieved for the maximal value of sinusoidal component (i.e., 1), and for the maximal frequency of oscillation f_{MAX}, i.e.,

$$|a_{A\_MAX}| = A \,.\, (2.\pi.f_{MAX})^2 \quad (6)$$

Substituting the value of maximum angular acceleration from (6) in (2) and (1), we can rewrite the latter as:

$$\tau_{MAX} = m_S.\, A \,.\, (2.\pi.\, f_{MAX})^2.R^2.L_{SA}/\,L_{CH} \quad (7)$$

Considering the equation (6) and that the moment of inertia I could be expressed as

$$I = m_S.\, R^2 \quad (8)$$

Then the Equation (7) could also be rewritten as

$$\tau_{MAX} = I.\, |a_{A\_MAX}|.\, L_{SA}/\,L_{CH} \quad (9)$$

Figure 5 depicts the values of τ_{MAX} obtained from Equation (7) for the fixed (actual) values of parameters m_S, A, L_{SA} and L_{CH}, as shown in Table 2, and the eventual variations in the values of the maximum frequency of oscillation f_{MAX} and the *radius of gyration R*.

Table 2. Values of parameters of the fish robot that define the maximal torque applied by the servo

Parameter	Value
Overall mass of the rear segment m_S, kg	0.25
Angular amplitude of oscillation A, rad (degrees)	π/6 (30)
Maximum frequency of oscillations f_{MAX}, Hz	2.4
Radius of gyration R, m	0.5
Length of the servo arm L_{SA}, mm	6
Length of the control horn L_{CH}, mm	9

As shown in Figure 5, reducing the value of parameter *R* (and, consequently, the momentum of inertia *I*), for the same values of the remaining parameters, yields a quadratic (i.e., faster-than-linear) reduction of the maximum value of torque that servo need to apply to undulate the segment.

The way of calculating the maximum torque applied by the rear servo (undulating the caudal fin) is rather similar. However, because the mass, the distance of the centre of the mass to the pivot point, and the effect of the surrounding water (due to the much lower area of the fin compared to that of the rear segment of the body), the values of torque are negligible compared to the values discussed above.

Task #2: Choosing a proper servo motor

For the highlighted values of parameters, shown in Figure 5, the maximum value of torque τ_{MAX} is 0.018 kg.m. However, in our calculations of this value we have ignored the forces that the surrounding water would have applied to the undulating segment and the caudal fin attached to it. These forces would depend on the area of the rear segment and the caudal fin, their respective coefficient of hydrodynamic drag (depending on shape), relative speed and direction of the fluid around the segments, and the vortices that the segments would generate. Assuming that these forces are in the same order of magnitude as the inertial forces that contribute to calculated value of maximum torque τ_{MAX} and adding a margin of about 40%, we choose a servo motors featuring a maximum value of torque τ_{MAX} = 0.05 (for power supply voltage *Vcc*=5 V). Moreover, in order guarantee an improved reliability of other potential points of failure -- the gears and the DC motor, we selected – for both the front- and read joint of the robot – the servo model A20CLS (Figure 6) featuring metal gears, and coreless DC motor, respectively [4]. The latter is characterized with reduced sparking, which, in turn, reduces the wear and tear of brushes and collectors of the DC motor. The most relevant characteristics of the chosen servo motor are shown in Table 3.

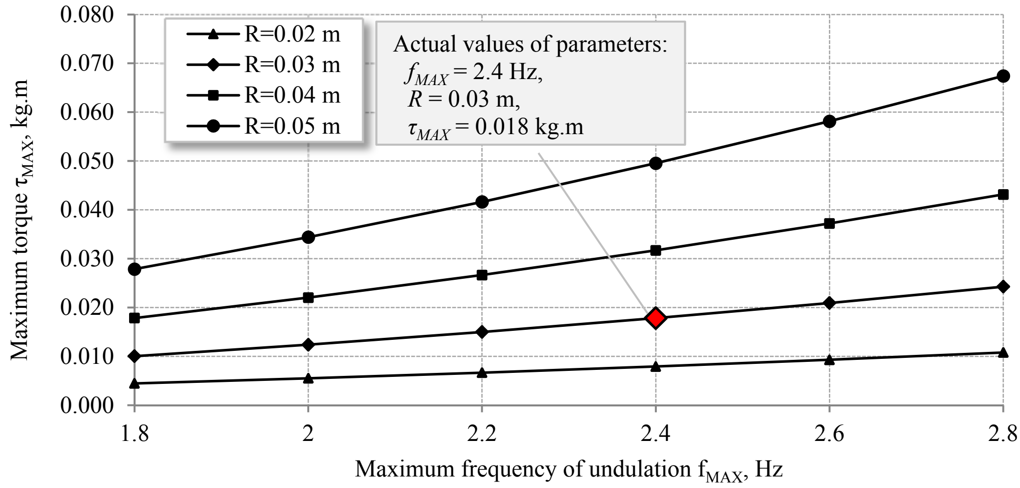

Figure 5. Values of maximum torque τ_{MAX} for different values of the radius of gyration *R* and maximum undulating frequency f_{MAX}.

Figure 6. Servo motor A20CLS

Table 3. Main characteristics of the servo motor A20CLS

Parameter	Value
Maximum torque (at *Vcc*=5 V), kg.m	0.05
Maximum angle of rotation, degrees	±55
Unload angular speed of rotation, degrees/s	460
Type of DC motor	coreless
Material of the gears	Titanium
Material of the case	Aluminium
Dimensions, mm	27.5✕23✕12
Weight, g	20

2.3 Discussion

Our choice of the number of joints – two – could be seen as a tradeoff between the plausibility (realism) of the design and the size of search space of the algorithms employed for the search of the optimal undulation of these joints. Indeed, a higher number of joints would enhance the realism of the physical design of the robot. On the other hand, however, an increased number of joints would result in exponential increase of the size of search space of, say, the genetic algorithms. Indeed, the number of combinations of the values of parameters of undulations, pertinent to each of these joints would increase exponentially. Ultimately, reducing the number of joints to just one would have been adequate for the modeling the natural fish belonging to the thunniform group, such as tuna. These group swims by undulating mostly the tail section of their body. However, the mimicked rainbow trout fish, which belongs to the group of sub-carangiform, achieves its propulsion solely by undulating both (i) the caudal (tail) fin and (ii) the rear part of the body. Therefore, a single joint (e.g., near the caudal fin) would have been inadequate to model such an undulation.

3. Control: Central Pattern Generator

3.1. Architecture of the Control System

The servo motors are controlled by setting their target turning angle via a pulse signal. In accordance to the concept of the pulse-width modulation (PWM), the width (duration) of the pulse – typically in the range between 1 ms and 2 ms – encodes for the target angle of the servo. The pulse width of 1 ms is interpreted by servo as an command to turn to the minimal turning angle (e.g., -55 degrees for the A20CLS servo, used in the fish robot), while the width of 2 ms – to the maximum angle (e.g., +55 degrees). The pulse width of 1.5 ms corresponds to the neutral position of the servo (i.e., 0 degrees). The control pulses are transmitted to the servo motor with some periodicity, typically – 50 times per second, corresponding to the sample period of 20 ms.

The precision in positioning of servo is determined by the precision of the pulse width, which, in turn, could not be guaranteed if the latter is defined by software running on the general-purpose, multitasking OS (such as MS Windows). Therefore, in our approach, we developed the controller of the servo as a system comprising two layers – *abstract*- (AL) and *physical* (PL) layers, as follows (Figure 7):

- AL: at discrete (sampling) intervals of 20 ms AL determines the values of target turning angles of the servo motors, and transmits these values to PL,
- PL: receives the values of target turning angles of the servo motors, converts these values into corresponding control pulses with precisely defined width, and transmits these control pulses to the servo motors.

In the following subsections we will elaborate on these two components of the control system.

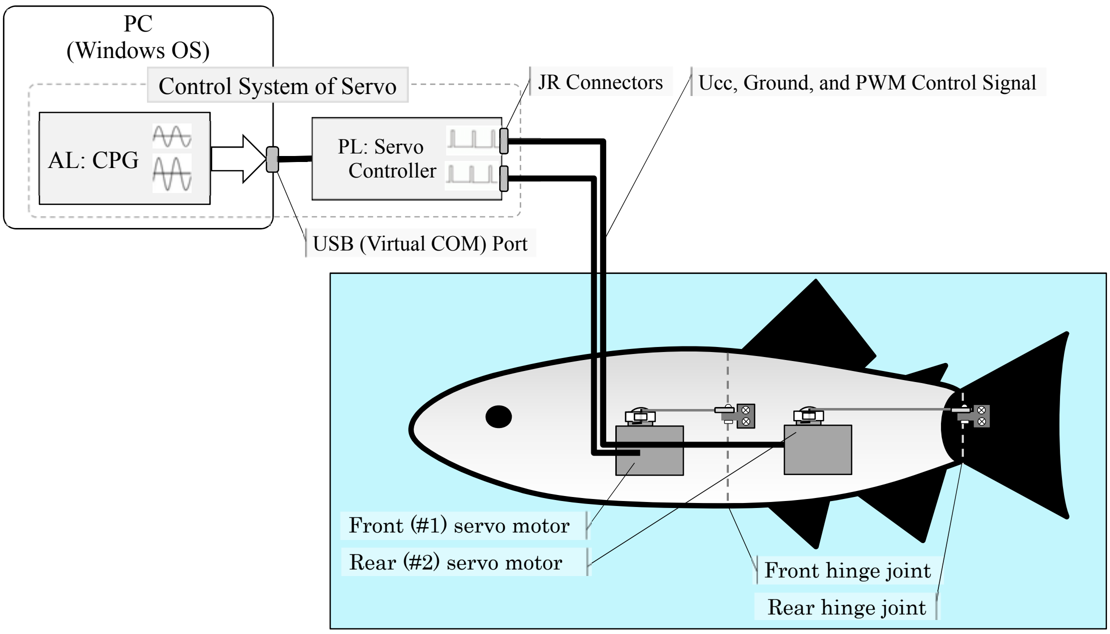

Figure 7. Servo control system comprising two layers: abstract layer (AL), represented by a central pattern generator (CPG) and physical layer (PL) – servo controller.

3.2. Abstract Layer: the Concept

The way we define the intended turning angles of servo is inspired by the central pattern generators (CPG) in nature [6]. CPG is a neural circuit in the central nervous system of animals that generates the rhythmic (periodical) patterns of locomotion in walking, crawling, swimming and flying. Consistent with the hypothesis, first expressed by P. Miturich, that the such a locomotion in nature features a wave-like (i.e., sinusoidal) pattern, in our implementation, CPG calculates at discrete instants of time – with sampling interval of 20 ms – the target turning angles α_1 and α_2 of the two servo motors of the fish robot as follows:

$$\alpha_1 = A_1 \,.\, sin(2.\pi.f.t) \tag{10}$$

$$\alpha_2 = A_2 \,.\, sin(2.\pi.f.t+\beta) \tag{11}$$

where

α_1 and α_1 are the target turning angles of front and rear servo motors, respectively,

A_1 and A_1 are the angular amplitudes of undulation of the front- and rear servo motors, respectively,

f is the frequency of oscillation,

β is the phase shift between the undulation of the two servo motors, and

t is the discrete instant of time (the sampling period is 20 ms) in seconds: 0 s, 0.020 s, 0.04 s, 0.06 s, etc.

In order to ensure a coordinated movement of the two joints of the robot, we stipulate that they undulate with the same frequency f. On the other hand, the target turning angles of the servos that undulate the two joints are calculated independently according to the concrete values of the remaining parameters in Equations (10) and (11) – A_1, A_2, and β. The main loop fog CPG is shown in Algorithm 1.

```
Start_of_sampling_interval:= Now;
for t=0 to tmax increment 20 ms do begin
   α1 = A1.sin(2.π.f.t);
   α2 = A2.sin(2.π.f.t+β);
   repeat until (Now - Start_of_sampling_interval) >= (20 ms - tSEND);
   Send_to_PL(α1,α2);
   Start_of_sampling_interval:= Now;
end;
```

Algorithm 1. Main loop of CPG

In order to tune the CPG, and consequently – to optimize (from the viewpoints of trust of propulsion, speed, or energy efficiency) the undulation of the fish robot, we provide the CPG with four "dials" that allow to discretely adjust the values of the four parameters of undulation – A_1, f, A_2, and β, as shown in Table 4. Notice that the angular amplitudes of undulation of servo motors A_1 and A_2 are scaled down to reduced amplitudes of undulation of the joints of the robot. The scaling coefficient is equal to the ratio of the length of the servo arm (L_{SA}) to the length of the control horn (L_{CH}). For the values of these parameters of 6 mm and 9 mm (as mentioned earlier in Sections 1 and 2), respectively, the scaling coefficient is equal to 2/3. Therefore, the amplitude of undulation of servo motors π/4 radians (45 degrees) would result in corresponding amplitude of undulation of segments of π/6 radians (30 degrees).

Table 4. Tuning of main parameters of undulation in CPG

Parameter	Unit of measure	Range of the coded discrete (integer) value	Range of actual value pertinent to servo motor	Corresponding value pertinent to the joint of the robot	Discretization step
A_1	radians	[0..10]	[0 .. π/4]	[0 .. π/6]	π/60
f	Hz	[1..12]	[0.2..2.4]	[0.2..2.4]	0.2
A_2	radians	[0..10]	[0 .. π/4]	[0 .. π/6]	π/60
β	radians	[0..10]	[0 .. π]	[0 .. π]	π/10

The number of combinations of the discrete values of the four parameters of undulation, and consequently – the size of the space S that should have been explored in the search for optimal values of these parameters – could be calculated as follows:

$$S = N_{A1}.N_f.N_{A2}.N_\beta = 11.12.11.11 = 15{,}972 \qquad (12)$$

where N_{A1}, N_f, N_{A2}, and N_β are the numbers of discrete values of A_1, f, A_2 and β, respectively.

Assuming that the "brute force" approach of complete enumeration followed by testing of each of these combinations on the real robot is unfeasible, we would rely on a heuristic approach based on evolutionary computations, namely – genetic algorithms, as we elaborate later in Section 4. The snapshot of the CPG is shown in Figure 8. The tunable four parameters of undulation are shown in the group entitled "Genotype obtained from XGP".

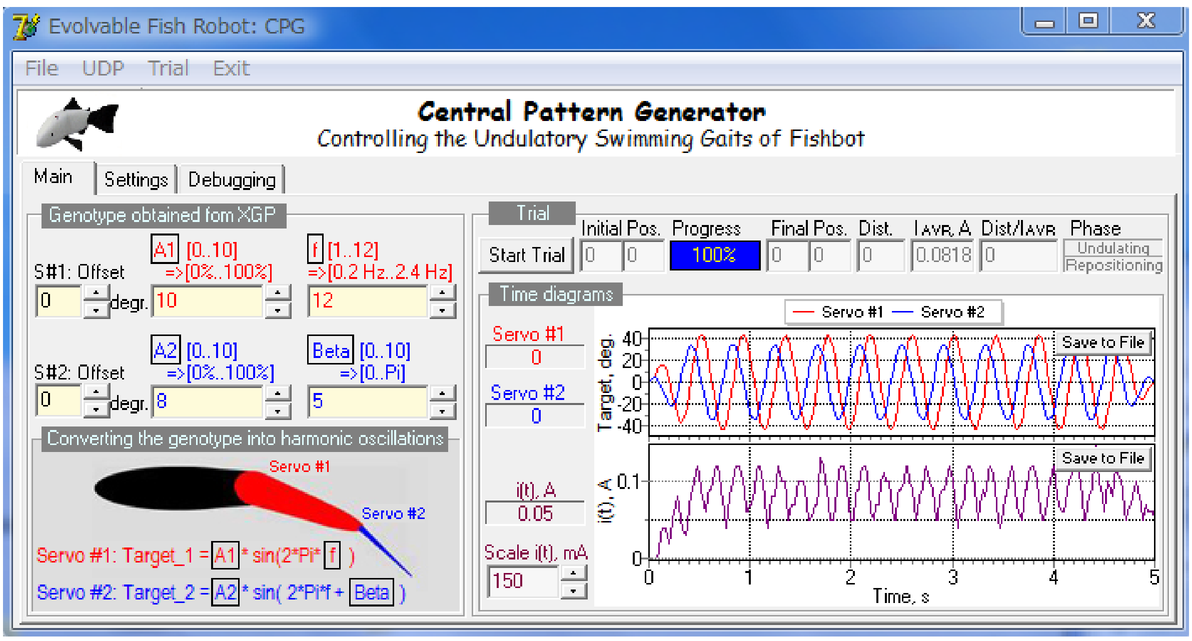

Figure 8. Snapshot of CPG

3.3. Abstract Layer: Implications on the Operational Reliability of the Robot

In section 2 we estimated the amount of inertial forces. Assuming a similar amount of drag forces of water, and a safety margin of 40% we selected the servo motor that could sustain the sum of these forces. However, in addition to the above mentioned forces, there are additional forces, that we have not accounted so far. These forces, as we will elaborate below, stem from the specifics of controlling the servo motor via the computer software.

Avoiding Jitter of Servo Motors

Jitter is a phenomenon that manifests itself in an uneven, jerky movement of the servo. For the computer-controlled servo, one of the main reasons for jitter to occur is the significant irregularity in the actual duration of the sampling interval. Assuming that the runtime of PL – due to its hardware implementation – is negligible, we would like to focus on the irregularities in the timing of issuing the commands (instructing the servo motor to turn to a target turning angle) by AL (i.e., CPG) to the PL. As each of these commands instructs the servo to turn to a given target angle, that is calculated based on the ideal (modeling) sampling interval, any deviation of the actual sampling interval from the ideal one would result in setting a value that does not correspond to the value pertinent to the actual timing of command, as illustrated in Figure 9.

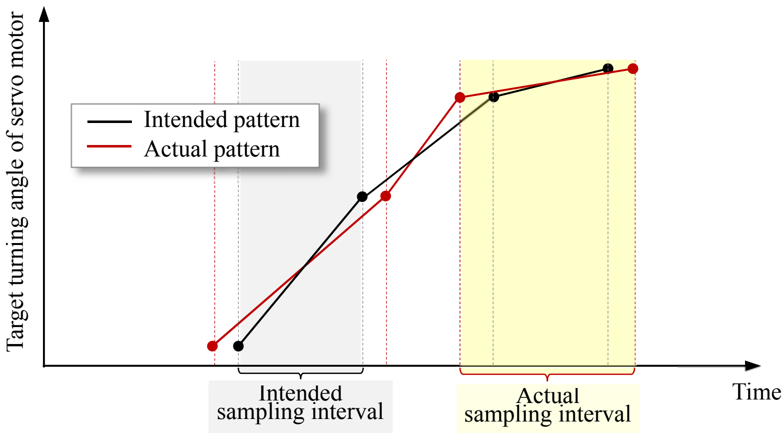

Figure 9. Deviation of the duration of actual sampling interval from the intended one results in the deviation of the actual pattern of dynamics of turning angle of servo from the intended one.

In order to guarantee that the actual sampling interval is as close as possible to the intended (ideal) one, in the implementation of function `Now` in CPG (Algorithm 1, lines #1 and #7) we invoked the Windows OS function `QueryPerformanceCounter`. The latter returns a timestamp with sufficiently high resolution (less than 1 microsecond). Moreover, in order to minimize the detrimental effect of the eventual interruptions (due to multitasking of the OS) of the main loop of CPG (Algorithm 1) on the variability of the actual sampling interval, we assumed that the sampling interval begins immediately after sending the command to the HL (Algorithm 1, line #7), rather than with the first command of the main loop. Consequently, all interruptions would result in delays that will be accounted for by the "barrier" (Algorithm 1, line #5), waiting for the expiration of the intended sampling interval. The experimentally obtained dynamics of the duration of actual sampling interval is shown in Figure 10. As illustrated in the figure, the standard deviation, maximum, and minimum durations of the sampling interval are 0.017 ms, 20.023 ms (+0.11%), and 18.861 ms (-0.69%), respectively. Moreover, we observed the latter (minimum) value only during the initial few sampling intervals. The exact reason why this happens (and, indeed, it happens in *all* the experimental runs of the robot), is unclear to us. However, we assume that both (i) the initial, lower than intended values of the sampling interval, and (ii) the fluctuations of the latter are related to the *variable* runtime of sending of the target turning angles of both servo motors to PL (Algorithm 1, line #6). The target turning angles are sent as a transmission of four bytes to the COM port associated with the PL, and the runtime of this operation is assumed to be a *constant* (denoted in Algorithm, line 5, as t_{SEND}) representing the mean runtime of such a transmission. Nevertheless, the deviations of the actual sampling interval from the actual one are negligible, and visually we could confirm that jitter does not occur in the movement of servo motors of the fish robot.

However, we could not guarantee that a software interruption (with unaccounted runtime) due to multitasking of the OS would not happen just immediately before- or after the transmission of the command to PL (Algorithm 1, either between lines 5 and 6, or between lines 6 and 7), and therefore – an incidental jerky movement of the servo motor would not occur. In order to dampen the effect of such a jerky movement, in accordance with the concept of series elastic actuators [8] we implemented the push rod of both servo motors as an improvised *omega-shaped compliant spring* (Figure 1).

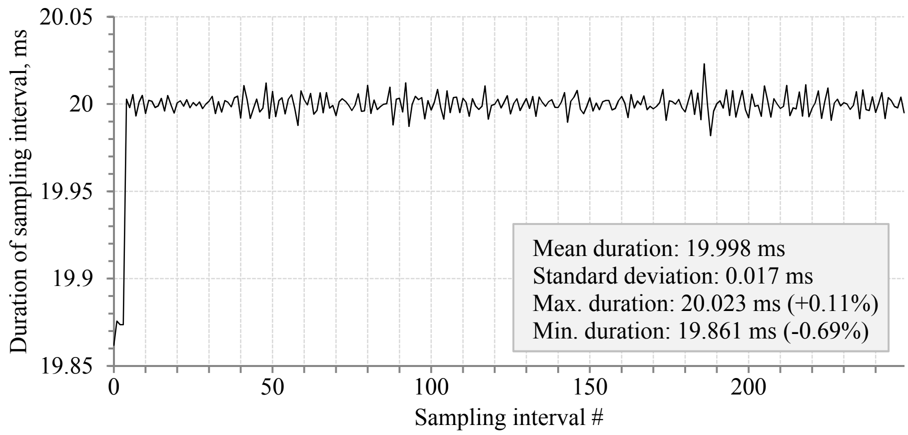

Figure 10. Dynamics of the actual duration of sampling interval during 5 s trial of the robot.

Avoiding Excessive Transient Torque Acting upon the Servo Motors

At the startup of the trial the two servo motors have to commence the undulation of the segments of the robot in accordance with the Equation (3), implemented by the lines #3 and #4 of the pseudocode shown in Algorithm 1. The sudden start of undulation (from standstill position of the robot) results in a significant acceleration, which, in turn, results in significant torque acting upon the servo motors. For similar reasons, the sudden stopping of the undulation at the end of the trial would also result in significant torque acting upon the servo motors. The results of the numerical simulations, shown in Figure 11(a) and Figure 11(a) suggest that for f=1.6 Hz, and an amplitude of undulation A_1=A_2=π/4 radians, the values of the transient acceleration (and consequently, according to the Equation (9) – the resulting torque) would assume a shape of a *delta function* [20] with a value which is about 10 times higher than that of the normal, continuous operation of servo motors.

As an approach to reduce the excessive *transient acceleration*, we implemented a warmup and cooldown phases (0.5 s each) at the beginning and ending of the trial, respectively. During the warmup phase the amplitude of undulation gradually, and nonlinearly increases from 0 to the intended value. This is accomplished by multiplying the amplitude of oscillations A_1 and A_2 (Algorithm 1, lines #3 and #4) to a *warmup*- and *cooldown* scaling coefficient K_{WC}. The value of the coefficient is time-dependent, and it is calculated according to Equation (13) as a sin function with a frequency f_{WC}=0.5 Hz (i.e., a period of 2 s). The time-dependent value of the scaling coefficient K_{WC} is shown in Figure 12.

$$K_{WC} = \begin{cases} sin(2.\pi.f_{WC}.t), & t = 0 \sim 0.5 \text{ s, increment } 0.02 \text{ s} \\ 1, & 0.5 \text{ s} < t < 4.5 \text{ s} \\ sin(2.\pi.f_{WC}.(t-4)), & t = 4.5 \text{ s} \sim 5 \text{ s, increment } 0.02 \text{ s.} \end{cases} \quad (13)$$

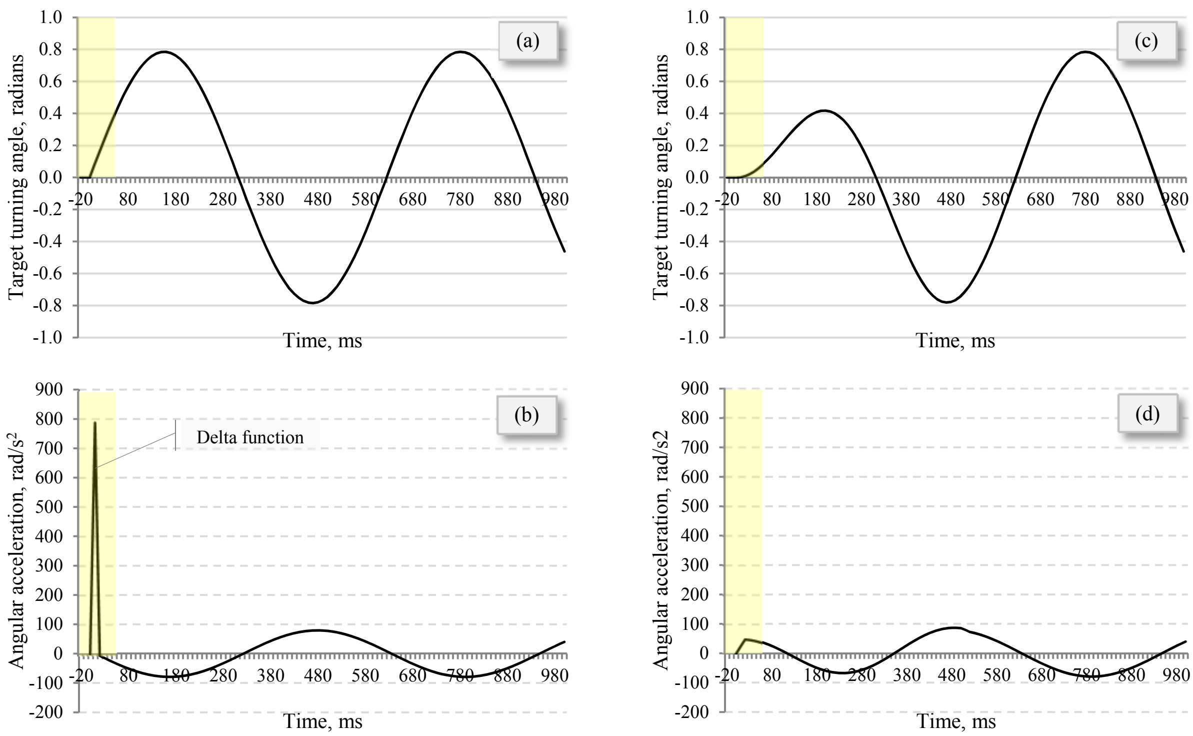

Figure 11. Transient target turning angle and angular acceleration of servo motors on startup of the trial without- (a, b) and with (c, d) gradual increase (warmup) of amplitude of undulation for A_1=A_2=π/4 radians, f=1.6 Hz, and duration of sampling interval 20 ms.

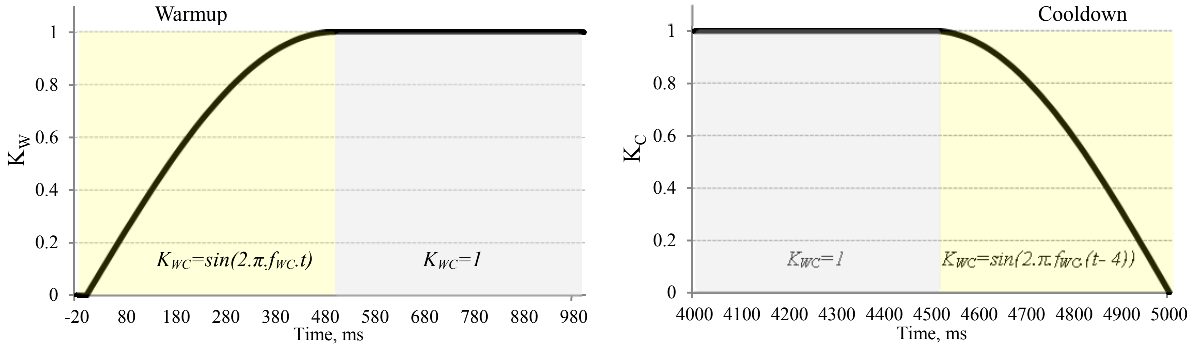

Figure 12. Time-dependent value of the amplitude scaling coefficient K_{WC} during the warmup (left) and cooldown (right) phases of the trial, respectively.

The results of numerical simulations (Figure 11, right) indicate that the proposed warmup and cooldown phases of the trial result in the values of transient accelerations do not exceed the maximum accelerations occurring during the normal (continuous) operation. The warmup and cooldown phases for undulating frequency f=1.6 Hz, phase shift β=π/2, and amplitudes of undulation A_1=A_2=π/4 of both servo motors are illustrated in Figure 13.

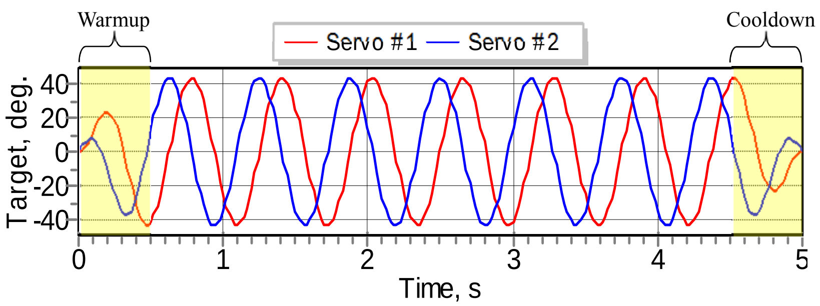

Figure 13. Dynamics of target turning angles of both servo motors during a sample 5 s trial. The parameters of oscillations are as follows: amplitudes A_1=A_2=π/4, phase shift β=π/2, and frequency f=1.6 Hz

3.4. Physical Layer

The hardware component of the Servo Control System – PL, as illustrated in Figure 7 – receives a series (in 20 ms interval) of commands from CPG about the target turning angles (Algorithm 1, lines #3, #4, and #6) of the two servo motors of the robot. The hardware of PL interprets these commands by transmitting control pulses with a corresponding width (duration) between 1.0 ms (corresponding to the minimal turning angle of -55 degrees) and 2 ms (for the maximum turning angle of +55 degrees) to the two servo motors.

For the implementation PL we adopted Micro Maestro 6-Channel USB Servo Controller produced by Pololu (Figure 13). It features a compact design, a native USB interface, and can control up to six servo motors via a high-resolution (0.25 μs) PWM.

Figure 13. Micro Maestro 6-Channel USB Servo Controller

3.5. Discussion

As we elaborated earlier, we developed the controller of the servo as a system comprising two layers – AL, implemented as CPG that periodically calculates (with sampling interval of 20 ms) the time series of the values of target turning angles of two servo motors of the robot, and PL, which receives these values, and transmits controls pulses with corresponding width (duration) to the servo motors. The rationale of such decomposition of the control system, and the introduction of the additional PL was the intention to obtain a high-resolution, jitter-free control of the servo motors of the robot.

In principle, it could have been possible to implement the functionality of PL – converting the value of turning angle into a pulse with corresponding duration – in the software as a part of CPG. However, as we illustrated, due to both the multitasking of the OS, and non-determinism of the runtime of communication with the USB port, the intended duration of sampling interval of 20 ms fluctuates with a standard deviation of 0.07 ms. While such a deviation is a very small part of the sampling interval (about 0.35%), for the duration of control pulses between 1 ms and 2 ms, the deviation of 0.07 ms would be prohibitively high – 7% and 3.5% respectively. Therefore, the introduction of PL yields at least 10 times higher precision of timing of control signals sent to the robot.

Considering the sinusoidal pattern of the target turning angles of undulation, generated by AL, we would like to note that any rhythmic (periodical) pattern would have resulted in an adequate control of the robot. Moreover, a carefully tuned (e.g., via genetic programming) non-sinusoidal signals might result in even better (e.g., faster, or more energy-efficient) swimming gaits compared to that of the sinusoidal, wave-like patterns [1, 7, 15]. However, the customization of the eventual non-sinusoidal patterns would definitely require an increased number of adjustable parameters, which would increase the search space of optimization algorithms. In case of applying genetic programming to evolve the optimal (non-sinusoidal) patterns would imply that we would deal with a functional-, rather than a parametrical optimization problem, with the former featuring a much larger search space. In addition, it would have been more difficult to estimate analytically the resulting torques that act upon the servo motors, and therefor – the choice of appropriate servo motors would have been much more challenging.

4. Evolutionary Framework: GA Manager

The search for optimal swimming gaits – i.e. gaits featuring maximum thrust, maximum speed, or maximum energy efficiency – could be paraphrased as a search for such values of the four main parameters – A_1, f, A_2 and β – that are used by CPG to control the undulation of the bot. In our approach, for the search of these values of parameters we have relied on GA as an unsupervised learning method inspired by natural selection. In GA, the optimal values of the four parameters would evolve as a genotype (chromosome) over many generations produced by the artificial evolutionary system by applying a selection of the best performing genotypes, and main genetic operations – crossover and mutation. In this section we shall elaborate on the main components of GA – the genetic representation, genotype-to-phenotype mapping, selection, crossover, mutation, and fitness evaluation.

4.1. Genetic Representation

The chromosome C of the GA consist of the following four *alleles*, representing the coded values of four main parameters used by CPG to control the robot:

$$C = \langle A_{1C}, f_C, A_{2C}, \beta_C \rangle \tag{14}$$

The discretization and the range of the coded values A_{1C}, f_C, A_{2C}, β_C are exactly the same as shown in Section 3, Table 4. We decided the discretization steps of the parameters from the standpoint of the tradeoff between the size of the search space and the resolution (precision) of representation of parameters.

4.2. Algorithm of GA. Genetic Operations

The algorithm of GA is shown in Algorithm 2. The main parameters of GA are given in Table 5.

```
1.  Create the initial population of chromosomes;
2.  Evaluate the population;
3.  while (not Termination Criterion) do begin
4.    Selection;
5.    Reproduction (Crossover);
6.    Mutation;
7.    Evaluate the population;
8.  end;
```

Algorithm 2. Algorithm of GA

Creation of Initial Population

The initial population is created (Algorithm 2, Line 1) by randomly generating the value of each of the four alleles in chromosomes as an integer within the corresponding range as described in Section 3, Table 4.

Selection

We employed a *binary tournament selection* (Algorithm 2, Line 4) – a computationally efficient selection that offers a good tradeoff between the fitness convergence and the diversity of population. The selected chromosomes comprise the mating pool.

Table 5. Main parameters of GA

Parameter	Value
Population size, chromosomes	14
Selection mechanism	Binary tournament
Selected as an elite, chromosomes	2
Size of the selected mating pool (including the elite) , chromosomes	6
Ratio of selected chromosome in population,	0.43
Crossover mechanism	Single point
Number of offspring produced by crossover, chromosomes	8
Ratio of offspring in population	0.57
Mutation mechanism	Single point
Probability of mutation	0.05
Termination criterion	#Generations = 20

Reproduction (Crossover)

The crossover operation (Algorithm 2, Line 5) picks randomly two parent chromosomes from the mating pool, and produces two offspring chromosome that are being inserted into the new population. We implemented an one point crossover in that only the value of one (random) allele is exchanged between the parent chromosomes

Mutation

The mutation operation (Algorithm 2, Line 6) alters randomly – with a given probability – a single, random allele from the offspring, produced by crossover operation.

Termination Criterion

The termination criterion (checked in Algorithm 2, line 3) is the number of generations equal to 20. Because do not know *a priori* the best values of fitness, we do not include the fitness value as a termination criterion. Totally, 180 trials of the robot would be needed to evolve the optimal swimming gaits over 10 generations. Assuming that – due to the non-determinism of GA – we need to run the latter 10 times in order to obtain the value of parameters that indeed, result in an optimal swimming gait, we would need 1800 trials of the robot. This number is just about 11% of the number of trials that we would have implemented in an eventual brute force approach.

Fitness Evaluation

The evaluation of fitness (Algorithm 2, Lines 2 and 7) depends on the objective of optimization – the thrust, speed, or energy efficiency of the robot. It is implemented by trials on the swimming robot, where the latter is being controlled by CPG with the four main parameters – A_1, f, A_2, β – set by decoding the values of corresponding alleles of the currently evaluated chromosome. We will elaborate on the fitness evaluation of the fish robot, evolved for maximum thrust in Section 5.

Evolutionary Framework: GA Manager

The managing of population of chromosomes, selection operation, and main genetic operations – crossover, and mutation are relatively domain-neutral. In our approach these operations are implemented by the GA Manager – a GA variant of the versatile XML-based evolutionary framework [14]. The snapshot of the GA Manager is shown in Figure 14.

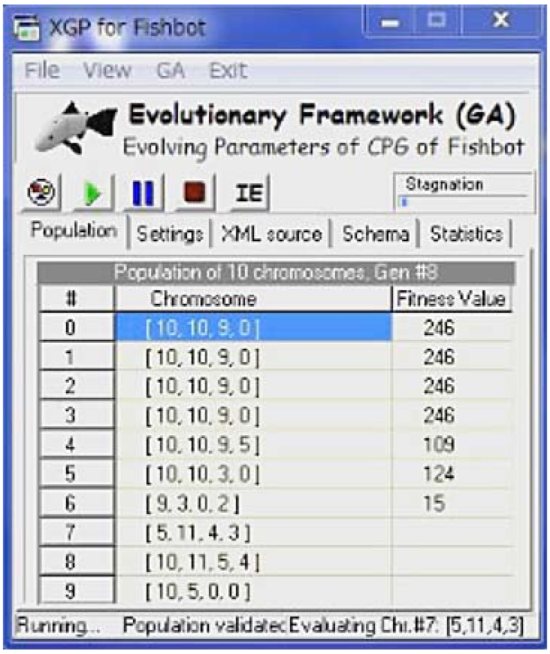

Figure 14. Snapshot of the GA Manager: managing the population of chromosomes, and implementing selection, crossover and mutation.

4.3. Discussion

The search for such values of the four main parameters of undulation – A_1, f, A_2 and β – in principle, could be accomplished by "brute force" – i.e., by complete enumeration of all possible combinations. As expressed in Section 3, Equation (12), the possible number of combinations for the given discretization of the four parameters of undulation, is about 16,000. Assuming the trial of a given combination of parameters takes 20 s (including the time needed to reposition the robot after the trial – i.e., to move it to its initial position), the total runtime would be about 89 hours. Thus, we consider such an approach as impractical. Another way to find the values of parameters that result in optimal swimming gaits is to employ supervised machine learning, which, in turn, implies that the set of good values of parameters are given and the respective evaluation of their quality (e.g., the value of thrust, speed, or efficiency) is provided by human in advance. This approach is unfeasible either, because we assume that neither the good values of parameters, nor the human annotated quality are *a priori* known.

5. Evolution of Maximum Thrust: Experimental Setup and System Configuration

In this section we will describe the experimental setup and the system configuration of the evolution (via GA) swimming gait featuring a maximum thrust.

5.1. Experimental Setup

Evaluating the Thrust Generated by Swimming Fish Robot

In the evolution of swimming gaits featuring maximum thrust we employ the GA framework – GA Manager – as described above in Section 4. Also, as we explained in the same section, the fitness of each chromosome of GA is evaluated based on the performance of the real robot, controlled by CPG, where the latter features the values of the four main parameters of undulation – A_1, f, A_2, β – being set from by decoding the values of corresponding alleles of the evaluated chromosomes. The question, that we have not answered yet, is: *how to evaluate the performance of the swimming fish robot, i.e., the amount of thrust generated by the latter?*

One possible approach, used by researches on physical fish robots, is to use a pressure sensor, and to allow the tip of the swimming robot to press a mechanism that is mechanically connected to the sensor. One of the advantages of this approach is that it is static – and consequently – it is fast because there is no need for the time-consuming reposition of the robot to its initial position after each trial. However, we assume that the physical contact between the tip of the robot and the mechanisms connected to the pressure sensors would inevitable alter the gait of the robots, and consequently – would compromise the realism of the evaluation of the generated thrust.

Therefore, in our research we proposed an approach of contact-less measurement of the thrust. We suspended the fish robot on a steel wire that rotates around a given pivot, as illustrated in Figure 15.

The initial position of the robot (at the start of the trial) is directly under the pivot. As the robot, controlled by CPG (tuned by the given values of parameters encoded in the alleles of evaluated chromosome) it swims away from its initial position. Swimming away would gradually increase the longitudinal (tangential) component F_{ST} of sinking force F_S, which, at some (equilibrium) position, would become equal to the generated thrust F_T:

$$|F_T| = |F_{ST}| \tag{15}$$

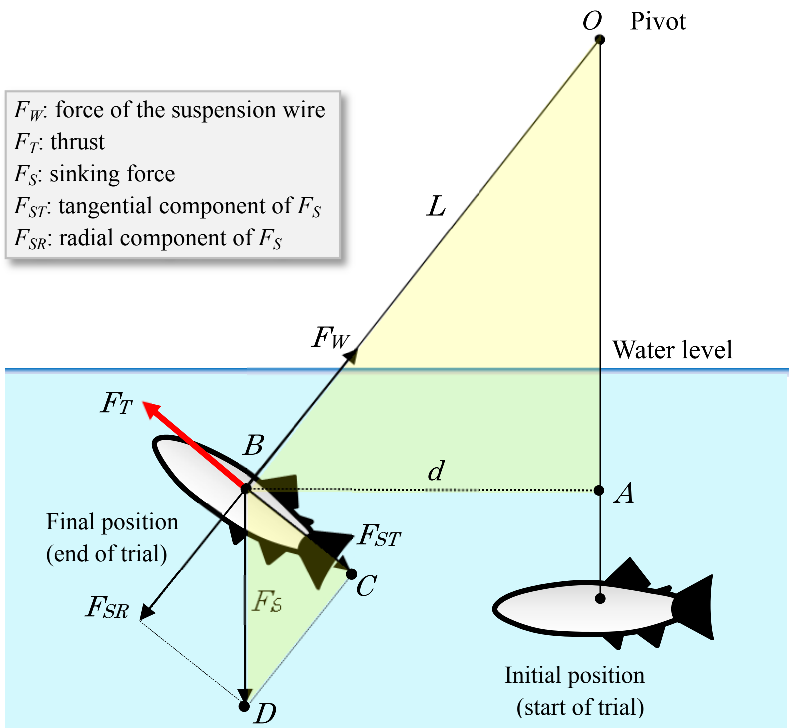

Figure 15. Balance of forces acting upon the fish robot at the end of the trial

The relationship between the traveled distance and the generated trust could be obtained from the similarity of triangles ΔBAO and ΔBCD:

$$BC/BD = BA/BO \tag{16}$$

Therefore,

$$BC = BD.BA/BO \tag{17}$$

Taking into consideration that $BC = |F_{ST}|$, $BD = |F_S|$, $BA=d$, and $BO=L$ (the length of suspension wire), the Equation (17) could be rewritten as

$$|F_{ST}| = (|F_S|/L).d \tag{18}$$

Because, as expressed in Equation (15), at the equilibrium point the tangential component of sinking force $|F_{ST}|$ is equal to the thrust force $|F_T|$ we could infer that

$$|F_T| = (|F_S|/L).d = k_T.d \tag{19}$$

Where, for sinking force $|F_S|$= 0.098 kg (as shown in Table 1), and L=1.9 m, the value of coefficient k_T is 0.052 kg/m. Therefore, we could conclude that the amount of the generated thrust is directly proportional to the distance between the initial- and equilibrium (final) positions of the fish robot, i.e., the distance robot travels during the trial. Thus, we decided to use this distance as a contactless metric for the evaluation of the thrust (i.e., the fitness) of the swimming fish robot.

The duration of the trial during the fitness evaluation is set to 5 s. We verified experimentally that, regardless of the amount of generated thrust, this duration is sufficient for the robot to reach its equilibrium point at the end of trial.

Measuring the Traveled Distance

We measured the distance robot travels during the trial by a tracking its position optically via a video camera mounted at a height of about 1 m above the water level. The principal optical axis of the camera is set to be perpendicular to the intended trajectory of the swimming robot. The tracker obtains – with a sampling frequency of 30 frames per second – the 2D coordinates (in pixels) of the bright LED (attached to the fish robot) at the initial and final (equilibrium) positions of the robot and transmits the coordinates of these two positions to the CPG. The latter calculates – and transmits it to the GA Manager – the raw fitness of the evaluated chromosome as Euclidian distance (in pixels) between these two positions. We consider this fitness as a raw one, because neither the scaling coefficient k_T (Equation 19) nor the scale of the visual field of the camera (that maps the pixels into appropriate physical metrics) are being considered. The snapshot of the Optical Tracker is shown in Figure 16. We shall explain algorithmically the interaction between the three components of the system – GA Manager, CPG, and Optical Tracker – in the subsection 5.2 System Configuration.

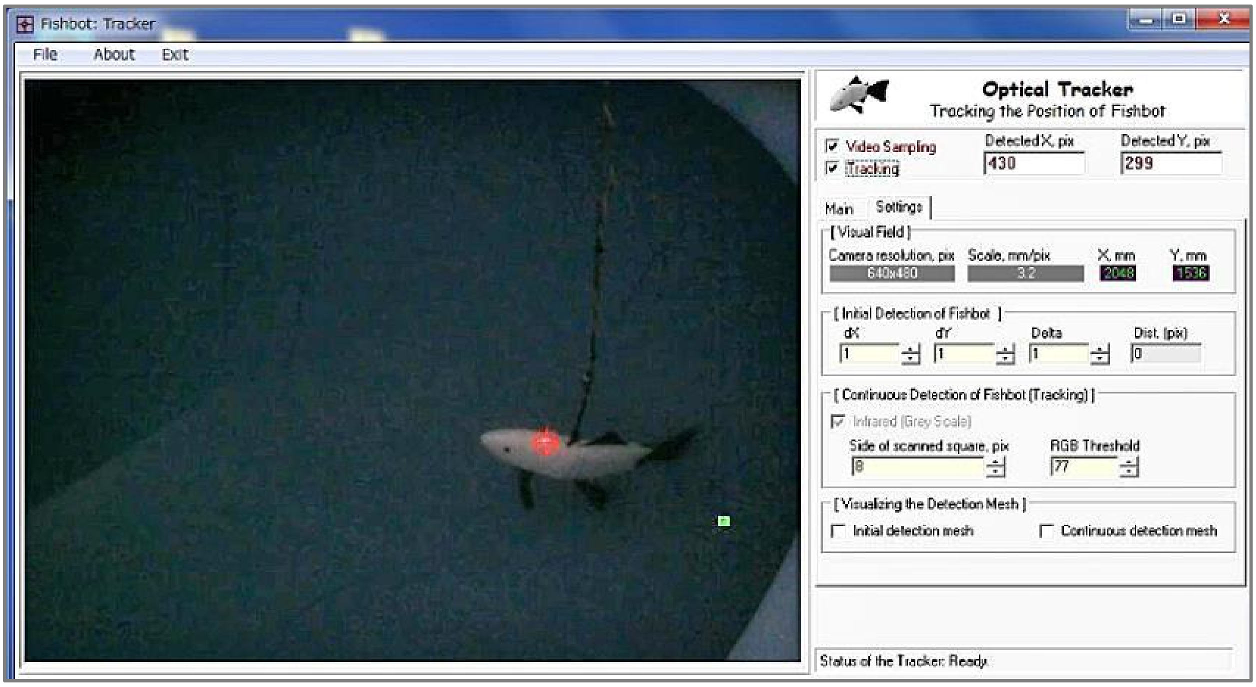

Figure 16. Snapshot of the Optical Tracker of fish robot

Repositioning the Robot after the Trial

In order to ensure equal conditions for the evaluations of chromosomes, we have to ensure that after each trial, the fish robot is re-positioned to almost (exactly) the same initial position before the next trial commences. The same initial position implies that at the beginning of each trial the robot should be standstill, and both the location and orientation of the robot should be (nearly) identical. In our approach, we relied on the tangential component F_{ST} of sinking force F_S to "pull" the robot to its initial position (Figure 15). However, as the robot moves to its initial position, in addition to the pulling force, it is subjected to two additional forces: torsion force of the suspension wire and hydrodynamic drag force. As the natural fish have evolved with vertical fins – acting as yaw stabilizers – in the rear part of their bodies, the center of the overall drag forces is behind their center of gravity. Similarly, the center of drag forces is behind the suspension point (i.e., the center of pulling force). Therefore, the backward movement of the robot results in a torque that attempts to rotate the robot around its yaw axis, i.e., around the suspension point. This rotation yields a deviation from the shortest, straight line trajectory between the final and the targeted initial positions of the robot. At some yaw angle the torque cause by drag forces will be balanced by the torsion force of the suspension wire, and this equilibrium will limit the yaw angle of backward moving robot. As the robot approaches the initial position, the pulling force will gradually decrease, causing a corresponding decrease of the speed of robot. This, in turn, will result in the gradual decrease of the drag forces and the yaw torque they create. The robot will slowly return to the initial location, assuming its initial orientation. The repositioning of the robot is illustrated in Figure 17 (left).

We empirically verified that such a repositioning takes prohibitively long time. The primary reason for such a slow repositioning is the significant deviation – from the ideal ones – of both the yaw angle and trajectory of the repositioning robot. Both deviations were caused by the presence of yaw torque as a result of offset drag forces acting upon the backward moving robot. In order to address this problem, we designed passively-moving *pectoral fins*, as illustrated in Figure 18. The fins are rotating around an axis that is orthogonal to the yaw axis of the robot. The axis of fins is offset relative to the center of gravity and center of drag of the fins. In forward moving robot, the fins assume a horizontal position and do not affect the swimming gait of the robot. During repositioning, as a result of drag forces acting upon the fins, the latter assume a vertical position, resulting in significant drag forces. These forces are sufficient to move the center of overall drag forces towards the tip of the robot, and crucially – in front of the suspension point. Consequently, the hydrodynamic drag would create a stabilizing torque that would keep both the deviation of the yaw angle and trajectory form ideal ones to the minimum. We have estimated experimentally, that the waiting time T_R (in seconds) for repositioning, calculated according to the following Equation (20) is adequate and allows the robot to assume its initial position in all of the variations of its final position:

$$T_R = 3 + 4 \,.\, d/100 \qquad (20)$$

Where d is the raw fitness value – the distance (in pixels) between the initial and final position of the robot. The repositioning with movable pectoral fins is illustrated in Figure 17 (right).

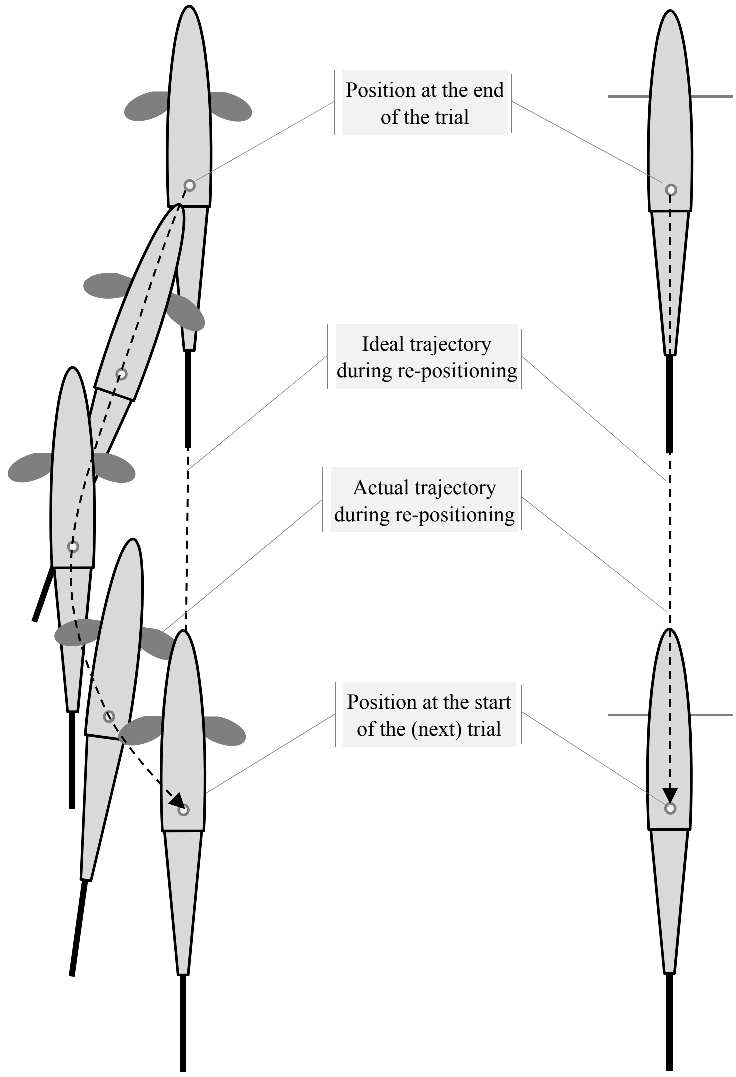

Figure 17. Repositioning of the robot with static (left) and movable (right) pectoral fins

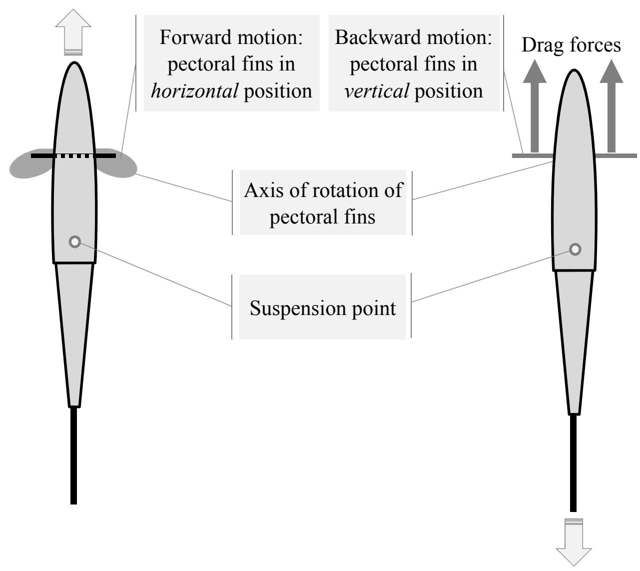

Figure 18. Position of the movable pectoral fins during forward (left) and backward (right) locomotion, respectively.

5.2. System Configuration

The software implementation of the evolutionary system used to evolve the swimming gaits featuring maximum thrust consists of the following three subsystems:

- GA Manager: manages the population of chromosomes that encode for the four main parameters of undulation. Performs selection and the main genetic operations – crossover and mutation. The evaluation of fitness of each chromosome is delegated (via UDP) to the second subsystem – CPG.
- CPG: Receives a chromosome from GA Manager, "tunes" the four main parameters of undulation from the decoded chromosome, and controls the undulation of the fish robot during the trial. Immediately before and after the trial requests (via UDP) the 2D coordinates (in pixels) of the initial- and final position of the robot, respectively, from the third subsystem – Optical Tracker, calculates the raw fitness of the chromosome as an Euclidean distance between these two coordinates, and – after waiting for the repositioning of the robot – transmits the raw fitness value to GA Manager.
- Optical Tracker: tracks – with a sampling frequency of 30 frames per second – the 2D coordinates of the current position of the robot in the 2D visual of video camera, and transmits (via UDP) these coordinates to CPG upon a request from the latter. From the viewpoint of GA Manager, CPG and optical tracker could be seen as a single entity – fitness evaluator.

The block diagrams of these three subsystems are shown in Figure 19. Functional modules that are involved in communication between the subsystems are shown in gray color. The snapshot of the three subsystems, obtained during the evolution of the main parameters of undulation that result in maximum thrust, is shown in Figure 20.

5.3. Discussion

The estimation of the thrust by video tracker implies that the measured traveled distance of the robot from the initial position to its position of equilibrium is proportional to the generated thrust. However, the relationship is not exactly linear, because the distance is measured by video camera. Therefore, there measurement would suffer from perspective distortion as the robot moves across the visual field of the camera. Moreover, there would be distortion in the actual position of the fish robot due to the refraction of light that occurs at the boundary of water and air. Therefore, it would be correct to note that the estimation of the thrust is not exact, but rather – a heuristical one. The important aspect of such *heursitical estimation* is that it is admissible, i.e., it never overestimates the actual traveled distance, and therefore – the actual thrust being generated by the robot. Consequently, from two gaits the gait that features higher actual thrust will always feature a higher – yet slightly distorted – value of its heuristical estimation. This is crucial in the context of fairness of the adopted binary tournament selection in GA: the chromosome-winner of the tournament, based on the heuristical estimate of the thrust, will always be the one that, indeed, features a higher value of actual thrust.

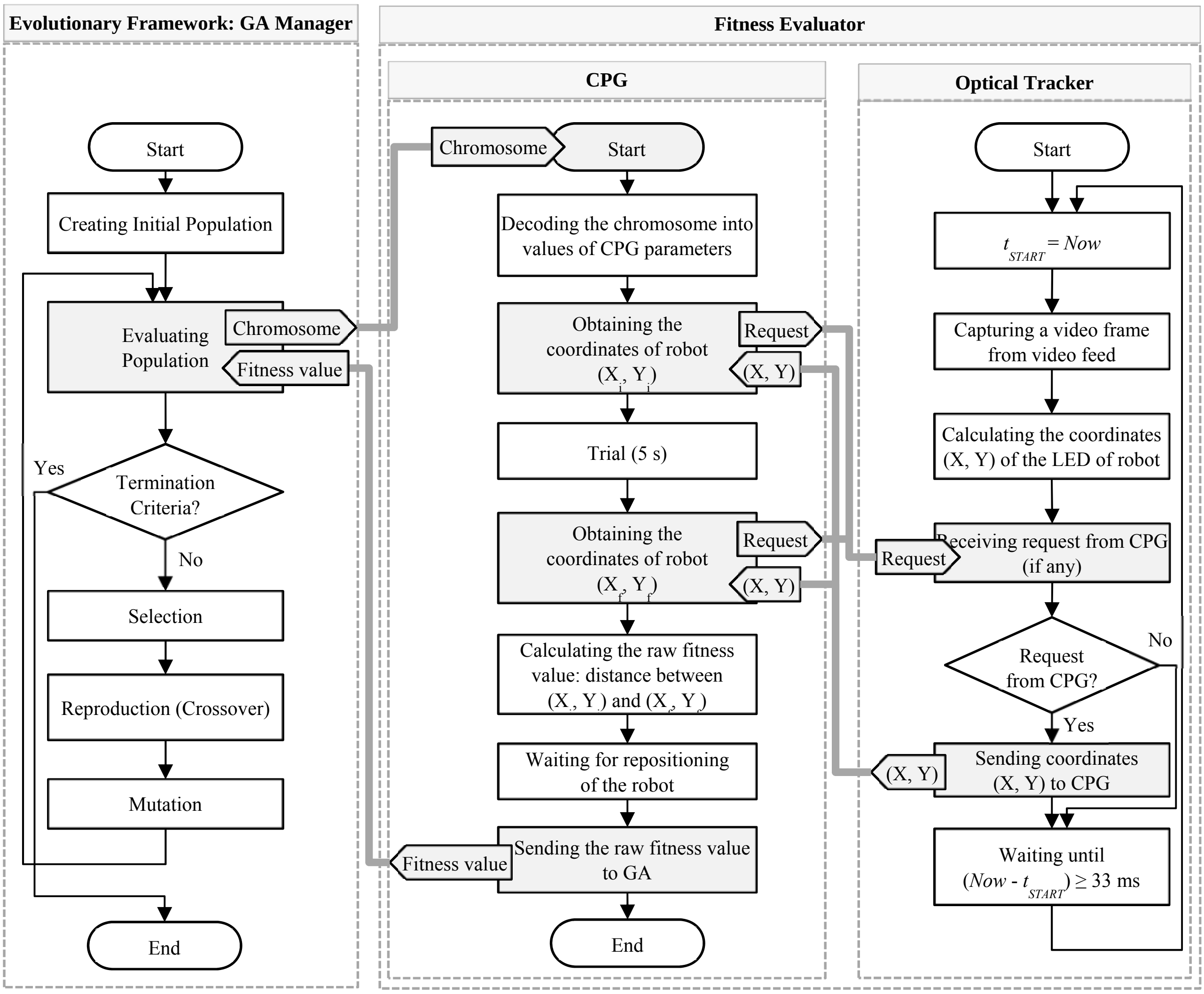

Figure 19. Block diagram of the three software subsystems: Evolutionary framework (GA Manager), CPG, and Optical Tracker.

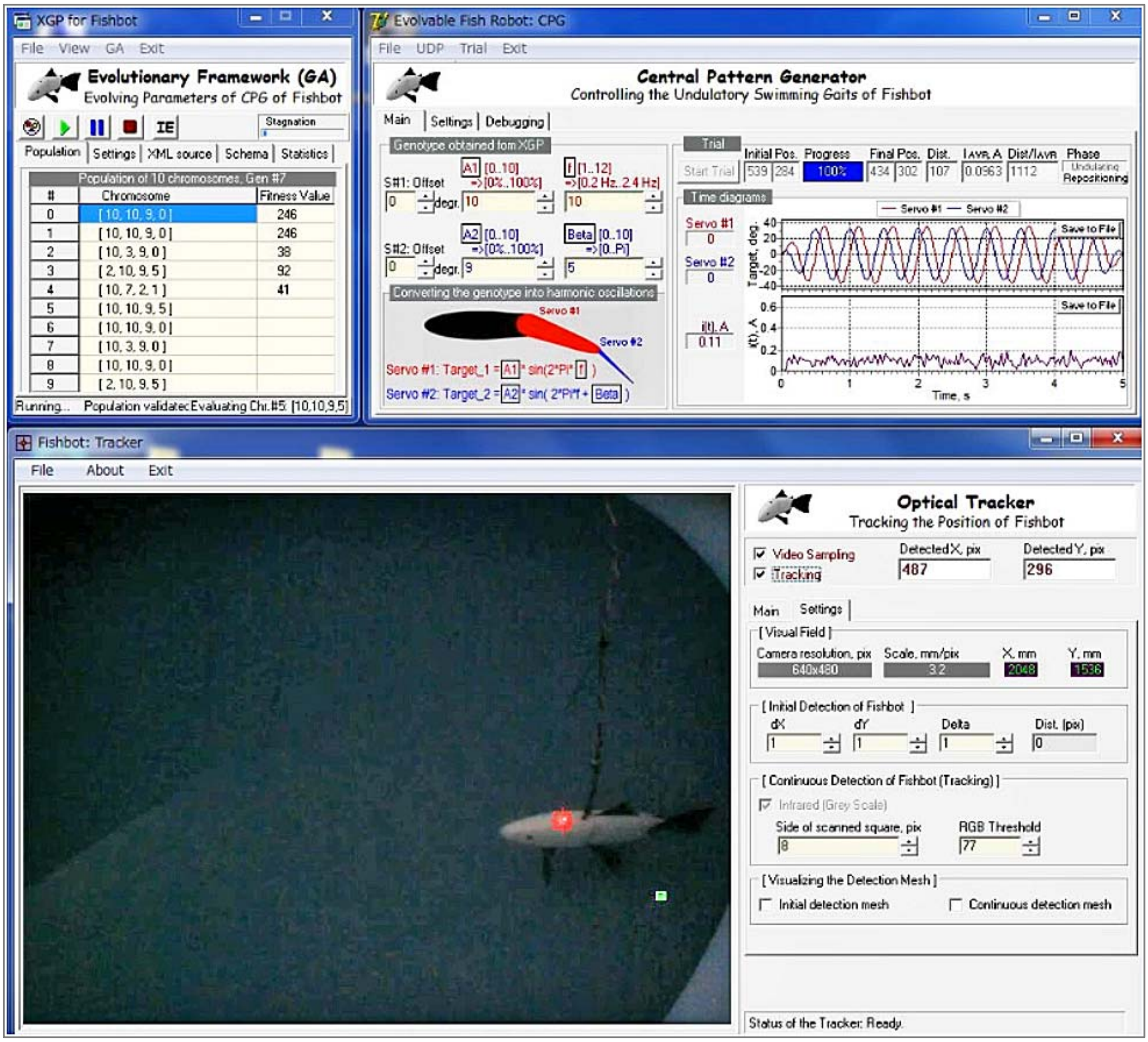

Figure 20. Snapshot of the three subsystems, obtained during the evolution of the vales of main parameters of undulation that result in maximum thrust

6. Experimental Results

In this section we present the experimental results of 10 independent runs of GA, intended to evolve such values of the four main parameters of undulation that result in maximum thrust generated by the swimming fish robot. These results are intended to serve as a proof of the technical soundness of the decisions taken during the design of the fish robot, and the three software subsystems.

The *fitness converge characteristics* of the best-of-generation chromosome of 10 independent runs of GA are shown in Figure 21. As illustrated in the figure, in all runs the best-of-generation chromosomes evolve from the initial values of raw fitness 72~284 to the final values of 271~387.

The values of the 10 best-evolved, unique chromosomes obtained from the 10 independent runs of GA are shown in Table 6. As the data shown in the table suggest, the highest values of raw fitness (e.g., highest thrust, generated by the fish robot) is achieved for relatively narrow range of values of alleles of chromosomes. This, in turn, suggests that it might be just one area in the fitness landscape that exhibits the highest values of the raw fitness. Indeed, the amplitude of the front servo motor A_1 of 9π/40 ~ π/4 radians, i.e., 90%~100% of its maximum value of π/4. The corresponding value A_2 for the rear servo motor is 8π/40~π/4 radians, i.e., 80%~100% of the maximum amplitude of oscillation. There is either no or a little phase shift β (between 0 and π/5 radians) between the oscillations of the two servo motors. The frequency of oscillation f is a range 2.2 Hz ~ 2.4 Hz, i.e., near or equal to the maximum possible value of 2.4 Hz.

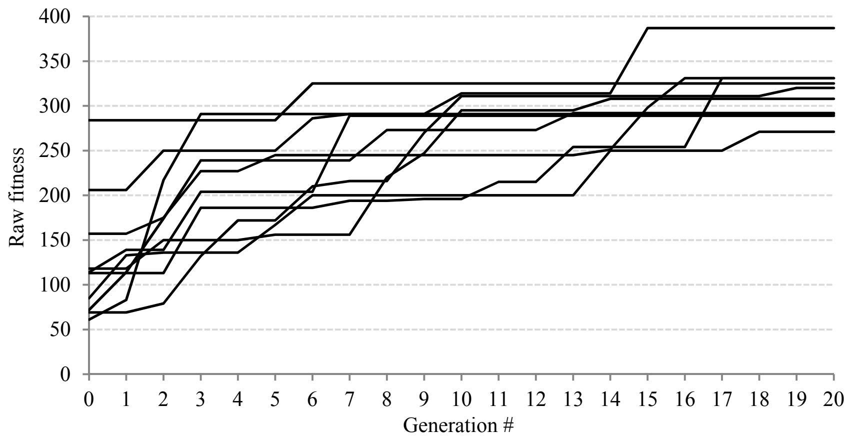

Figure 21. Fitness convergence of the best-of-generation chromosomes of 10 independent runs of GA.

Table 6. Values of 10 best evolved, unique chromosomes

#	A_1		f		A_2		β		Raw Fitness
	Discrete integer value encoded in chromosome	Actual value, radians	Discrete integer value encoded in chromosome	Actual value, Hz	Discrete integer value encoded in chromosome	Actual value, radians	Discrete integer value encoded in chromosome	Actual value, radians	
1	9	9π/40	12	2.4	10	π/4	0	0	387
2	10	π/4	12	2.4	10	π/4	2	π/5	331
3	9	9π/40	12	2.4	10	π/4	2	π/5	326
4	10	π/4	11	2.2	10	π/4	0	0	325
5	10	π/4	12	2.4	8	8π/40	1	π/10	320
6	9	9π/40	11	2.2	10	π/4	0	0	314
7	10	π/4	11	2.2	9	9π/40	0	0	314
8	9	9π/40	12	2.4	8	8π/40	1	π/10	311
9	10	π/4	12	2.4	9	9π/40	1	π/10	308
10	7	7π/40	12	2.4	9	9π/40	0	0	301

7. Conclusion

In this report we presented the main aspects of designing and controlling the fish robot. We also elaborated on the evolution – via genetic algorithms – of the main parameters of undulation that result in highest thrust generated by the undulating robot. In our future work we are planning to extend our experiments with evolution of both (i) fast and (ii) energy efficient swimming gaits. Also, we are contemplating investigating the value of *Strouhal number* of the best evolved gaits, in order to understand whether it could be applied for analytical estimation (rather than evolution) of the optimal values of some of parameters of undulation.

References

[1] E. Angeline, P. J., Kinnear, K.E. Jr., Genetic Programming and Emergent Intelligence, Advances in Genetic Programming, MIT Press, pp. 75–98, 1994

[2] F. Fish, Swimming Strategies for Energy Economy, Fish Locomotion: An Eco-ethological Perspective, 2010

[3] D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning; Addison-Wesley Longman, Boston, MA, USA, 1989

[4] Huizhou AGF-RC Electronic Technology Company, A20CLS Programmable Micro Servo, URL: https://www.agfrc.com/index.php?id=1906

[5] J. Holland, Adaptation in Natural and Artificial Systems; The University of Michigan Press: Ann Arbor, MI, USA, 1975

[6] P. S. Katz, Evolution of central pattern generators and rhythmic behaviours, Philosophical Transactions B, Royal Society Publishing, Vol. 371, Issue 1685, 2016, URL: https://doi.org/10.1098/rstb.2015.0057

[7] J. R. Koza, M. A. Keane, W. Mydlowec, J. Yu, and G. Lanza, Genetic Programming IV: Routine Human-competitive Machine Intelligence, Kluwer Academic Publishers, 2003

[8] A. G. Leal Junior., R. Milanezi de Andrade, and A. B. Filho, Series Elastic Actuator: Design, Analysis and Comparison, Recent Advances in Robotic Systems, Edited by G. Wang, IntechOpen, 2016, URL: https://www.intechopen.com/chapters/51224

[9] K. H. Low, C. Zhou, and Yu Zhong, Gait Planning for Steady Swimming Control of Biomimetic Fish Robots, Advanced Robotics, Vol. 23, No. 7-8, 2009, pp.805-829, Taylor & Francis, URL: https://doi.org/10.1163/156855309X443124

[10] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press: Cambridge, MA, USA, 1998.

[11] S. Nolfi, and D. Floreano, Evolutionary Robotics: The Biology, Intelligence, and Technology of Self Organizing Machines, MIT Press, Cambridge, MA, USA, 2000.

[12] L. Phillips, Turbulence, the Oldest Unsolved Problem in Physics, Ars Technica, 2018, URL: https://arstechnica.com/science/2018/10/turbulence-the-oldest-unsolved-problem-in-physics/

[13] J. Sánchez-Rodríguez, F. Celestini, C. Raufaste, and M. Argentina, Proprioceptive Mechanism for Bioinspired Fish Swimming, Physical Review Letters, American Physical Society, Vol. 126, Issue 23, 2021, URL: https://link.aps.org/doi/10.1103/PhysRevLett.126.234501

[14] I.Tanev, DOM/XML-Based Portable Genetic Representation of Morphology, Behavior and Communication Abilities of Evolvable Agents, *Artificial Life and Robotics*, Vol.8, Number 1, 2004, pp.52-56, Springer-Verlag

[15] I.Tanev, T.Ray and A.Buller, Automated Evolutionary Design, Robustness and Adaptation of Sidewinding Locomotion of Simulated Snake-like Robot, *IEEE Transactions on Robotics*, Vol.21, Number 4, August 2005, pp.632-645

[16] S. Verma, G. Novati, and P. Koumoutsakos,Efficient collective swimming by harnessing vortices through deep reinforcement learning, Proceedings of the National Academy of Sciences of USA, Vol. 115, Issue 23, 2018, pp. 5849-5854, URL: http://www.pnas.org/content/115/23/5849.abstract

[17] T. Vo, H. Kim, and B. Lee, Smooth gait optimization of a fish robot using the genetic-hill climbing algorithm. Robotica, 30(2), 2012, pp. 257-278, doi:10.1017/S0263574711000555

[18] W. Wang, D. Gu and G. Xie, Autonomous Optimization of Swimming Gait in a Fish Robot With Multiple Onboard Sensors, in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 5, pp. 891-903, May 2019, doi: 10.1109/TSMC.2017.2683524

[19] W. Wang, J. Guo, Z. Wang, and G. Xie, Neural Controller for Swimming Modes and Gait Transition on an Ostraciiform Fish Robot, in Proc. IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Wollongong, Australia, July 9-12, 2013, pp. 1564-1569

[20] E. W. Weisstein, Delta Function, MathWorld – a Wolfram Web Resource, URL: https://mathworld.wolfram.com/DeltaFunction.html

[21] C. Zhou, and K. H. Low, On-line Optimization of Biomimetic Undulatory Swimming by an Experiment-based Approach, *Journal of Bionic Engineering,* Vol. 11, Issue 2, Elsevier, 2014, pp. 213-225

[22] J. Zhu, C. White, D. K. Wainwright, V. Di Santo, G. V. Lauder and H. Bart-Smith, Tuna robotics: A high-frequency experimental platform exploring the performance space of swimming fishes, S*cience Robotics,* Vol. 4, Issue 34, 2019, URL: https://robotics.sciencemag.org/content/4/34/eaax4615