



## Original Research Article

### Plain Swimming Algorithm for a Mackerel (*Scomber scombrus*) Robotic Fish

\*Afolayan, M.O. and Iorpenda, M.J.

Department of Mechanical Engineering, Faculty of Engineering, Ahmadu Bello University, Zaria, Kaduna State, Nigeria.

\*tunde\_afolayan@yahoo.com

<http://doi.org/10.5281/zenodo.5805402>

#### ARTICLE INFORMATION

##### Article history:

Received 17 May, 2021

Revised 29 Jun, 2021

Accepted 09 Jul, 2021

Available online 30 Dec, 2021

##### Keywords:

Robotic fish

Teleost sp.

Hyper-redundant

Swimming

Biomimicry

Fish

Locomotion

#### ABSTRACT

*This work presents the performance of the plain swimming motion algorithm used in the design of a robotic fish based on Mackerel (*Scomber scombrus*) which belong to the Teleost species of fishes. The wriggling motion patterns were built into the robot firmware. The robot tail is a six segments plywood panels with vulcanized rubber acting as joints. This tail structure is driven by three remote-control servomotors (Futaba 3003) and that are controlled by a single PIC18F4520 microcontroller. The microcontroller which is the heart of the robot firmware, use the built-in algorithm to determine the sequence of the operation of those three servomotors in a way that they are all simultaneously out of phase at any given instance, thus creating a traveling wave that propels the robot forward. Experimental results based on the algorithm shows that the robot was able to achieve a maximum linear speed of 8.26 cm/s (0.21 body length per second) while the tail was oscillating at a frequency of 1.15 Hz.*

© 2021 RJEES. All rights reserved.

## 1. INTRODUCTION

Fishes are known for their fulgurating acceleration inside water as they can execute rapid, agile maneuvers while trying to escape a predator (Jindong and Huosheng 2007; Marchese *et al.*, 2014; NMRI 2020). They are also quiet and efficient swimmers according to Triantafyllou and Triantafyllou (1995) when compared with manmade underwater devices. These abilities have inspired several researchers to build whole robot structures (mimicking fishes) with novel materials such as shape memory alloy (Suleman and Crawford, 2008; Yan *et al.*, 2012), piezoelectric materials (Heo *et al.*, 2007; Nguyen *et al.*, 2011), vulcanized carbon-filled rubber (Afolayan *et al.*, 2012), ICPF actuators (Guo *et al.*, 1998), etc. Different fish swimming mode and shapes were mimicked as well for various reasons according to the researchers (Yan *et al.*, 2005; Sefati *et al.*, 2012; Yangwei *et al.*, 2015).

Fishes swim using different modes such as straight wiggling motion, sharp turning, steady turning, diving, surfing (going to surface), gliding as in sailfish, lateral undulation, sinus lifting, sidwinding, and climbing, etc. These swimming modes can be classified broadly into two fundamental groups; body and/or caudal fin (BCF) locomotion, and median and/or paired fin (MPF) locomotion (Sfakiotakis *et al.*, 1999). BCF uses body undulation or the caudal fin and allows much thrust and acceleration; however, swimming at low speed and higher propulsive efficiency and precise maneuverability is better off with MPF locomotion (Yangwei *et al.*, 2015). *Teleost* species of fishes like mackerel use BCF locomotion.

Locomotion in fishes is shared by other serpentine creatures such as snakes, worms, and even lizards. In biorobotic systems, the motion generation can be broadly divided into three according to Kevin (1997), namely (1) built-in motion patterns (e.g., use of central pattern generators (CPGs) or motion database), (2) follow the leader approach where other segments follow the first segment or last when reversing and (3) mathematically generated motions (e.g., serpenoid curves (Guo *et al.*, 1998; Yamada and Hirose, 2006), machine learning (Gong *et al.*, 2015).

The use of built-in patterns presents a rather inflexible but less computational demanding design. This method is good for un-tethered robots and where power consumption is a concern. The follow the leader approach requires the leading segment to be controlled while the control instruction is sequentially passed to the subsequent segments. This design is good for motion in an unstructured environment. Mathematically generated motions are power demanding and may not be good for mobile devices except power management of some sort is included in its implementation.

This work is about the plain swimming algorithm developed for a Mackerel based robotic fish. It could be suitable for other robotic fishes imitating *Teleost* species of fishes such as Herring, Pike, Carp, Cod, Salmon, Bonito, Tuna, and Sword Fish. *Teleost* species of fish are the fastest moving creature inside the water and imitating them will be a step further into designing an efficient and fast underwater system.

This work is justified as many other works on robotic fishes (for example Sitorus *et al.*, 2009, Liu and Hu, 2010, Marchese *et al.*, 2014) do not present the nitty-gritty of the inner workings or communication protocols within their robots and how the swimming patterns were implemented beyond a mechanical description of their devices. Some works like that in Liu and Hu (2010) and Suleman and Crawford (2008) were non-specific on the control sequence of their robotic fish tail, though they went beyond others in using flow chart (Liu and Hu, 2010) and succinct diagrams (Suleman and Crawford, 2008) to explain how they implement their control mechanism in their robotic fishes.

## 2. METHODOLOGY

### 2.1. Design Description

A brief description of the robot for which the algorithm was developed for is given in this section. However, a more exhaustive mechanical description is in the previous work by Afolayan *et al.* (2012). Using the 3D CAD model of the robot tail, (Figure 1), “the tail, consisting of 1.5 mm thick vulcanized rubber joint (A) was sandwiched between pairs of rigid support segments (1) to (6) made from 3.2 mm thick seasoned plywood. The support (6) is attached to oval support (B) having six pass-through holes (C) for the cables support. The servomotor (Futaba 3003) (D) is attached to the oval support (B) having pass-through holes for the cables. The nylon cables were connected to the servomotor lever (E). The microcontroller (PIC18F4520) controls the sequence of turning of the servomotors (D). The three servomotors are connected to segments (1), (3), and (5) with nylon cables. It sends the angular displacement information to the servomotors in such a manner that its lever (E) will oscillate at  $\pm$  angle (F). On both sides of each segment (1) to (6) is located quarter pulleys (H) (made from 19 mm thick plywood) over which the 0.5 mm diameter nylon cable (G) passes before hooking to those segments, only one cable is shown for clarity. The other segment simplifies the design as they act to restore the joints to their static states and in getting the desired serpentine shape. This approach simplifies the number of motors required and hence the control scheme. Six segments are controlled by three motors connected to segments 1, 3, and 5.

## 2.2. Steady Swimming

Steady swimming refers to movement of a fish from a point to another point following a straight path as opposed to several other motions such as pointed out in the introductory section.

### 2.2.1. Steady swimming methodology as approximated by the robot mechanism and algorithm

*Teleost* sp. of fishes like Mackerel swim by generating a traveling wave such that the amplitude is least at the beginning (towards the head) and is highest at the fin tip (Figure 1a). Thus, the algorithm is to control the servomotors such that that wave pattern (Figure 1a) is generated continuously. In this work, the six links implementation in the robot (Figure 2a) was approximated by three links configurations as shown in Figure 1b. The rubber joints allow the shape to be closer to the desired pattern required for swimming and also for restoring the links to straight shape. To swim at a steady pace, the algorithm must meet the following requirements:

- Generation of phase difference in the robot tail section such that Figure 2a and Figure 2b are formed.
- It should possess the ability to dynamically adjust the segment displacement (i.e., amplitude) on either side of its lateral axis.
- It must take into consideration the fact that the servomotors moving parts and tail segments have inertia that will not allow them to follow the pulse width modulated (PWM) signal in real-time.

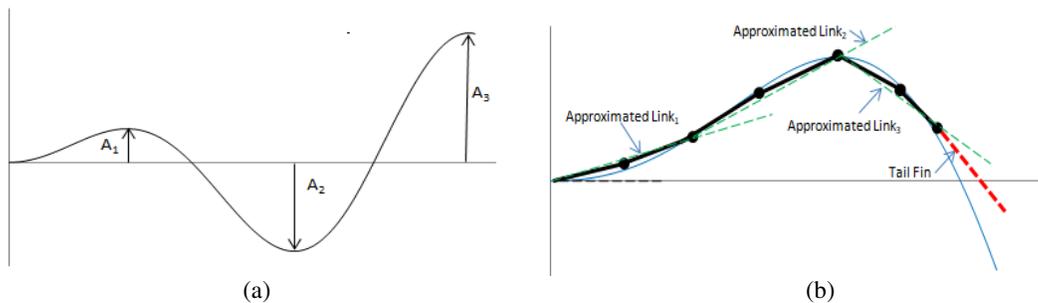


Figure 1: *Teleost* fish swimming pattern – (a) Tail amplitude increases toward the tail fin, such that the amplitude at  $A_1 < A_2 < A_3$  all the time. (b) The six links implementation in the robot is approximated by three rigid links configuration

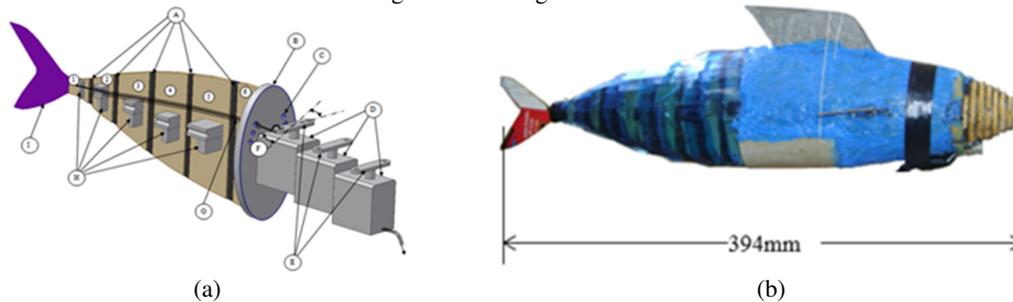


Figure 2: (a) 3D CAD model of the robot (b) the assembled robot

### 2.2.2. The algorithm as implemented in the robot controller

The flow chart for the algorithm is as shown in Figure 3, it is a firmware-based Pulse Width Modulated (PWM) signal generator for 3 or more rigidly coupled/linked PWM for motor control. It is divided into 4 sections; Section A is the time base. It uses Timer0 (INT0) interrupt set to trigger repetitively at 20 ms interval.

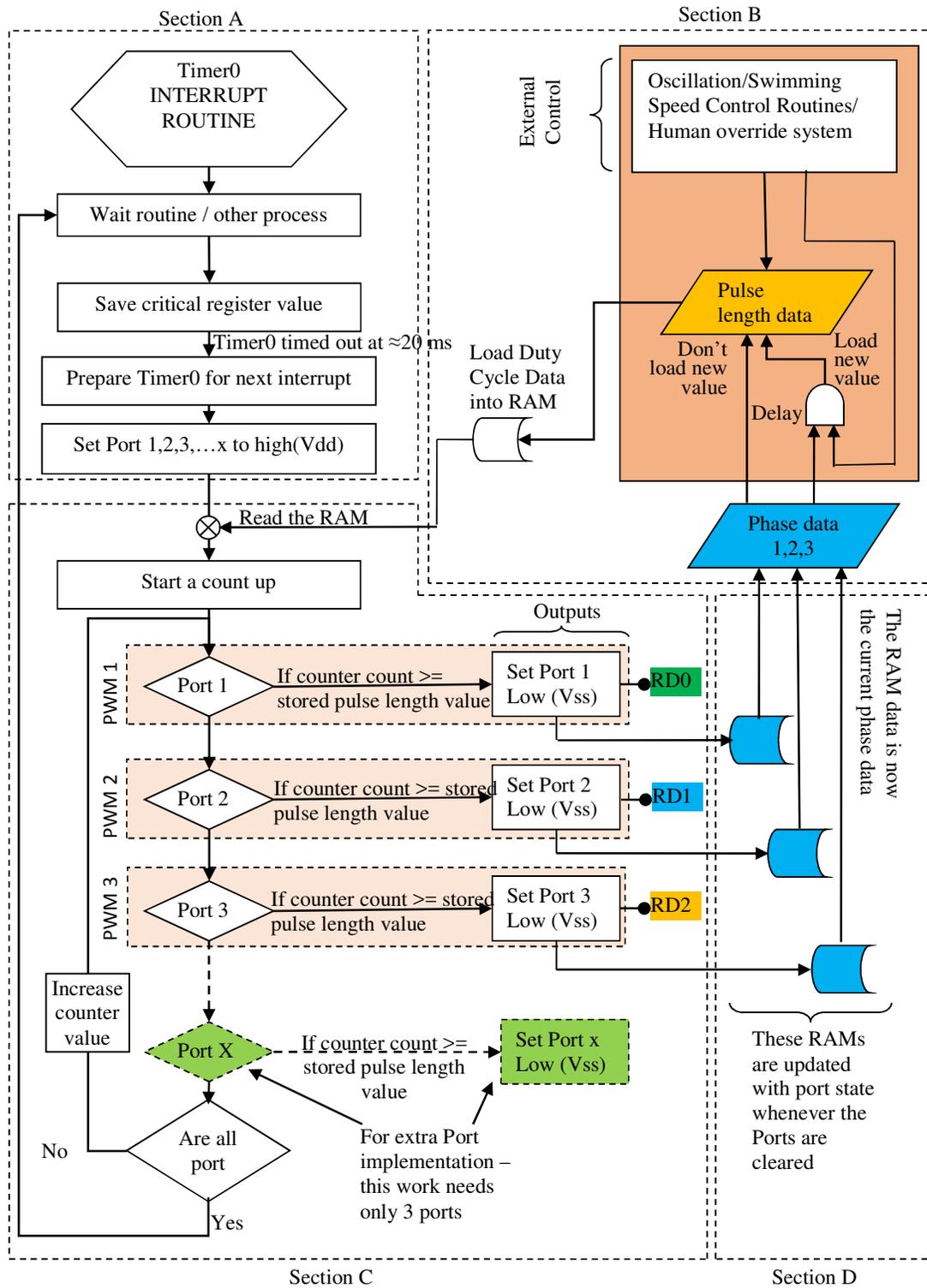


Figure 3: Flow chart of the concurrent PWM generator

The targeted servomotor is Futaba RC servomotor and it requires a 20 ms data refresh rate. Section B is where the pulse length data and phase data exist, and what value to load into the duty cycle random access memory (RAM) is determined and manipulated by internal and external inputs. The data are in pairs, one for the right side and the other for the left side of the robot. Section C generates the Pulse Width Modulated (PWM) whose duty cycles are based on the inputs from section B. Section D keeps a record of the current state so that it can be regenerated again or used as a reference for the next data to be loaded in section B.

For the phase difference generation, the pulse lengths are retrieved from a table of pre-calculated values hardcoded into the firmware, and each corresponds to about 1 ms to 2 ms long ( $0^\circ$  to  $180^\circ$ ) when loaded into the PWM generator of Figure 3. Approximately, a 1.5 ms long pulse translates to  $90^\circ$  turn of the servomotor lever. The pulse lengths are retrieved during the idle time. The values are selected such that the motor levers are turned to different angles that are out of phase at any point in time as shown in Figure 4.

The length of each pulse determines the angular displacement of each servomotor lever. To generate a different duty cycle for each servomotor, the three ports are first set high by the Timer0. A counting up routine (Section C of Figure 3) is then initiated by looping (sequentially for each port data) until the value is equal to or greater than the retrieved value. This leads to the port (that met the criteria) being set low (to vss). Other ports remain high until set low, thus generating the specified pulse lengths which translate to the duty cycle as desired while the phase remains different. The total lag between the first and the last signal is calculated as follows:

If Port 0 start time is represented as  $T_{port0}$ , then:

Then Port X start time is given as:

$$T_{portx} = T_{port0} + T_{port1} + T_{port2} + \dots + T_{portx-1} \quad (1)$$

$$T_{portx} = T_{port0} + (x - 1) \times T_{CY} \quad (2)$$

Where  $T_{CY}$  is the length of an instruction cycle and Port X refers to other Ports 0, 1, 2

With the clock set to run at 32 MHz, and dividing that by 4 (Microchip microcontroller uses 4 clock pulses per instruction execution generally).

$$T_{CY} = (1/32\text{Mhz}) \times 4 = 0.000000125 \text{ s or } 125 \text{ ns}$$

Hence the lag for  $Port_x = (x - 1) \times T_{CY} = (x - 1) * 125 \text{ ns}$  (from Equation 2)

The lag between the first  $Port_1$  and last  $Port_3$  will be:

$$= (3 - 1) * 125 \text{ ns} = 250 \text{ ns} \quad (x=3 \text{ for our own case, three PWMs, are needed})$$

This lag is fixed since the PWM signals have the same reference, i.e., they have the same period and it is started by a single timer (Timer0).

Dynamic adjustment of the displacement of the segments from the midline (Figure 5) was performed by uniformly adding or removing a fixed value to the pulse length data of each motor. This causes the servomotor lever excursion to either left or right to be adjusted uniformly also. The flowchart is shown in Figure 6.

Also, the tail swing frequency is inversely proportional to the delay variable value. This routine (which is a subset of the flow chart of Figure 3) is expanded in Figure 7. The command to increase or reduce is from a human interfacing device.

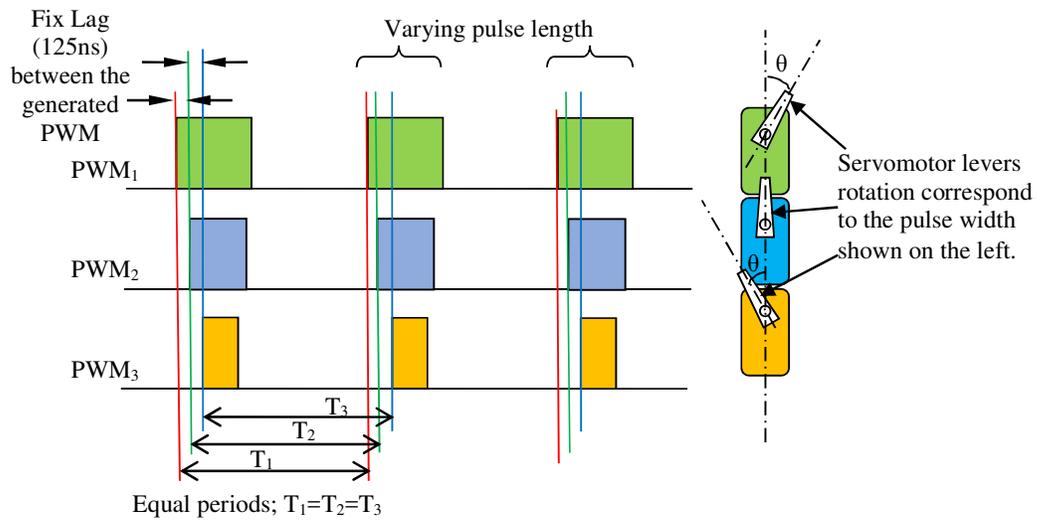


Figure 4: Phase generation with three servomotors

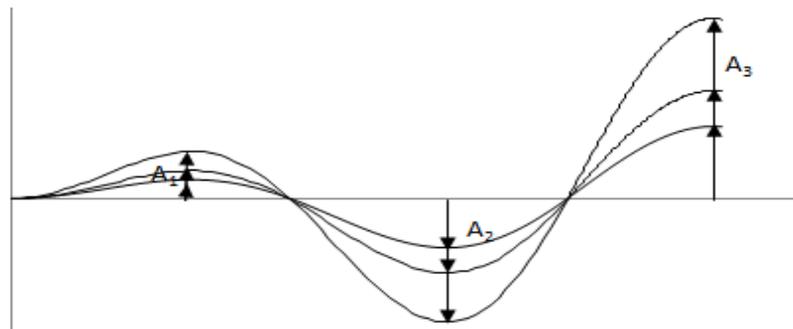


Figure 5: Amplitude control by incrementing the swing of the servomotors horns

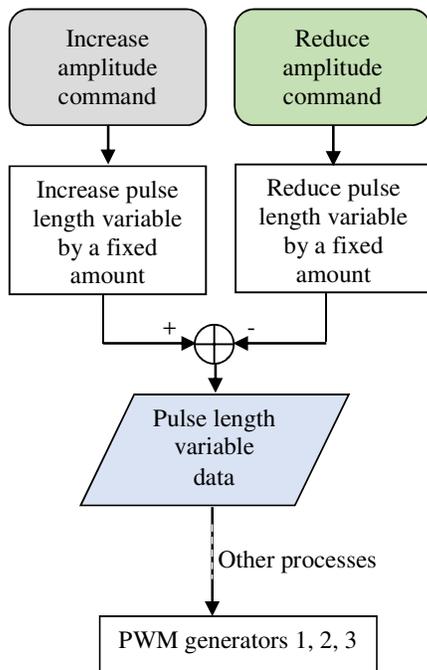


Figure 6: Amplitude control routine

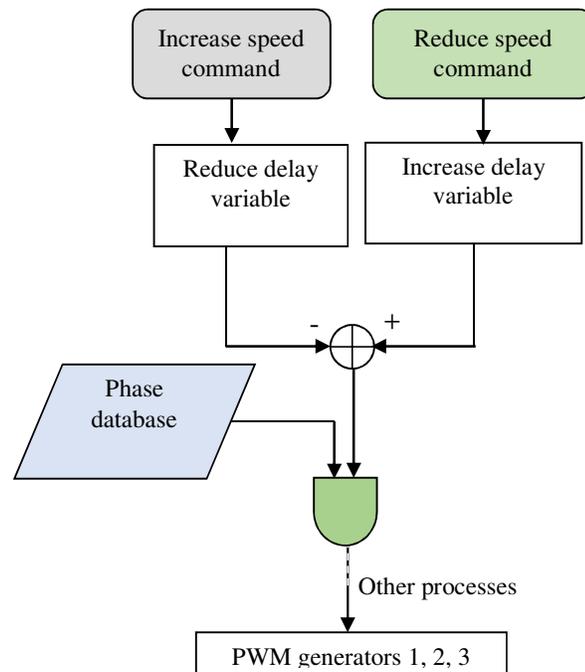


Figure 7: Tail oscillation speed control routine

The motor inertia is handled by introducing a delay routing (Figure 3 Section B and Figure 8). The delay routine prevents loading new/updated pulse length data until a predetermined time. The algorithms 20 ms pulse rate and PWM output are unaffected; the 20 ms pulse rate is based on Timer0 which is external to section B where the delay routine is located. Also, the PWM generator (Section C of Figure 3) will keep using the old data to work. Thus, the duty cycle will be unchanged and the motor will have time to catch up. Furthermore, the delay can be externally adjusted for other unforeseen purposes.

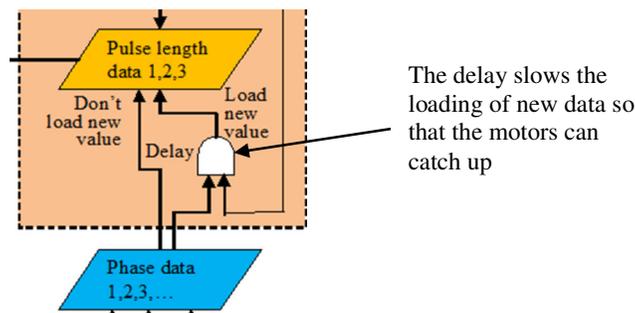


Figure 8: The delay element for the motor inertial

### 2.3. Experimental Verification

Experiments conducted are of two folds: the first set is on algorithm verification and the second is the field tests. The algorithm correctness at generating 3 concurrent PWM (Figure 4) was verified by using a virtual Logic analyzer (Microchip® MPLab SIM simulator) and an oscilloscope on the 3 output pins of the virtual model and physical PIC18F4520 microcontroller. Furthermore, the three built-in motions were verified out of water. The second experiment is the field testing of each implemented algorithms. This involves placing

the robot inside a pool of water and executing the 3 built-in algorithms. For all the experiments, a Sony Cyber-shot digital camera (model DSC-S730) was used for recording the videos of the experiments at a resolution of 3 megapixels, F-stop of  $f/2.8$ , exposure time of  $1/40$  s, ISO speed of 100. Still images and time stamp were extracted from the recordings using Microsoft Windows Live™ Movie Maker Version 2011 with up to 10 ms resolution in the interval between the still images.

### 3. RESULTS AND DISCUSSION

#### 3.1. Constant Phase Difference of the three Concurrent PWM Eneerator

The Logic Analyzer output (Figure 9a) shows the output of the programmed ports (RD0, RD1, and RD2). The pulse width for each port was not the same, as desired. The implication is that the servomotors' lever would all turn to different angles at that instance from the zero reference position ( $0^\circ$ ) as indicated in Figure 9b. Servomotor lever rotation direction depends on the pulse width. From Figure 9b, the RD1 is assumed to set the middle servomotor to  $0^\circ$ , while RD0 will set the first servomotor to  $(0+\theta)^\circ$  and RD2 will set the third servomotor to  $(0-\theta)^\circ$ . When RD1 rotates by  $\sigma^\circ$ , the other servomotor would have to adjust either by subtraction ( $-\sigma^\circ$  for RD0) or by addition ( $+\sigma^\circ$  for RD2) simultaneously, this will ensure constant phase difference. Summarily, the RD1 is the leader while all the other motors keep a constant phase difference from it and between themselves. Furthermore, the oscilloscope output (Figure 10) of pin RD0 shows a 20 ms repetition rate. A deviation from the 20 ms rate will cause jittering of the assembly; however, this was not observed in the final model.

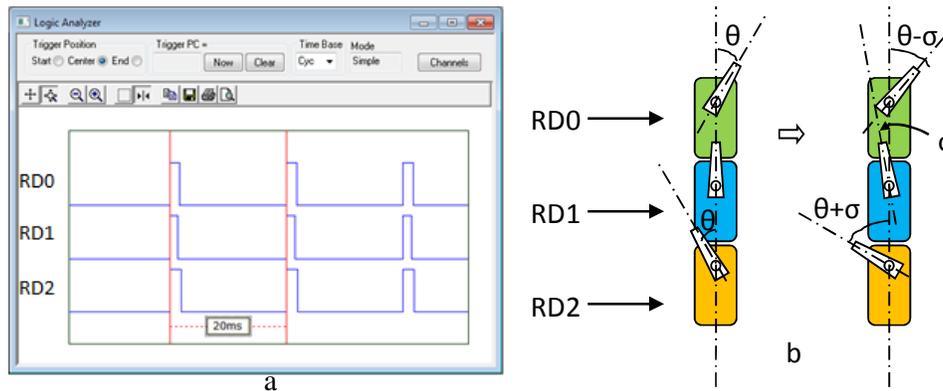


Figure 9: The output shows varying pulse length, all occurring simultaneously at 20 ms interval. The RD0, RD1, and RD2 are the microcontroller ports physical addresses.

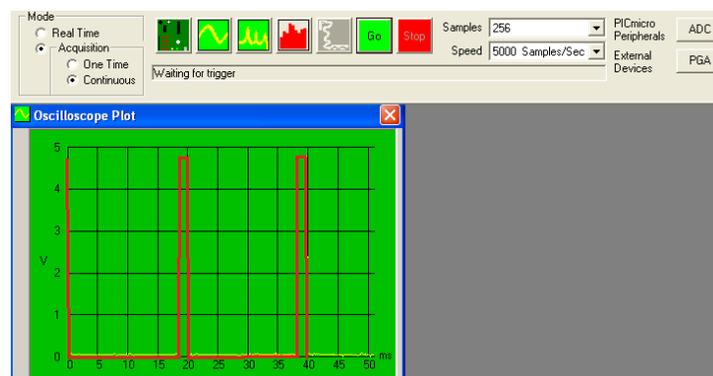


Figure 10: Oscilloscope plot of one channel showing repeated pulses at 20 ms

### 3.2. Traveling Wave Pattern Generation for Steady Swimming

In Figure 11, an amplitude variation in the tail segments towards the peduncle is observable. With reference to the middle line, the amplitude of swing increases from C to A as desired (Figure 2a). In Figure 11, A is  $0.09L$ , B is  $0.05L$  and C is  $0.02L$ , where  $L$  is the robot length ( $L$  is 394 mm).

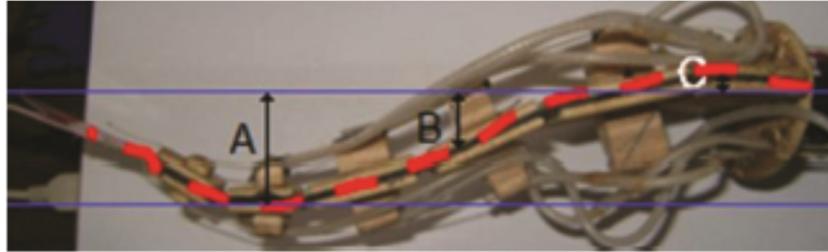


Figure 11: The tail mechanism without its covering shows an increasing amplitude of the segments towards the peduncle

### 3.3. Change in the Amplitude of Oscillation While Swimming

The robot peduncle amplitude changes are shown in Figure 12 and Figure 13a (the peduncle maximum swing). For each set in Figure 12,  $A_p$  is 130 mm, 93 mm, and 72 mm respectively. In between these, the robot could generate other amplitudes by changing the pulse length of each PWM signal (Figure 6). The peduncle displacement with time for the still pictures of Figure 12 is shown in Figure 13b. As the amplitudes change, the pattern remains fairly the same and approaches a bell shape. There is an increase in acceleration towards the midline as  $A_p$  increases. Furthermore, the patterns indicate that the algorithm instruction to change the pulse length (which determines the amount of the motor angular displacement) works consistently; the flaw in the preselected value is still carried over with the changes in the pulse length.

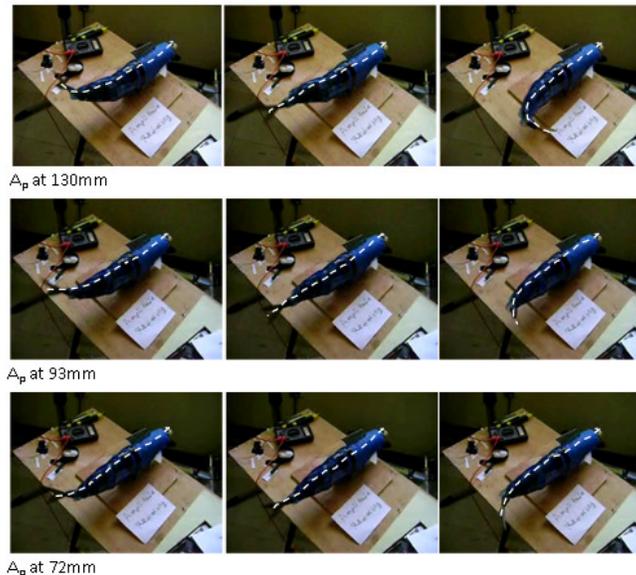


Figure 12: Change in peduncle amplitude from the top set to the bottom set, each set is a complete cycle (left to right motion) at 0.1s interval.  $A_p$  is the amplitude from the tip of the peduncle to the midline.

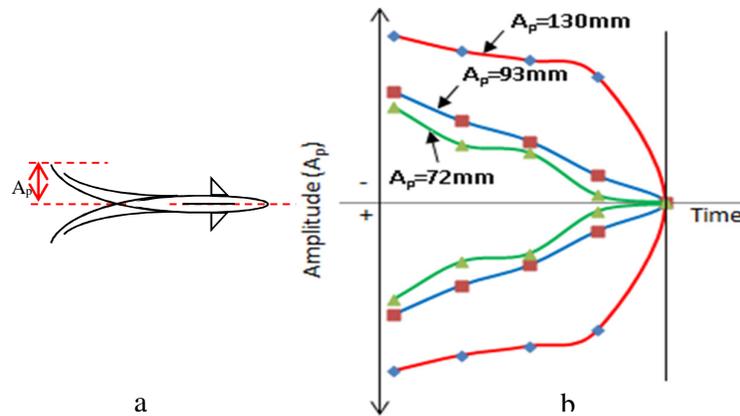


Figure 13: (a)  $A_p$  is the amplitude from the tip of the peduncle to the midline (b) Shows the relative displacement from the midline for the 3 sets of still pictures of Figure 12

### 3.4. Steady Swimming

The robot is shown in Figure 14 performing steady swimming. The average speed was measured as 8.26 cm/s (0.21 body length per second) while the tail oscillation frequency was at 1.15 Hz. The work by Marchese *et al.* (2014) shows a result of 0.6 to 1.02 body length at tail flapping frequency of 1.3 Hz, although for a different fish model (a Carp). The work of Heo *et al.* (2007) shows that a swimming speed of 2.519 cm/s at a tail beat frequency of 0.9 Hz is achievable. However, since the other researchers and our own work involve using different fish models and different mechanical implementations (or design), the outcomes could be affected by these factors.



Figure 14: Steady swimming inside a pool of water

## 4. CONCLUSION

The presented algorithm swim parameters like the PWM timing are hardcoded and were able to make the robot move as intended. The three servomotors combination, each driven by one of the three constantly out of phase PWM signals and the tail inter-segment rubber joints were capable of generating the traveling wave necessary for the straight swimming mode. However, every error in the parameter selection is scaled up or down as evidence in the amplitude adjustment. This is not a desirable behavior. The average speed of 8.26 cm/s (0.21 body length per second) needs improvement when compared to live Mackerel (306 cm/s).

## 5. CONFLICT OF INTEREST

There is no conflict of interest associated with this work.

## REFERENCES

- Afolayan, M. O., Yawas, D. S., Fodayan, C. O. and Aku, S. Y. (2012). Mechanical Description of a Hyper-Redundant Robot Joint Mechanism Used for a Design of a Biomimetic Robotic Fish. *Journal of Robotics*, 2012, pp. 1-16.
- Breder C M (1926) The locomotion of fishes. *Zoologica*, 4, 159–297.
- Gong, C., Travers, M. J., Astley, H. C., Li, L., Mendelson, J. R., Goldman, D. I. and Choset, H. (2015). Kinematic Gait Synthesis For Snake Robots. *The International Journal of Robotics Research*, pp. 100–113.
- Guo, S., Fukuda, T., Kato, N. and Oguro, K. (1998). Development of underwater microrobot using ICPF actuator. In: Proceedings of the 1998 IEEE International Conference on Robotics and Automation, Leuven, Belgium, pp. 1829–1834.
- Heo, S., Wiguna, T., Park, A. C. and Goo, N. S. (2007). Effect of an artificial caudal fin on the performance of a biomimetic fish robot propelled by piezoelectric actuators. *Journal of Bionic Engineering*, 4, pp. 151–158.
- Jindong, L. and Huosheng, H. (2007). A methodology of modelling fish-like swim patterns for robotic fish. In: Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation, Harbin, China, pp. 1316–1321
- Kevin, J. D. (1997). Limbless locomotion: learning to crawl with a snake robot. Ph.D. thesis, Robotics Institute Carnegie Mellon University, Pittsburg, USA.
- Liu, J. and Hu, H. (2010). Biological inspiration: from carangiform fish to multi-joint robotic fish. *Journal of Bionic Engineering*, 7, pp. 35–48.
- Marchese, A. D., Onal, C. D. and Rus, D. (2014). Autonomous Soft Robotic Fish Capable of Escape Maneuvers Using Fluidic Elastomer Actuators. *Soft Robotics*, 1(1), pp. 75-87.
- Nguyen, Q. S., Park, H. C. and Byun, D. (2011). Thrust analysis of a fish robot actuated by piezoceramic composite actuators. *Journal of Bionic Engineering*, 8, pp. 158–164.
- NMRI (2020) National Maritime Research Institute. Available online at <http://www.nmri.go.jp/oldpages/eng/khirata/fish/>. Accessed 04 July, 2020.
- Sefati, S., Neveln, I., Maclver, M. A., Fortune, E. S. and Cowan, N. J. (2012). Counter-propagating waves enhance maneuverability and stability: A bio-inspired strategy for robotic ribbon-fin propulsion. In: The 4th IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics, Roma, Italy, 1620–1625.
- Sfakiotakis, M., Lane, D. M. and Davies, J. B. C. (1999). Review of fish swimming modes for aquatic locomotion. *IEEE Journal of Oceanic Engineering*, 24, pp. 237–252.
- Sitorus, P. E., Nazaruddin, Y. Y., Leksono, E. and Budiyo, A. (2009). Design and implementation of paired pectoral fins locomotion of labriform fish applied to a fish robot. *Journal of Bionic Engineering*, 6, pp. 37–45
- Suleman, A. and Crawford, C. (2008) Design and testing of a biomimetic tuna using shape memory alloy induced propulsion. *Computers and Structures*, 86, pp. 491–499.
- Triantafyllou, M. S. and Triantafyllou, G. S. (1995). An efficient swimming machine. *Scientific American*, 272, pp. 64–70.
- Yamada, H. and Hirose, S. (2006). Development of practical 3-dimensional active cord mechanism ACM-R4. *Journal of Robotics and Mechatronics*, 18(3), pp. 305–311.
- Yan, Q., Han, Z., Zhang, S. and Yang, J. (2005). Parametric research of experiments on a carangiform robotic fish. *Journal of Bionic Engineering*, 5, pp. 95–101.
- Yan, Q., Wang, L., Liu, B., Yang, J. and Zhang, S. (2012). A novel implementation of a flexible robotic fin-actuated by shape memory alloy. *Journal of Bionic Engineering*, 9, pp. 156–165.
- Yangwei, W., Jinbo, T. and Dongbiao, Z. (2015). Design and Experiment on a Biomimetic Robotic Fish Inspired by Freshwater Stingray. *Journal of Bionic Engineering*, 12, pp. 204–216.