Proceedings of the 33rd Chinese Control Conference
July 28-30, 2014, Nanjing, China

Digital Implementation of CPG controller in AVR system

Liang Li[1], Chen Wang[1,2], Guangming Xie[1], Hong Shi[3]

1. Intelligent Control Laboratory, College of Engineering, Peking University, Beijing 100871, P. R. China
E-mail: liatli,wangchen,xiegming@pku.edu.cn

2. Faculty of Mathematics and Natural Sciences, University of Groningen, 9747 AG Groningen, The Netherlands

3. Department of Mathematics and Physics, Beijing Institute of Petrochemical Technology, Beijing, 102617, P. R. China
E-mail: shihong@bipt.edu.cn

Abstract: This paper focuses on the implementation of AVR based Central Pattern Generator(CPG) controller. CPGs are biological networks that generate rhythmic outputs for the locomotion control of animals. Since previous CPG controllers are too complex to be suitable for microcontrollers, we propose a simple yet powerful linear CPG controller. To implement this controller in AVR system, several implementation techniques are presented which includes discretization, cosine curve approximation and variable overflow handling. Both simulations and experiments are carried out in an AVR-based robotic fish, which validate the effectiveness of our controller and implementation architecture.

Key Words: Robotic fish, Central Pattern Generator(CPG), Digital Implementation

1 Introduction

It is proven that animals' movements, such as walking, running, swimming and flying, are generated by the Central Pattern Generators (CPGs) [1] composed of neural networks in the spinal cord or ganglion. It can produce coordinated patterns of high-dimensional rhythmic outputs without any rhythmic inputs from sensory feedback or from higher level controller. Simple stimulating signals, such as a chemical signal or an electrical signal, can drive the networks into corresponding stable oscillating outputs. Another important property is smooth transitions between locomotion patterns [2, 3]. This helps animals adapt to complex surroundings such as looking for food, avoiding predators and migration. Compared to the traditional trajectory approximation method [4, 5], CPG controllers have the following advantages:

1) Complex calculations for accurate points on the trajectory of the robot are avoided, and numerous derivations of equations are eliminated.

2) Robots embedded with CPG controller can adapt to dynamic environment because their controlling parameters can be modified online.

3) CPG controller provides a smooth transition between two locomotion patterns, while trajectory approximation method needs a special consideration.

Recently, more and more researchers have applied CPGs in the field of robotic locomotion controls [6–12]. For example, a humanoid robot capable of stepping and walking with coupled oscillators has been developed by Gordon Cheng group [7]. Inagaki etc. and Yoshihito Amemiya group have applied different CPG oscillator patterns to a hexapod robot and a quadruped robot for different locomotion patterns [8, 9]. Scientists have designed special CPG controllers to mimic salamander and dolphin both in shaping and locomotion [10, 11]. A novel turtle-like robot is also controlled by a coupled CPG controller proposed by Wei Zhao etc.[12].

As most robots are equipped with microcontrollers, one of the most important step is how to embed CPG models in these controllers with low computing. Some hardware implementations of CPG-based locomotion controllers are described in [9, 13]. For example, in [9] an analog CMOS CPG controller is proposed and tested on a quadruped robot . Another hardware implemention of a CPG controller using Field Programmable Gate Array (FPGA) is presented in [13]. Although hardware implementations can improve calculation speed, they are expensive, hard to implement and no need for general robots. In the present paper, we propose an implementation method of a simple yet powerful CPG controller suitable for AVR controllers. AVR controller is one of the most widely used microcontrollers with 8-bit RISC architectures for its easy usage, low power consumption and comparatively low price.

During the application of CPG controller in AVR system, We focus our study on the engineering problems of how to determine discrete system, how to deal with phase overflow and how to handle communications between between CPG controller and upper controller. Simulations between continuous system and discrete system are used to test the validity of our implementation architecture. Experiments of free swimming and locomotion transitions are applied on a robotic fish with whole CPG system, which verifies the effectiveness of our proposed CPG networks including implementation process.

The rest of the paper is organized as following: in section 2, description and stability analysis of the linear CPG model are given. Section 3 presents the implemention architecture of our CPG model in AVR system. In section 4, a robotic fish prototype is introduced, and then simulations and experiments are presented. we conclude this paper in section 5.

2 CPG controller

There are kinds of models to imitate the basic CPG network of organism [14–17]. However, most of them are complicated and nonlinear models ,which hardly can be applied to general microcontrollers. Based on our previous work in [18], we propose a CPG network composed of three linear oscillators.

The architecture of CPG network composed of three oscillators is shown in Fig. 1. And the mathematical model

This work was supported by Beijing Municipal Commission of Education Science and Technology Program (KM201310017006).

$$\ddot{r}_i(t) = \alpha_i\left[\alpha_i\left(R_i - r_i(t)\right) - 2\dot{r}_i(t)\right] \tag{1}$$

$$\ddot{x}_i(t) = \beta_i\left[\beta_i\left(X_i - x_i(t)\right) - 2\dot{x}_i(t)\right] \tag{2}$$

$$\ddot{\phi}_i(t) = \sum_{j=1,j\neq i}^{N} \mu_{ij}\left[\mu_{ij}\left(\phi_j(t) - \phi_i(t) - \varphi_{ij}\right) - 2\left(\dot{\phi}_i(t) - 2\pi\omega\right)\right] \tag{3}$$

$$\theta_i(t) = x_i(t) + r_i(t)\cos\left(\phi_i(t)\right), \tag{4}$$

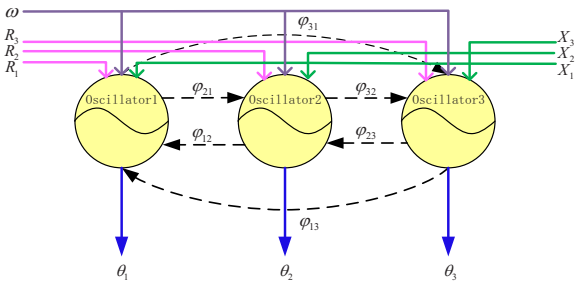


Fig. 1: A simple description of the CPG network. The black dashed line denotes a phase bias φ_{ij} between the two connected oscillators. The ith oscillator has an amplitude input, R_i, and an offset input, X_i, described in pink and green lines, respectively. All the frequency inputs are the same, represented in gray line. The blue solid line aside with θ_i describes the output of the ith oscillator.

of oscillator is expressed as equations (1)-(4), where the state variables $r_i(t)$, $x_i(t)$, and $\phi_i(t)$ respectively represent, the amplitude, the offset and the phase of the ith oscillator at time t. $\theta_i(t)$ represents the output of oscillator i, that is the desired deflection angle of the corresponding joint i at time t. The controlling parameters are R_i, X_i and ω denoting amplitude set, offset set and common frequency set, respectively. The coupling property is defined by phase biases φ_{ij}, representing how oscillator j influences oscillator i. Structural parameters α_i, β_i and μ_{ij} determine the transient dynamics which means how quickly r_i, x_i and $\phi_j - \phi_i$ converge to the desired ones.

Overall, The presented linear CPG model is composed of four parts, amplitude regulation in equation (1), offset regulation in equation (2), phase and frequency regulation in equation (3) and output magnitudes computation in (4). Of all the variables in this mode, there is one coupled variable inhibiting neighbour's oscillating, which causes the consensus of these oscillators. And finally, formula (4) integrates all the four parameters to generate a sine-wave.

3 Implementation of CPG based on Microcontroller

There are some challenges to implement CPG controller in AVR system, such as how to improve the computational accuracy with an 8-bit MCU, how to find a way to minimise calculation time and how to make the system respond to interruptions quickly. In this section we first give a simple introduction of AVR. And then, discrete CPG model is proposed inspired by Euler difference. An algorithm of cosine curve approximation with high accuracy is shown, and solution to the problem of phase overflow is also discussed. At last, implementation of communication between upper computer and AVR is presented.

3.1 Introduction of AVR

AVRs are a family of 8-bit RISC microcontrollers which are produced by Atmel [19]. Their memory is composed of a Harvard architecture with a 16-bit word program memory and an 8-bit word data memory. Because of its comparatively low price, high-performance and low power consumption, the AVR microcontrollers are very popular in embedded system.

In this paper, we take ATmega 128L-8MUR as our controller on robots. The ATmega 128L-8MUR is equipped with 128K Bytes of Flash memory, 4K Bytes of SRAM, a 4K Bytes of EEPROM, an 8-channel 10-bit A/D converter, and a JTAG interface for on-chip debugging. The device supports throughput of 8MIPS at its Maximum Operating Frequency 8MHz and operates between 2.7-5.5v Direct Current. The CPU core of ATmega 128L-8MUR is able to access memories, perform calculations, control peripherals and handle interrupts.

3.2 Discrete CPG Model

Equation (4) expresses the cosine outputs of the oscillator model by inputting phase values, amplitude values and offset values. However, all the direct solutions of phase, amplitude and offset are exponential functions which are difficult to deal for AVR controller. To solve differential equations with small calculation, Euler difference method is applied for our AVR system.

Supposing $dr_i = \dot{r}$, $dx_i = \dot{x}$, $d\phi_i = \dot{\phi}_i$, using Euler difference method, the equations (1)-(4) can be rewritten as equations (5)-(11), where h is set to be $0.02s$, which means the calculation period for the system is $20ms$. n denotes the current step of calculation. Through this way, values of θ can be obtained from AVR on time, and calculations of exponential functions are avoided. Fig. 2 shows a simulation result by this method. The system starts at time $t = 0s$ with $R = \begin{bmatrix}0.05\pi & 0.15\pi & 0.25\pi\end{bmatrix}^T$, $X = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$, $\omega = 0.5Hz$, $\varphi_{12} = 0.5\pi$ and $\varphi_{13} = 0.3\pi$, and the system changes at time $t = 4s$ with $R = \begin{bmatrix}0.08\pi & 0.18\pi & 0.28\pi\end{bmatrix}^T$, $X = \begin{bmatrix}0.05\pi & 0.05\pi & 0.05\pi\end{bmatrix}^T$, $\omega = 1Hz$, $\varphi_{12} = 0.1\pi$ and $\varphi_{13} = 0.3\pi$. Within two seconds after control variables changed, the system reaches the steady state which corresponds to the CPG parameters (see Fig. 2).

Fig. 3 gives a detailed comparison between the analytic

$$dr_i(n+1) = dr_i(n) + h \cdot \left\{-\alpha^2 \left[r_i(n) - R_i\right] - 2\alpha \cdot dr_i(n)\right\} \tag{5}$$

$$r_i(n+1) = r_i(n) + h \cdot dr_i(n) \tag{6}$$

$$dx_i(n+1) = dx_i(n) + h \cdot \left\{-\beta^2 \left[x_i(n) - X_i\right] - 2\beta \cdot dx_i(n)\right\} \tag{7}$$

$$x_i(n+1) = x_i(n) + h \cdot dx_i(n) \tag{8}$$

$$d\phi_i(n+1) = d\phi_i(n) - h \cdot \left\{\mu^2 \sum_{j=1, j\neq i}^{N} \left[\phi_i(n) - \phi_j(n) - \varphi_{ji}\right] + 2(N-1) \cdot \mu \left[d\phi_i(n) - 2\pi\omega\right]\right\} \tag{9}$$

$$\phi_i(n+1) = \phi_i(n) + h \cdot d\phi_i(n) \tag{10}$$

$$\theta_i(n+1) = x_i(n+1) + r_i(n+1)\cos\left(\phi_i(n+1)\right), \tag{11}$$

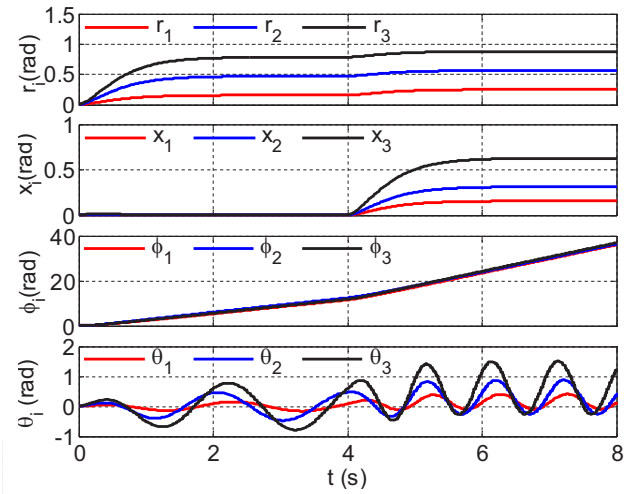


Fig. 2: The simulation of CPG by Euler difference method, r_i, x_i, ϕ_i and θ_i respectively represent the amplitude, the offset of the phase, the phase and the output angle. The system starts at 0 s and changes its parameters at 4 s.

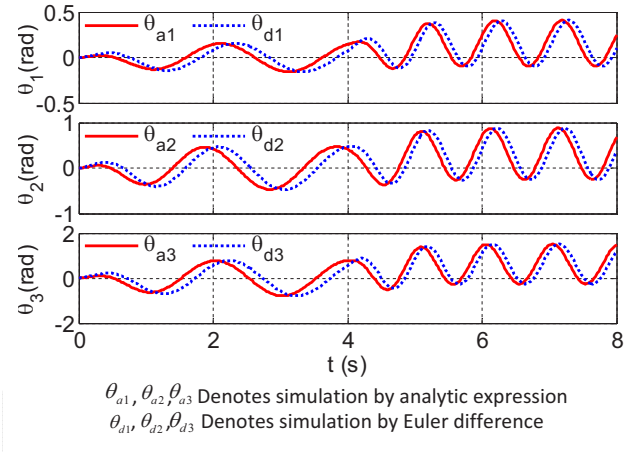


Fig. 3: The figures represent the comparison of θ_i obtained by analytical formula (the red ones) and Euler difference (the blue ones) respectively.

method and Euler difference method with the same setting parameters. The red solid lines depict the variation of θ_i obtained by the analytic method, while the blue dash ones denote the variation of θ_i obtained by the Euler difference method. Apparently, some phase differences exist between the red and blue lines. Fig. 4 illustrates the aforementioned phase difference and the error of ϕ_i . The red solid curve shown in Fig. 4 reflects that the two curves in Fig. 3 are almost the same after the system is stable. after a modification of the phase the outputs are almost the same which means there is only a fixed phase difference between Euler difference and analytical method. With further simulation by MATLAB, we find that the difference is mainly caused by the configurable coefficients, such as μ and ω, and the relationship between μ and ω influences the difference remarkably. By selecting a simple linear function $\mu = 3.5\omega$, we can reduce the phase difference to 0.042 which can be neglected.

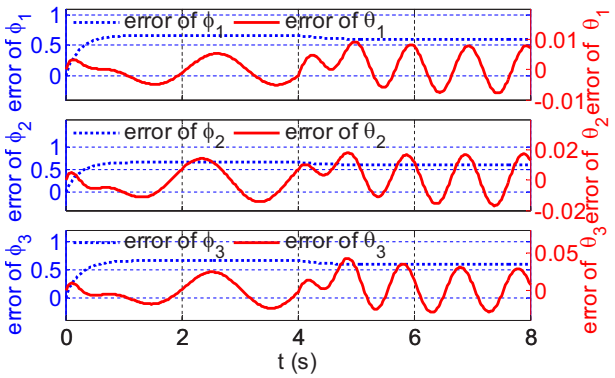


Fig. 4: The figures represent the error of ϕ_i (dotted blue lines) calculated by analytical formula and Euler difference, and red lines represent errors of θ_i.

3.3 The Cosine Curve Approximation

θ_i in equation (4) is a nonlinear periodic cosine function. As ATmega 128L-8MUR only supports basic arithmetical operations, such as add, subtract, multiply and divide, algorithm solving cosine function should be developed before implementing CPG controller in AVR system. Cosine function can be obtained by the formula $cos(t) = sin(t + \pi/2)$ if we can get a good approximation of the sine function. We only need to consider one period from $-\pi$ to π because sine is a period function of 2π,. Here we choose Michael's ap-

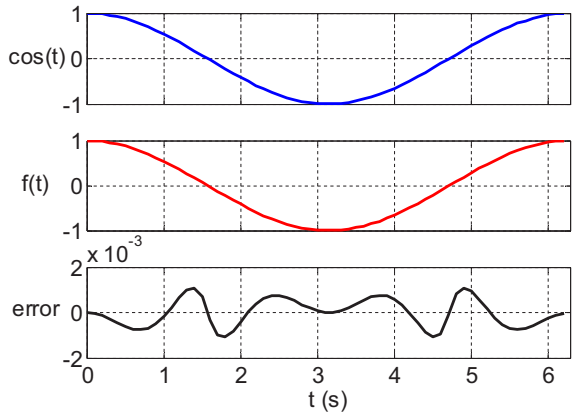


Fig. 5: The comparision between $\cos(t)$and $f(t)$ in a period. The first one describes the cos function and the second one represents the fitting function $f(t)$, The last one shows the error between these two functions in a period.

proximation method to implement sine function, owing to its high speed and accuracy [20]. An intermediate function $g(t)$ is first calculated, then the desired sine value $f(t)$ of time t is got by this intermediate value $g(t)$. A detailed description of this algorithm is shown as below:

$$g(t) = \begin{cases} 1.27t + 0.405t^2 & -\pi < t < 0 \\ 1.27t - 0.405t^2 & -\pi < t < 0 \end{cases} \quad (12)$$

$$f(t) = \begin{cases} 0.225\left[-g^2(t) - g(t)\right] + g(t) & g(t) < 0 \\ 0.225\left[g^2(t) - g(t)\right] + g(t) & g(t) > 0 \end{cases} \quad (13)$$

Fig.5 shows a comparison between equation (4) and the approximative function $f(t)$. The maximum error is 0.0011, which can be neglected.

3.4 The Overflow of Phase Φ_i

Another difficulty of implementing CPG controller in AVR system is that the variables Φ_i will continuously increase with time (see Fig. 2).

The ceaseless ascending of Φ_i will eventually make it too large that the microcontroller fails to manage. Since cosine is a function of period 2π, to solve the aforementioned problem, one way is to replace the value of ϕ_i with $\phi_i - 2\pi$ whenever Φ_i reaches 2π. However, during this calculation of ϕ_i, one of the oscillator phase value is replaced with $\phi_i - 2\pi$, while the other two are kept at their original. As a result, the phase bias φ_{ij} becomes $\varphi_{ij} \pm 2\pi$, which makes the system failed to approach the desired state. So we modify this method considering three oscillator phases together. When the minimum ϕ_i surpass the oscillator's period, we subtract one oscillator's period for all ϕ_i.

In Fig. 6, the upper two sub-figures show the results produced by subtracting 2π from each ϕ_i, whenever ϕ_i exceeds 2π. In this way, phase differences change after periodization. The phase of θ_3 in black line is ahead of that of θ_1 by 1.2π and that of θ_2 by 0.8π. This means φ_{12} is 1.2π and φ_{13} is -0.4π. However, the desired value of φ_{12} is 0.5π and the desired value of φ_{13} is 0.3π. The lower two sub-figures

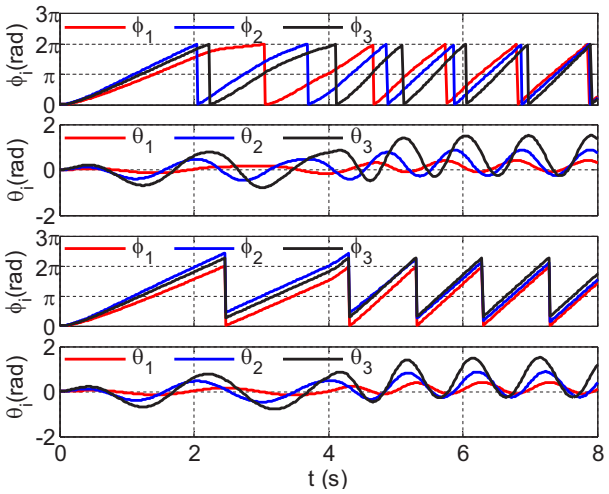
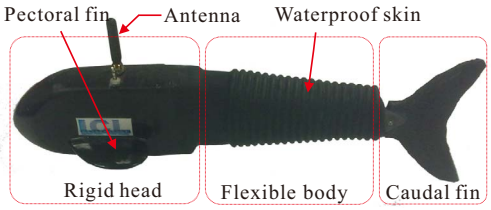
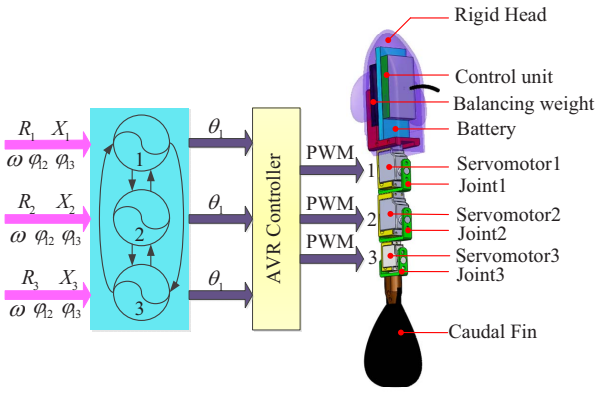


Fig. 6: simulation comparison of the solutions for the overflow, ϕ_i and θ_i respectively represent the phase and the output angle. They all change by the time in second. The system starts at 0 s and changed its parameters at 4 s.



(a) Outside view of the prototype of the robotic fish



(b) Inside view of the robotic fish

Fig. 7: Illustrations of our robotic fish prototype. It is composed of three parts, a rigid head, a flexible body and a elastomeric caudal fin. The insight view shows, three servomotors, a group of batteries and a control circuit board.

illustrate the results produced by subtracting 2π from all ϕ_i, whenever the minimum ϕ_i exceeds 2π. Outputs θ_i are the same with that shown in Fig. 2. Through this way we can find that phase differences keep the same after periodization.

4 Experiments and Tests

In order to evaluate the presented AVR-based CPG controller, a prototype robotic fish and experimental testing procedures have been developed. We designed two groups of

CPG parameters to make robotic fish swim forward and switch to turn right.

Table 1: CPG Parameter Values Applied to The Robotic Fish for Different Locomotion Modal

Parameters	Unit	Swimming Forward	Turning Right
Amplitude of joint one R_1	rad	0.19	0.23
Amplitude of joint two R_2	rad	0.37	0.4
Amplitude of joint three R_3	rad	0.63	0.49
Offset of joint one X_1	rad	0	1.05
Offset of joint two X_2	rad	0	0.96
Offset of joint three X_3	rad	0	0.87
Phase bias between joint one and two φ_{12}	rad	0.698	0.698
Phase bias between joint two and three φ_{13}	rad	2.513	2.513
Frequency ω	Hz	2	0.858

4.1 Prototype of Robotic Fish

The robotic fish (see Fig 7(a)) comprises three parts: a rigid head, a flexible body and a caudal fin. The rigid head is made from fiberglass protecting components inside. In the interior of the flexible body, there are three sequential joints actuated by three servomotors, which are covered with waterproof rubber cloth. The body is succeeded by a caudal fin made of rubber too, but it is thick and rigid.

The inner components of the robotic fish, as illustrated in Fig. 7(b), are elaborated to be compact. An electronic board, a wireless communication module and a battery pack are piled up at the front, while three servomotors are embedded in a skeleton made of aluminium. Embedded in the electronic board, microcontroller communicates with upper computer via wireless duplex communication module. AVR generates Pulse-Width Modulation (PWM) signals to drive three servomotors. Process of CPG controller is shown in Fig 7(b).

4.2 Experimental Results

The experiments were conducted in a $200 \times 300mm$ tank, and there is a global view camera to catch the video of fish swimming. The tests last for $8s$, robotic fish first swims forward $4s$ and then changes to turning model immediately . We made AVR send all the CPG controller values to the upper computer to help us determine whether the AVR controller supports our CPG algorithm. Parameters used in these locomotion pattern testing are set in table 1.

As a result, Fig. 8 illustrates that the AVR-based CPG

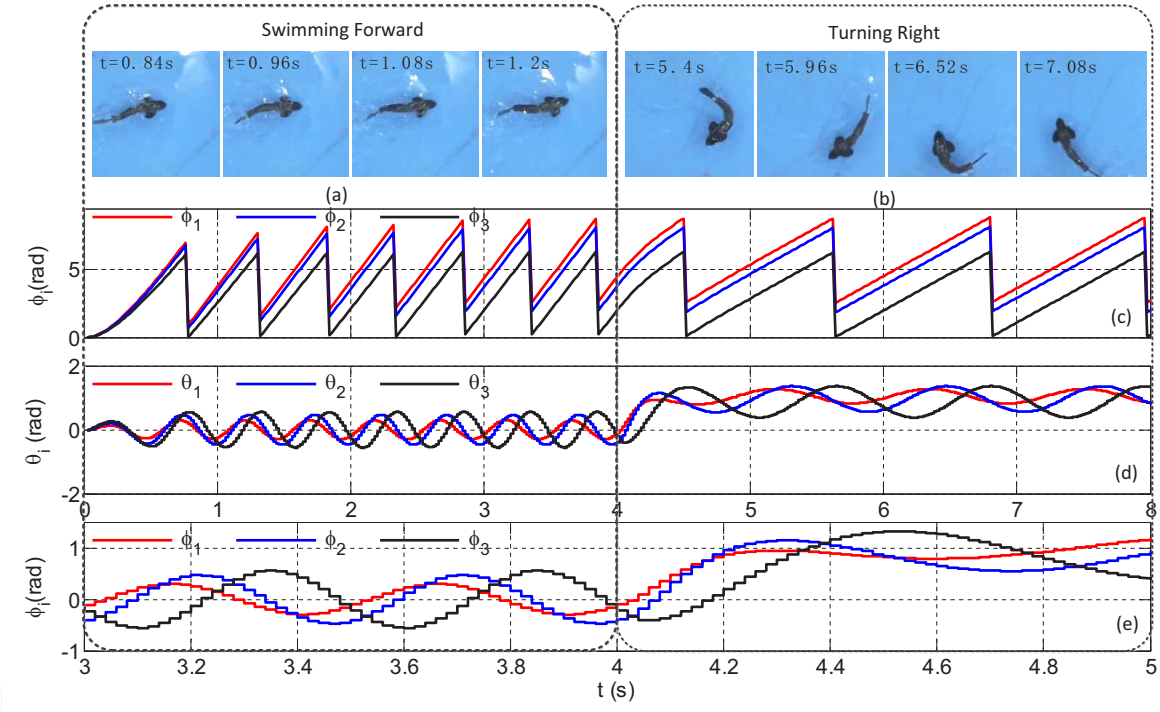


Fig. 8: Snapshots and controlling parameters of the CPG controller embedded robotic fish : (a) snapshot sequence of the robotic fish swimming forward with a speed of $37cm/s$. (b) snapshots sequence of the robotic fish turning right with an angular speed of $5.6rad/s$. (c) The change of CPG parameter ϕ_i , robotic fish starts from static swimming forward and then turning right at $t = 4s$.(d)The change of the AVR-based CPG outputs corresponding to the robotic fish swimming forward before $t = 4s$ and then turning right. (e)The enlarged diagram of the CPG outputs from $t = 3s$ to $t = 5s$.

controller sufficiently controls our robotic fish. The fish can swim forward or turn around under different setting parameters. A snapshot of the sequence is shown in Fig. 8(a), when the fish swims forward in an oscillating period. With the turning mode parameters in table 1 fed to CPG controller, the robotic fish finishes a 180^o turn within 2 oscillating periods (see details Fig. 8(b)). Fig. 8 (c) and (d) describe the CPG parameter variations corresponding to the robotic fish swimming process. All the data values are collected from AVR controller, when the robotic fish swims. The curves can be divided into two part, the first part (from 0-4 s) represents the robotic fish swims forward, and the second part (from $4-8s$) depict the robotic fish changes from swimming forward to turning right. As illustrated in Fig. 8(d), CPG locomotion controller achieves stability within one oscillating period. Transition from swimming straight mode to turning right mode is smooth, which is beneficial to the servomotors. An enlargement of outputs θ_i from $t = 3.0s$ to $t = 5.0s$ is shown in Fig. 8 (e). The value of θ_i, calculated by AVR microcontroller, is discrete and constant within $20ms$.

5 Conclusion and future work

This paper focuses on the methodologies of embedding the CPG controller in AVR system. Euler difference method is used to solve our CPG controller in numerical calculation. High accuracy cosine calculation algorithm for AVR is proposed based on Michael's method. Phase overflow has been overcame without impacting other variables in CPG. Some error analyses are made between the analytical method and the Euler difference method using MATLAB simulations. After solving this series of engineering problems, we developed an AVR based CPG controller and applied it to a robotic fish. Simulations and Experiments validate the effectiveness of our model and implementation architecture.

This CPG-based AVR system is incapable of finding the most suitable parameters. For instance, the robotic fish controlled by the CPG can finish any locomotion task with a given set of parameters. But it can not determine which group of parameters is the most suitable one for the swimming forward action. An adaptive optimization is useful in designing for the solutions of these problems. Researches on the relationship between the parameters and the simple low-dimension control commands need to be further identified and discussed.

References

[1] F. Delcomyn, "Neural basis of rhythmic behavior in animals," *Science*, vol. 210, no. 4469, pp. 492–498, 1980.

[2] M. MacKay-Lyons, "Central pattern generation of locomotion: A review of the evidence," *Physical Therapy*, vol. 82, no. 1, pp. 69–83, 2002.

[3] D. M. Armstrong, "The supraspinal control of mammalian locomotion," *Journal of Physiology-London*, vol. 405, pp. 1–37, 1988.

[4] J. Liu and H. Hu, "Biological inspiration: From carangiform fish to multi-joint robotic fish," *Journal of Bionic Engineering*, vol. 7, no. 1, pp. 35–48, 2010.

[5] Y. P. Ivanenko, R. Grasso, V. Macellari, and F. Lacquaniti, "Control of foot trajectory in human locomotion: Role of ground contact forces in simulated reduced gravity," *Journal of Neurophysiology*, vol. 87, no. 6, pp. 3070–3089, 2002.

[6] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: A review," *Neural Networks*, vol. 21, no. 4, pp. 642–653, 2008.

[7] J. Morimoto, G. Endo, J. Nakanishi, and G. Cheng, "A biologically inspired biped locomotion strategy for humanoid robots: Modulation of sinusoidal patterns by a coupled oscillator model," *Robotics, IEEE Transactions on*, vol. 24, no. 1, pp. 185–191, 2008.

[8] S. Inagaki, H. Yuasa, and T. Arai, "CPG model for autonomous decentralized multi-legged robot systemłgeneration and transition of oscillation patterns and dynamics of oscillators," *Robotics and Autonomous Systems*, vol. 44, no. 3-4, pp. 171–179, 2003.

[9] K. Nakada, T. Asai, and Y. Amemiya, "An analog CMOS central pattern generator for interlimb coordination in quadruped locomotion," *Neural Networks, IEEE Transactions on*, vol. 14, no. 5, pp. 1356–1365, 2003.

[10] A. J. Ijspeert, A. Crespi, D. Ryczko, and J.-M. Cabelguen, "From swimming to walking with a salamander robot driven by a spinal cord model," *Science*, vol. 315, no. 5817, pp. 1416–1420, 2007.

[11] J. Yu, M. Wang, M. Tan, and J. Zhang, "Three-dimensional swimming," *Robotics & Automation Magazine, IEEE*, vol. 18, no. 4, pp. 47–58, 2011.

[12] W. Zhao, Y. H. Hu, and L. Wang, "Construction and central pattern generator-based control of a flipper-actuated turtle-like underwater robot," *Advanced Robotics*, vol. 23, no. 1-2, 2009.

[13] C. Torres-Huitzil and B. Girau, *Implementation of Central Pattern Generator in an* FPGA*-Based Embedded System*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2008, vol. 5164, pp. 179–187.

[14] K. Matsuoka, "Mechanisms of frequency and pattern control in the neural rhythm generators," *Biological Cybernetics*, vol. 56, no. 5, pp. 345–353, 1987.

[15] P. Arena and L. Fortuna, "Collective behaviour in cellular neural networks to model the central pattern generator," *International Journal of Systems Science*, vol. 31, no. 7, pp. 827–841, 2000.

[16] M. G. Felipe, F. Yang, and Z. Yang, "Building artificial CPGs with asymmetric hopfield networks," in *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, vol. 4, 2000, pp. 290–295 vol.4.

[17] J. H. Barron-Zambrano, C. Torres-Huitzil, and B. Girau, "Hardware implementation of a CPG-based locomotion control for quadruped robots." Springer-Verlag, 2010, pp. 276–285.

[18] C. Wang, G. M. Xie, L. Wang, and M. Cao, "CPG-based locomotion control of a robotic fish: Using linear oscillators and reducing control parameters via PSO," *International Journal of Innovative Computing Information and Control*, vol. 7, no. 7B, pp. 4237–4249, 2011.

[19] Atmel Corporation, "8-bit atmel microcontroller with 128kbytes in-system programmable flash." ATmega 128 Datasheet, Jun. 2011.

[20] [Online]. Available: http://lab.polygonal.de/?p=205