Artificial Life and Robotics manuscript No.
(will be inserted by the editor)

Multi-objective optimization for efficient motion of underwater snake robots

E. Kelasidi · M. Jesmani · K. Y. Pettersen · J. T. Gravdahl

Received: date / Accepted: date

Abstract Underwater snake robots constitute a bio-inspired solution within underwater robotics. Increasing the motion efficiency in terms of the forward speed by improving the locomotion methods is a key issue for underwater robots. Furthermore, the energy efficiency is one of the main challenges for long-term autonomy of these systems. In this study, we will consider both these two aspects of efficiency, which in some cases can be conflicting. To this end, we formulate a multi-objective optimization problem to minimize power consumption and maximize forward velocity. In particular, the optimal values of the gait parameters for different motion patterns are calculated in the presence of trade-offs between power consumption and velocity. As is the case with all multi-objective optimization problems, the solution is not a single point but rather a set of points. We present a weighted-sum method to combine power consumption and forward velocity optimization problems. Particle Swarm Optimization (PSO) is applied to obtain optimal gait parameters for different weighting factors. *Trade-off curves* or *Pareto fronts* are illustrated in a power consumption–forward velocity plane for both lateral and eel-like motion pattern. They give information on objective trade-offs and can show how improving power consumption is related to deteriorating the forward velocity along the trade-off curve. Therefore, decision makers can specify the preferred Pareto optimal point along the trade-off curve. Moreover, we address some interesting questions regarding the optimal gait parameters, which is a significant issue for the control of underwater snake robots in the future.

Keywords Underwater snake robot · multi-objective optimization · PSO · energy efficiency

1 Introduction

The use of underwater vehicles has rapidly increased the last decades since these systems are able to operate in deep and high risk areas which humans cannot reach. Nowadays, autonomous underwater vehicles (AUVs) and remotely operated vehicles (ROVs) are widely used in the subsea environment for different challenging tasks [6]. These vehicles are suitable for various work assignments such as inspection, surveillance, maintenance, repairing equipment, building structures, and data collection, and they are extensively used in the subsea oil and gas industry and by the science community. For the long term autonomy of these systems, energy efficiency is one of the main challenges.

For centuries, engineers and scientists have gained inspiration from the natural world in their search for solutions to technical problems, and this process is termed biomimetics. As has been noted in the bio-robotics community, underwater snake robots bring a promising prospective to improve the efficiency and maneuverability of next generation underwater vehicles [1,11,30]. They have several promising applications for underwater exploration, monitoring, surveillance and inspection, and they carry a lot of potential for inspection of subsea oil and gas installations. Also, for the biology and marine archeology communities, snake robots that are able to swim smoothly without much noise, and that can navigate in difficult environments such as ship wrecks, are very interesting [11]. To realize operational snake robots for

E. Kelasidi
Centre for Autonomous Marine Operations and Systems, Dept. of Engineering Cybernetics at NTNU, NO-7491 Trondheim, Norway.
Tel.: +47 73594886
Fax: +47 73594599
E-mail: Eleni.Kelasidi@itk.ntnu.no

M. Jesmani
Dept. of Engineering Cybernetics at NTNU, NO-7491 Trondheim, Norway.

K. Y. Pettersen
Centre for Autonomous Marine Operations and Systems, Dept. of Engineering Cybernetics at NTNU, NO-7491 Trondheim, Norway.

J. T. Gravdahl
Dept. of Engineering Cybernetics at NTNU, NO-7491 Trondheim, Norway.

such underwater applications, a number of different control design challenges must first be solved. An important control problem concerns the ability to achieve efficient motion with preferably a minimum amount of consumed energy in order to be able to undertake longer missions, and this is the topic of this paper.

In [10], the relationships between the gait parameters, the consumed energy and the forward velocity for different motion patterns for underwater snake robots were investigated. In addition, based on simulation studies empirical rules were proposed in order to choose the most efficient motion pattern. Furthermore, in [9], comparison results are obtained for the power consumption of underwater snake robots and ROVs. In particular, it is shown that the biologically inspired swimming robots are more energy efficient compared to the ROVs. These initial simulation studies presented in [9, 10] provide interesting insights regarding the power consumption and the efficiency of underwater snake robots. To our knowledge, however, no research has been published formulating a multi-objective optimization problem considering both the minimization of the power consumption and maximization of the forward velocity. In this paper, we formulate this problem and propose a method to solve the multi-objective optimization problem.

In the motion optimization problem, the mentioned multi-objective function is optimized by iterating on the gait parameters subject to the dynamic model of the underwater swimming robot, which is quite challenging mainly due to the hydrodynamic effects. In [11, 12], the authors propose a model of underwater snake robots, where the dynamic equations are written in closed form. Compared to the models in [1, 4, 21, 24, 27, 30] it is an advantage from an analysis point of view that the model is in closed form, as opposed to including numerical evaluations of the drag effects. In addition, it is beneficial that it includes both resistive and reactive fluid forces, since swimming snake robots operate at Reynolds numbers that require both these effects to be taken into account. In particular, the modeling approach presented in [11, 12] takes into account both the linear and the nonlinear drag forces (resistive fluid forces), the added mass effect (reactive fluid forces), the fluid moments and current effects. This model will thus be used in the optimization.

In order to address optimization of both the power consumption and the forward velocity, a combination of the goals are proposed as a single objective function. Therefore, we formulate the problem as a constrained optimization problem subject to constraints. There exist two types of optimization algorithms to solve this kind of problem: gradient-based methods and derivative free algorithms. Generally, gradient-based algorithms are faster than derivative free ones. However, gradient-based algorithms are susceptible to getting trapped in local optima, meaning that the optimized solution depends on the initial points [18]. Therefore, derivative free and stochastic methods, which have the ability to avoid local solutions, have received attention in problems with highly non-smooth objective functions containing multiple optima. Consequently, as also mentioned in [30], derivative free and stochastic methods are an appropriate choice for the motion optimization. Derivative free algorithms are adopted from researchers in the fields of swimming robots to investigate the efficiency of undulatory locomotion. In [19], a genetic algorithm (GA) is applied to a three link swimmer in order to optimize the swimming gait. Furthermore, a notable study regarding the optimization of swimming gaits is performed in [16], where optimized patterns of anguilliform swimming are investigated using 3D simulations. In particular, a custom evolutionary optimization algorithm combined with a three-dimensional numerical solution is obtained for the swimmer. By using Navier-Stokes equations an optimization problem with two distinct optimization objectives, efficiency and velocity, is solved. Note that the computational cost of solving the Navier-Stokes equations of their model restricts the results to only one optimization run for each fitness goal. In [28], multi-objective optimization was applied for a land-based snake robot to obtain the gait parameters in order to optimize head stability and the speed of the robot simultaneously. Furthermore, in [3] a multi-objective evolutionary algorithm (MOEA) was proposed in order to design and optimize heterogeneous snake-like modular robot. In particular, the MOEA was implemented to maximize the moving behaviour of the robot while minimizing the total number of segments. However, the energy efficiency was not considered neither in [28] nor in [3]. In [2] a probability-based reinforcement learning approach (PI^2) was applied to a land-based snake robot to learn the locomotion control parameters of the robot in order to generate motor primitives and locomotion. In [30], the total required energy was minimized by manipulating the gait parameters and keeping the velocity constant. A penalty term was introduced in the objective function to penalize any deviation of the velocity from its desired value. Therefore, a trade-off between the velocity and the required energy was not addressed in [30]. In this paper, however, we propose to minimize the energy and maximize the velocity, simultaneously.

In this study, the PSO algorithm is applied to solve the optimization problem. The PSO algorithm has received significant attention during the past decades [13]. The idea of the PSO algorithm is inspired from the social behavior of animals such as bird flocking. PSO is initialized with a population of random solutions, called particles. Each particle is assigned a random velocity according to the experiences of the particle itself and its neighborhoods. As opposed to GA, PSO has memory and the knowledge of good solutions is retained during generations. Moreover, in PSO particles share the information and speed up the convergence by a mechanism of constructive cooperation. Recently, application of Reinforcement Learning (RL) method on robotic domain has received attention [17]. RL is a learning method which uses interaction with environment to maximize the expected cumulative reward for an agent. Both RL and PSO methods require many function evaluation to converge. However, parallel implementation of PSO may reduce the computational time significantly. A comparison between PSO and RL methods for multi-robot obstacle avoidance can be found in [5], which shows the highest fitness of PSO.

In this paper, based on the dynamic model presented in [11, 12], a multi-objective optimization problem is developed with the aim of maximizing the achieved forward velocity of the robot and minimizing the corresponding average power consumption of the system. Results are obtained for the two most common swimming patterns for underwater snake robot locomotion: lateral undulation and eel-like motion patterns. To the authors' best knowledge, investigation of efficient motion patterns by solving a multi-objective optimization problem has not been considered in previous literature. Furthermore, please note that the proposed optimization framework is applied to obtain the parameters of the most efficient motion pattern, which can be used in the future for control and design of underwater snake robots.

The paper is organized as follows. Sect. 2 presents the dynamic model of an underwater snake robot. The multi-objective optimization framework is presented in Sect. 3, followed by results obtained for an underwater snake robot in Sect. 4. Finally, conclusions and suggestions for further research are given in Sect. 5.

2 Dynamic Model

This section briefly presents the model of the kinematics and dynamics of an underwater snake robot moving in a virtual horizontal plane. A more detailed presentation of the model can be found in [11], [12].

2.1 Notations and Defined Symbols

The underwater snake robot consists of n rigid links of equal length $2l$ interconnected by $n-1$ joints. The links are assumed to have the same mass m and moment of inertia $J = \frac{1}{3}ml^2$. The mass of each link is uniformly distributed so that the link CM (center of mass) is located at its center point (at length l from the joint at each side). The total mass of the robot is therefore nm. The following vectors and matrices are used in the subsequent sections:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & & \\ & \ddots & \ddots & \\ & & 1 & 1 \end{bmatrix}, \mathbf{D} = \begin{bmatrix} 1 & -1 & & \\ & \ddots & \ddots & \\ & & 1 & -1 \end{bmatrix},$$

where $\mathbf{A}, \mathbf{D} \in \mathbb{R}^{(n-1)\times n}$. Furthermore,

$$\mathbf{e} = \begin{bmatrix} 1, \ldots, 1 \end{bmatrix}^T \in \mathbb{R}^n, \mathbf{E} = \begin{bmatrix} \mathbf{e} & \mathbf{0}_{n\times 1} \\ \mathbf{0}_{n\times 1} & \mathbf{e} \end{bmatrix} \in \mathbb{R}^{2n\times 2},$$

$$\mathbf{S}_\theta = \mathrm{diag}(\sin\theta) \in \mathbb{R}^{n\times n}, \qquad \mathbf{C}_\theta = \mathrm{diag}(\cos\theta) \in \mathbb{R}^{n\times n}$$
$$\dot{\theta}^2 = \left[\dot{\theta}_1{}^2, \ldots, \dot{\theta}_n{}^2\right]^T \in \mathbb{R}^n, \mathbf{K} = \mathbf{A}^T \left(\mathbf{D}\mathbf{D}^T\right)^{-1}\mathbf{D}.$$

2.2 Kinematics of Underwater Snake Robot

The snake robot is assumed to move in a virtual horizontal plane, fully immersed in water, and has n+2 degrees of freedom (n links angles and the x-y position of the robot). The *link angle* of each link $i \in 1,\ldots,n$ of the snake robot is denoted by $\theta_i \in \mathbb{R}$, while the *joint angle* of joint $i \in 1,\ldots,n-1$ is given by $\phi_i = \theta_i - \theta_{i-1}$. The link angles and the joint angles are assembled in the vectors $\theta = [\theta_1,\ldots,\theta_n]^T \in \mathbb{R}^n$ and $\phi = [\phi_1,\ldots,\phi_{n-1}]^T \in \mathbb{R}^{n-1}$, respectively. The *heading* (or *orientation*) $\bar{\theta} \in \mathbb{R}$ of the snake is defined as the average of the link angles, i.e. as $\bar{\theta} = \frac{1}{n}\sum_{i=1}^n \theta_i$ [22]. The global frame position $\mathbf{p}_{\mathrm{CM}} \in \mathbb{R}^2$ of the CM (center of mass) of the robot is given by

$$\mathbf{p}_{\mathrm{CM}} = \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} \frac{1}{nm}\sum_{i=1}^n mx_i \\ \frac{1}{nm}\sum_{i=1}^n my_i \end{bmatrix} = \frac{1}{n}\begin{bmatrix} \mathbf{e}^T\mathbf{X} \\ \mathbf{e}^T\mathbf{Y} \end{bmatrix}, \tag{1}$$

where (x_i, y_i) are the global frame coordinates of the CM of link i, $\mathbf{X} = [x_1,\ldots,x_n]^T \in \mathbb{R}^n$ and $\mathbf{Y} = [y_1,\ldots,y_n]^T \in \mathbb{R}^n$.

2.3 Hydrodynamic Modeling

The dynamic modeling of the contact forces is quite complicated compared to the modeling of the overall rigid motion. In [11], it is shown that the fluid forces and torques on all links can be expressed in vector form as

$$\mathbf{f} = \begin{bmatrix} \mathbf{f_x} \\ \mathbf{f_y} \end{bmatrix} = \begin{bmatrix} \mathbf{f_{A_x}} \\ \mathbf{f_{A_y}} \end{bmatrix} + \begin{bmatrix} \mathbf{f^{I}_{D_x}} \\ \mathbf{f^{I}_{D_y}} \end{bmatrix} + \begin{bmatrix} \mathbf{f^{II}_{D_x}} \\ \mathbf{f^{II}_{D_y}} \end{bmatrix}, \tag{2}$$

and

$$\tau = -\Lambda_1\ddot{\theta} - \Lambda_2\dot{\theta} - \Lambda_3\dot{\theta}|\dot{\theta}|, \tag{3}$$

respectively, where the vectors $\mathbf{f_{A_x}}$ and $\mathbf{f_{A_y}}$ represent the effects from added mass forces and the vectors $\mathbf{f^{I}_{D_x}}$, $\mathbf{f^{I}_{D_y}}$ and $\mathbf{f^{II}_{D_x}}$, $\mathbf{f^{II}_{D_y}}$ represent the linear and nonlinear drag forces, respectively. The detailed derivation of the vectors $\mathbf{f_{A_x}}$, $\mathbf{f_{A_y}}$, $\mathbf{f^{I}_{D_x}}$, $\mathbf{f^{I}_{D_y}}$, $\mathbf{f^{II}_{D_x}}$, $\mathbf{f^{II}_{D_y}}$ and the matrices Λ_1, Λ_2 and Λ_3 can be found in [11].

2.4 Equations of Motion

This section presents the equations of motion for the underwater snake robot. In [11,12] it is shown that the acceleration of the CM may be expressed as

$$\begin{aligned} \begin{bmatrix} \ddot{p}_x \\ \ddot{p}_y \end{bmatrix} &= -\mathbf{M}_p \begin{bmatrix} \mathbf{k}_{11} & \mathbf{k}_{12} \\ \mathbf{k}_{21} & \mathbf{k}_{22} \end{bmatrix} \begin{bmatrix} l\mathbf{K}^T(\mathbf{C}_\theta\dot{\theta}^2 + \mathbf{S}_\theta\ddot{\theta}) \\ l\mathbf{K}^T(\mathbf{S}_\theta\dot{\theta}^2 - \mathbf{C}_\theta\ddot{\theta}) \end{bmatrix} \\ &\quad -\mathbf{M}_p \begin{bmatrix} \mathbf{k}_{12} & -\mathbf{k}_{11} \\ \mathbf{k}_{22} & -\mathbf{k}_{21} \end{bmatrix} \begin{bmatrix} \mathbf{V}^a_x \\ \mathbf{V}^a_y \end{bmatrix} \dot{\theta} + \mathbf{M}_p \begin{bmatrix} \mathbf{e}^T\mathbf{f_{Dx}} \\ \mathbf{e}^T\mathbf{f_{Dy}} \end{bmatrix}, \end{aligned} \tag{4}$$

where the detailed derivation of the matrix $\mathbf{M}_p$ and vectors $\mathbf{V}^a_x$, $\mathbf{V}^a_y$, $\mathbf{k}_{11}$, $\mathbf{k}_{12}$, $\mathbf{k}_{21}$and $\mathbf{k}_{22}$ is given in [11,12]. In addition, it is shown that under the influence of fluid forces (2) and torques (3), the complete equation of motion of the underwater snake robot are obtained by (4) and

$$\mathbf{M}_\theta \ddot{\theta} + \mathbf{W}_\theta \dot{\theta}^2 + \mathbf{V}_\theta \dot{\theta} + \Lambda_3 |\dot{\theta}| \dot{\theta} + \mathbf{K_{Dx}} \mathbf{f_{Dx}} + \mathbf{K_{Dy}} \mathbf{f_{Dy}} = \mathbf{D}^T \mathbf{u}, \tag{5}$$

where $\mathbf{f_{Dx}} = \mathbf{f}^{\mathrm{I}}_{\mathbf{D_x}} + \mathbf{f}^{\mathrm{II}}_{\mathbf{D_x}}$ and $\mathbf{f_{Dy}} = \mathbf{f}^{\mathrm{I}}_{\mathbf{D_y}} + \mathbf{f}^{\mathrm{II}}_{\mathbf{D_y}}$ representing the drag forces in x and y directions and $\mathbf{u} \in \mathbb{R}^{n-1}$ the control input. For more details and the derivation of the matrices $\mathbf{M}_\theta$, $\mathbf{W}_\theta$, $\mathbf{V}_\theta$, $\mathbf{K_{Dx}}$ and $\mathbf{K_{Dy}}$, see [12].

Remark 1. It is interesting to note that if, in the dynamic model (4) and (5), we set the fluid parameters to zero and replace the drag forces in x and y direction with ground friction models [22], then the model reduces exactly to the dynamic model of a ground snake robot described in [22]. The underwater snake robot model is thus an extension of the land snake robot model, and may be used for amphibious snake robots moving both on land and in water.

3 Optimization of Motion

This section presents an optimization framework to investigate the efficient motion of the underwater snake robot model presented in Sect. 2. We depict the optimization framework in Fig. 1, consisting of the plant (the underwater snake robot model), a joint actuation controller, and an optimizer. The optimizer requires the simulation of the model in order to evaluate the objective function for different gait parameters. The constraints defining the feasible region are also inputs of the optimizer. In the following, the procedures of the joint controller and the optimizer are discussed in detail.

3.1 Joint actuation controller

The joint actuation controller consists of the gait pattern (motion pattern) generator and joint control, which are discussed in this subsection.

Motion Pattern Previous studies on swimming snake robots have focused on two motion patterns; lateral undulation and eel-like motion. In this paper, we will use a general sinusoidal motion pattern that describes a broader class of motion patterns including lateral undulation and eel-like motion [8]. Lateral undulation [22], which is the fastest and most common form of ground snake locomotion, can be achieved by creating continuous body waves, with a constant amplitude, that are propagated backwards from head to tail. In order to achieve lateral undulation, the snake robot is commanded to follow the serpenoid curve as proposed in [7]. Eel-like motion can be achieved by propagating lateral axial undulations with increasing amplitude from head to tail [11]. In this paper, a general sinusoidal motion pattern is achieved by making each joint $i \in \{1, \cdots, n-1\}$ of the underwater snake robot track the sinusoidal reference signal

$$\phi_i^*(t) = \alpha \mathrm{g}(i,n) \sin(\omega t + (i-1)\delta) + \gamma, \tag{6}$$

where α and ω are the maximum amplitude and the frequency, respectively, δ determines the phase shift between the joints, while the function $\mathrm{g}(i,n)$ is a scaling function for the amplitude of joint i which allows (6) to describe a quite general class of sinusoidal functions, including several different snake motion patterns. For instance, $\mathrm{g}(i,n) = 1$ gives lateral undulation, while $\mathrm{g}(i,n) = (n-i)/(n+1)$ gives eel-like motion [11]. The parameter γ is a joint offset coordinate responsible for the direction control [22].

Low-level joint control A PD-controller is used to calculate the joints' actuator torques from the joints' reference angles according to

$$u_i = k_p(\phi_i^* - \phi_i) + k_d(\dot{\phi}_i^* - \dot{\phi}_i), \quad i = 1, \ldots, n-1, \tag{7}$$

where $k_p > 0$ and $k_d > 0$ are the gains of the controller.

3.2 Optimization problem formulation

An optimization problem is the minimization or maximization of a function from all feasible solutions. The goal of motion optimization is to minimize energy consumption and to maximize the forward velocity, simultaneously. This type of optimization problem which involves more than one objective function is known as multi-objective optimization [25]. In the following, both the energy consumption and the forward velocity are formulated and an approach to combine these two goals is introduced.

For underwater snake robots, the propulsion is generated by the motion of the joints and its interaction with the surrounding fluid. The actuator torque input to the joints is thus transformed into a combination of joint motion and energy that is dissipated by the fluid. We assume that the joints are ideal and thus the total amount of energy of the system (E_s) generated by this input is the summation of kinetic energy (E_kinetic) and the energy that is dissipated to the surrounding fluid (E_fluid) [30]. Therefore, the total energy consumption for the propulsion of the robot can be written as

$$E_\mathrm{s} = E_\mathrm{kinetic} + E_\mathrm{fluid} = \int_0^T \left(\sum_{i=1}^{n-1} u_i(t) \dot{\phi}_i(t) \right) \mathrm{d}t, \tag{8}$$

where T is the time that corresponds to a complete swimming cycle, u_i is the actuation torque of joint i given by (7) and $\dot{\phi}_i$ is the joint's angular velocity defined as $\dot{\phi}_i = \dot{\theta}_i - \dot{\theta}_{i-1}$.

For a complete swimming cycle T the average power consumption, P_avg, is calculated as follows

$$P_\mathrm{avg} = \frac{1}{T} \int_0^T \left(\sum_{i=1}^{n-1} u_i(t) \dot{\phi}_i(t) \right) \mathrm{d}t. \tag{9}$$

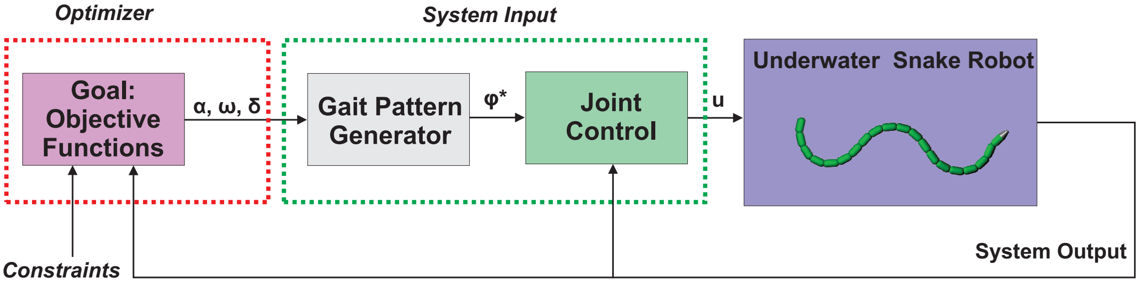


Fig. 1: Illustration of the optimization framework.

The forward velocity for a complete swimming cycle T is defined as

$$\bar{\upsilon} = \frac{\sqrt{(p_x(T)-p_x(0))^2+(p_y(T)-p_y(0))^2}}{T}, \tag{10}$$

where the initial and the final points are used to calculate the travelled distance of the robot.

The optimization problem can be formulated by

$$\min_{\alpha,\omega,\delta} J_{\text{opt}} = [P_{avg}, -\bar{\upsilon}], \tag{11a}$$

$$\text{s.t:} \mid \phi_i^* \mid \leq \phi_i^{max}, \mid \dot{\phi}_i^* \mid \leq \dot{\phi}_i^{max}, \mid u_i \mid \leq u_i^{max}, \tag{11b}$$

$$0 \leq \alpha \leq \alpha^{max}, 0 \leq \omega \leq \omega^{max}, 0 \leq \delta \leq \delta^{max}, \tag{11c}$$

where (11b) gives the physical constraints of the joints due to the servo motors and the physical design of the snake [23], and (11c) expresses the possible range of the parameters of the sinusoidal motion pattern (6).

Generally, in multi-objective optimization such as (11), there does not exist a single global solution to optimize all objective functions simultaneously. Particularly in this case, the objective functions are often in conflict, meaning that maximizing the velocity results in increasing the power consumption and vice versa. A solution is called Pareto optimal [25] of optimization problem (11), if there does not exist another point such that power consumption can be decreased in value without degrading the forward velocity or the forward velocity can be increased without enhancing power consumption. The Pareto frontier or efficient frontier is the collection of Pareto optimal solutions. For bi-objective functions (e.g., (11)), the frontier can be expressed in Cartesian coordinates. An efficient way to generate the Pareto frontier is the weighted-sum method that combines both objective functions in a single criterion function [31]:

$$J_{\text{bal}} = w_p (P_{\text{avg}})_{sc} - w_{\upsilon} (\bar{\upsilon})_{sc}, \tag{12}$$

$$w_p = 1 - w_{\upsilon}, \tag{13}$$

where $(P_{\text{avg}})_{sc}$ and $(\bar{\upsilon})_{sc}$ are scaled values of the power consumption and forward velocity, respectively, and w_p and w_{υ} are the weighting factors of respectively the power consumption and the forward velocity. The solution is always the Pareto frontier, if the coefficients are positive. In order to plot the Pareto frontier, one can find the solutions for different values of w_p varying from 0 to 1 (and w_{υ} changing from 1 to 0) using a small step size Δw_p. Therefore, the optimization problem (11) can be transformed to a new optimization problem by replacing the objective function J_{opt} to J_{bal} in (12). In order to find the solution of the new optimization problem for specific weighting factors, an optimization algorithm is required. In the following PSO is introduced as a proper optimization method for motion optimization.

3.3 PSO algorithm

The PSO algorithm is a population-based probabilistic algorithm first introduced by Kennedy and Eberhart in 1995 [13]. The PSO algorithm exploit a set of potential solutions to the optimization problem. Each potential solution is called a particle, and the set of particles is named a population. The first population is randomly initialized using a random number generator. The location of each particle in the new generation is determined by PSO update equations, which mimic the social behavior of members of bird flocks or fish schools.

Consider an unconstrained nonlinear problem where the objective function $f(\mathbf{x})$ must be minimized. Let $\mathbf{x}_i(k)$ denote the ith particle of the kth generation, and $\nu_i(k)$ represent its velocity. The update equations for all $i \in \{1, 2, \cdots, n_p\}$ and $k \in \mathbb{N}$ are as follows:

$$\begin{aligned} \nu_i(k+1) =& \nu_i(k) + c_1\rho_1(k)(\mathbf{p}_{l,i}(k) - \mathbf{x}_i(k)) \\ &+ c_2\rho_2(k)(\mathbf{p}_{g,i}(k) - \mathbf{x}_i(k)), \end{aligned} \tag{14a}$$

$$\mathbf{x}_i(k+1) = \mathbf{x}_i(k) + \nu_i(k+1). \tag{14b}$$

where $c_1, c_2 \in \mathbb{R}^+$, $\rho_1(k), \rho_2(k) \sim U(0,1)$ are uniformly distributed random numbers between 0 and 1, $\nu_i(0) \triangleq \mathbf{0}$, $\mathbf{p}_{l,i}(k)$ is the best location of the ith particle over all generations, and $\mathbf{p}_{g,i}(k)$ is the location for the best particle over all generations,

$$\mathbf{p}_{l,i}(k) \triangleq \underset{\mathbf{x} \in \{\mathbf{x}_i(j)\}_{j=0}^{k}}{\arg\min} \; f(\mathbf{x}), \tag{15a}$$

$$\mathbf{p}_{g,i}(k) \triangleq \underset{\mathbf{x} \in \{\{\mathbf{x}_i(j)\}_{j=0}^{k}\}_{i=1}^{np}}{\arg\min} \; f(\mathbf{x}). \tag{15b}$$

Therefore, c_1 and c_2 are named cognitive and social acceleration terms, respectively. The objective function, $f(\mathbf{x})$, defined in (12) as J_{bal}, is a weighted sum of the average power consumption and the forward velocity. Therefore, the PSO algorithm minimizes the average power consumption while it maximizes the forward velocity by manipulating the gait parameters α, ω and δ. For each set of weights, the PSO algorithm obtains different optimal values of the gait parameters. We vary the weights in the different optimization runs in order to obtain different Pareto optimal points.

In this paper, we use the *von Neumann* neighborhood topology which has the best performance among other topologies such as *lbest* and *gbest* [15]. In the von Neumann neighborhood topology, each particle has 4 neighborhoods which are defined by special enumeration.

The performance of the PSO algorithm can be improved by introducing an inertia weight. In this case for $\mathbf{x}_i(k) \in \mathbb{R}^{n_c}$ the velocity equation (14a) is replaced by

$$\begin{aligned}\hat{v}_i(k+1) =& w(k)v_i(k) + c_1\rho_1(k)(\mathbf{p}_{l,i}(k) - \mathbf{x}_i(k)) \\ &+ c_2\rho_2(k)(\mathbf{p}_{g,i}(k) - \mathbf{x}_i(k)), \qquad (16a)\end{aligned}$$

$$\begin{aligned}v_i^j(k+1) =& \text{sign}(\hat{v}_i^j(k+1)) \min\{|\hat{v}_i^j(k+1)|, v_{max}^j\}, \\ & j \in \{1,2,\cdots,n_c\}, \qquad (16b)\end{aligned}$$

$$v_{max}^j = \lambda(u_{iw}^j - l_{iw}^j), \qquad (16c)$$

$$w(k) = w_0 - \frac{k}{K}(w_0 - w_1). \qquad (16d)$$

where the maximum velocity gain λ is a scalar, $\mathbf{l}_{iw}, \mathbf{u}_{iw} \in \mathbb{R}^{n_c}$ are, respectively, the lower and upper bound of the independent variables, w_0, w_1 are the initial and final inertia weights, respectively, K is the maximum number of generations.

Remark 2: As stated before, the optimizer only requires the plant to evaluate the objective function. Therefore, the proposed multi-objective optimization framework can be considered as a general tool for investigating motion efficiency of different dynamic models of swimming snake robots controlled by sinusoidal motion patterns such as (6).

Remark 3: Note that the cost of transportation (COT) metric could be considered as an alternative objective function to obtain efficient locomotion of a robot. The minimum COT would then be obtained at some speed at which the robot can cover the given distance with the least energy expenditure. However, using COT as the objective function will not provide optimal values of the gait parameters for different velocities. Multi-objective optimization with the proposed objective function provides a guideline for the optimal values of the gait parameters for different values of the achieved forward velocity and the average power consumption. Hence, these optimal values of the gait parameters are available to the operator of the robot, something which would not be possible to obtain by using COT as the objective function.

4 Optimization Study

In this section, optimization results are presented for both lateral undulation and eel-like motion. The dynamic model presented in Sect. 2 was implemented in *Matlab R2013b*. The time evolution was calculated using the *ode23tb* solver with a relative and absolute error tolerance of 10^{-4}. The PSO is implemented using GenOpt which is developed by Lawrence Berkeley National Laboratory and is written in java. GenOpt runs multiple simulations in parallel to reduce computation time and allows using any simulation software to evaluate the cost function [29].

4.1 Parameters of the underwater snake robot

We consider an underwater snake robot with $n = 10$ links, each one having length $2l = 0.18$ m and mass $m = 0.8$ kg, i.e. identical to the physical robot Mamba presented in [23]. The initial values of the states of the snake robot were set to initial reference values at $t = 0$. The hydrodynamic related parameters c_t, c_n, μ_n λ_1, λ_2 and λ_3 for the elliptic section with major and minor diameters $2a = 2 \cdot 0.055$ m and $2b = 2 \cdot 0.05$ m, respectively, $\rho = 1000$ kg/m^3 and for the fluid coefficients set to $C_f = 0.03$, $C_D = 2$, $C_A = 1$, $C_M = 1$ were calculated by using equations derived in [11]. An extensive discussion about the values of the fluid parameters can be found in [11]. In these simulations the joint PD-controller (7) was used with parameters $k_p = 20$, $k_d = 5$, while lateral undulation or eel-like motion were achieved by moving the joints according to (6) by choosing $\text{g}(i,n) = 1$ and $\text{g}(i,n) = (n-i)/(n+1)$, respectively. Note that current effects have not been considered in this study.

4.2 Optimization parameters

The PSO parameters are given in Table 1. It was suggested in [26] to use a population size of about $5n_c$ for $\mathbf{x}_i(k) \in \mathbb{R}^{n_c}$, where $n_c = 3$ in this case study. Different numbers of generations are tested and we conclude that PSO finds the optimal point in less than 20 generations. The inertia weights are chosen based on the suggestion in [20], the maximum of velocity gain, λ is commonly set to 0.5 [29] and the accelerations are chosen such that $c_1 + c_2 = 4$ [14].

The values of the physical constraints of the joints due to the servo motors in (11b) are set to $u_i^{max} = 2300$ Nm, $\phi_i^{max} = 90^o$, $\dot{\phi}_i^{max} = 429^o/\text{s}$, while the range of the parameters of sinusoidal motion pattern in (11c) are set to $\alpha^{max} = 90^o$, $\omega^{max} = 210^o/\text{s}$ and $\delta^{max} = 90^o$. Furthermore, in this initial study the join offset is set $\gamma = 0$.

The step size for changing the weights, Δw_p, is equal to 0.05. In the first step, we optimize the velocity regardless of the power consumption which means $w_p = 0, w_\upsilon = 1$ in (13), and the maximum values of the velocity and the power consumption are obtained and can be used to scale the objective functions for the next sets of weights. In the next step, we start to reduce the weight of the velocity w_υ, while increasing the power consumption weight w_p. Note that the initial value of PSO in the first step are defined based on expert

number of particles	n_p	16
number of generations	n_g	20
cognitive acceleration	c_1	2.8
social acceleration	c_2	1.2
max velocity gain	λ	0.5
initial inertia weight	w_0	1.2
final inertia weight	w_1	0

Table 1: PSO parameters.

knowledge, and the optimal value of each step is used as the initial guess of the next step. For instance, the initial value of the optimization problem where $w_p = 0.05, w_\upsilon = 0.95$ is the optimal value of the first step optimization problem where $w_p = 0, w_\upsilon = 1$.

In this study, PSO was implemented within a distributed computing framework consisting of 12 computing cores. In order to obtain a Pareto optimal point, $n_p \times n_g = 16 \times 20 = 320$ simulation runs are required. Distributing the simulation runs among 12 nodes results in a speedup factor larger than one. For this case study, the computation time to obtain a Pareto optimal point was about 65 minutes.

4.3 Results

In this section, the results of the proposed multi-objective optimization is presented for both lateral undulation and eel-like motion patterns for the underwater snake robot with parameters as shown in Sect. 4.1; and the optimal forward velocity, $\bar{\upsilon}$, and the optimal average power consumption, P_{avg}, are illustrated with the corresponding gait parameters.

Pareto fronts are presented in Fig. 2-3 for lateral undulation and eel-like motion pattern, respectively. One can observe, as it was expected, that the maximum power is consumed in the case of achieving maximum velocity, and minimum power consumption (equal to zero) occurs for zero forward velocity. In the case of the motionless condition, the objective function is to minimize the power consumption regardless of the velocity ($w_p = 1, w_\upsilon = 0$); thus the optimal gait pattern parameter $\alpha = 0$, and two other parameters ω and δ are not effective. The underwater snake robot investigated in this study achieved maximum forward velocity $\bar{\upsilon} = 0.84$ m/s and $\bar{\upsilon} = 0.60$ m/s for lateral undulation and eel-like motion, respectively, with the corresponding maximum average power consumption of $P_{avg} = 34.25$ W and $P_{avg} = 13.44$ W.

Furthermore, the Pareto front presented in Fig. 2 illustrates that the average power consumption of the robot can be decreased significantly from $P_{avg} = 34.25$ W to $P_{avg} = 18.92$ W by slightly decreasing the achieved forward velocity from $\bar{\upsilon} = 0.84$ m/s to $\bar{\upsilon} = 0.81$ m/s for the lateral undulation. This means that a 44.76% decrease of the average power consumption can be obtained, while the forward velocity is only reduced by 3.57%. For the eel-like motion pattern, Fig. 3 shows that the power consumption is decreased from $P_{avg} = 13.44$ W to $P_{avg} = 7.88$ W, if we choose to pay a marginal penalty by decreasing the forward velocity from $\bar{\upsilon} = 0.60$m/s to $\bar{\upsilon} = 0.58$m/s. Therefore, the average power consumption can be reduced by 41.37% in the eel-like motion pattern while the corresponding forward velocity is decreased only by 3.33%. Therefore, decision makers can choose the optimal gait pattern parameters such that a significant reduction of the power consumption only results in a slight reduction of the forward velocity.

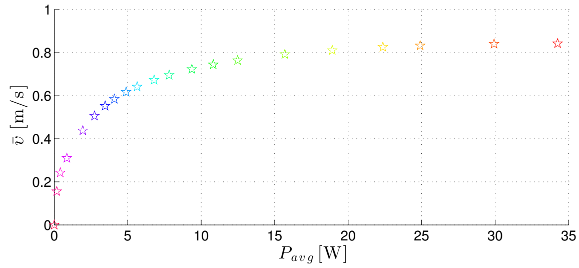


Fig. 2: Pareto front for lateral undulation.

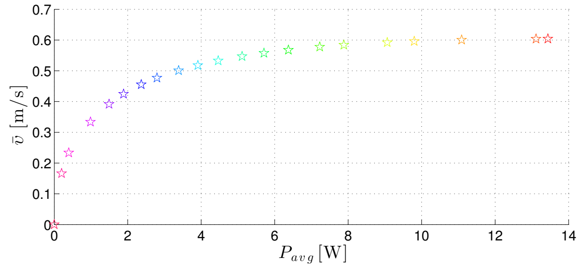
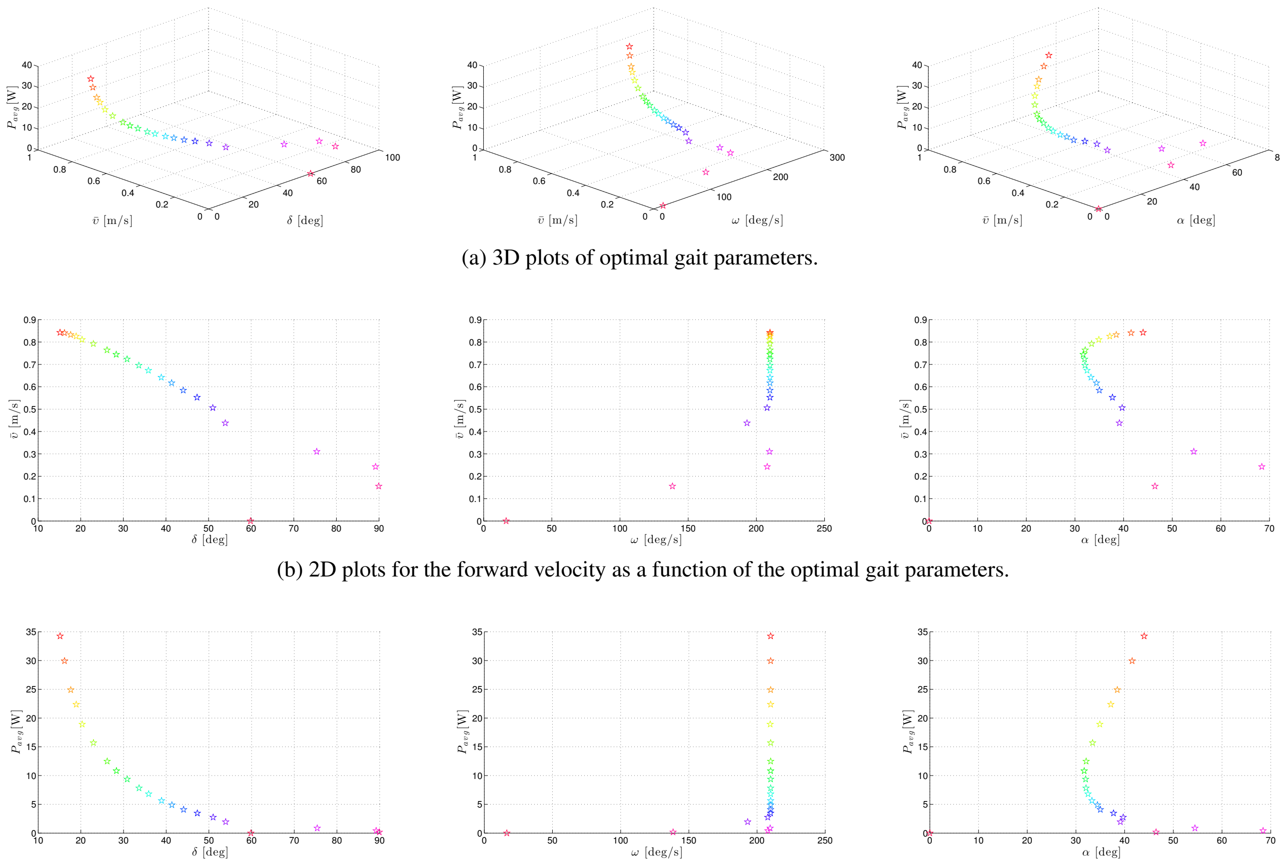


Fig. 3: Pareto front for eel-like motion pattern.

Remark 4: Note that the Pareto fronts shown in Fig. 2-3 can be considered as a useful tool to make the trade-off between the power consumption and the forward velocity. Therefore, based on the Pareto front, a proper set of gait parameters of the swimming robot can be chosen considering the requirement of the control strategies and the available power of the system.

Fig. 4a-5a show how the average power consumption and the forward velocity vary for different optimal values of the gait parameters in (6) for lateral undulation and eel-like motion patterns, respectively. In addition, the 2D projections of the 3D plots presented in Fig. 4a- 5a, for forward velocity and power consumption, are presented in Fig. (4b,5b) and Fig. (4c,5c) for the investigated motion patterns. As was expected, the maximum velocity is achieved for the set of weighting factor w_υ=1 and $w_p = 0$. The maximum forward velocity $\bar{\upsilon} = 0.84$ m/s is achieved for the gait parameters $\alpha = 44.01^o$, $\omega = 210^o$/s and $\delta = 15.14^o$ for lateral undulation, while for eel-like motion pattern the robot achieves maximum forward velocity $\bar{\upsilon} = 0.60$ m/s for the gait parameters $\alpha = 59.24^o$, $\omega = 209.98^o$/s and $\delta = 26.20^o$. The maximum velocity is achieved when ω is at the maximum



(a) 3D plots of optimal gait parameters.

(b) 2D plots for the forward velocity as a function of the optimal gait parameters.

(c) 2D plots for the average power consumption as a function of the optimal gait parameters.

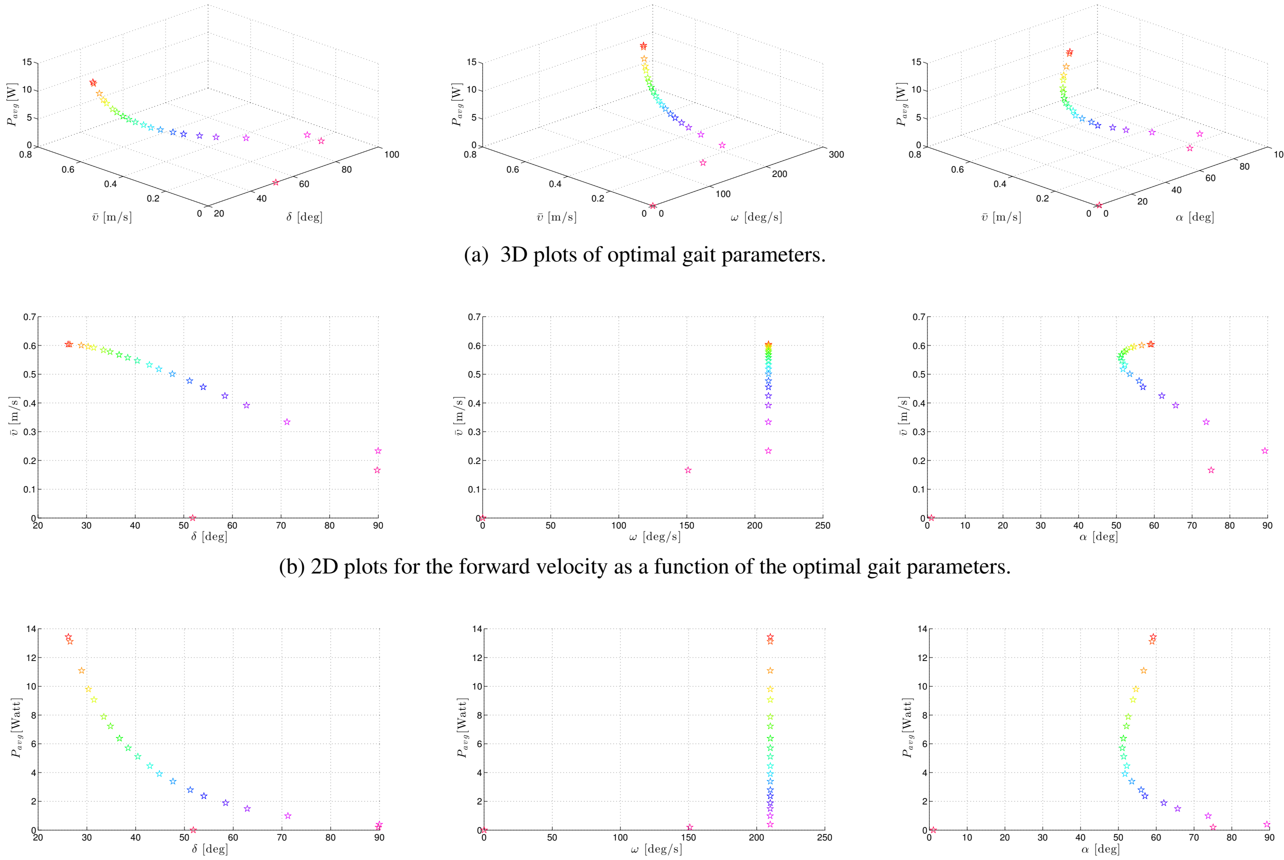
Fig. 4: Optimal solution of lateral undulation motion pattern.

value for both patterns, while the optimal values of α and δ, however, are different for each pattern. Fig. (4b,5b) and Fig. (4c,5c) show that an increase of the parameter δ results in a decrease of the forward velocity and power consumption. Furthermore, the optimal value of parameter α is greater than 30^o for lateral undulation and greater than 50^o for eel-like motion pattern in all weighting sets. These observations are important not only for control purposes but also for formulating the constraints for further optimization investigations in future. Table 2 and Table 3 show the obtained values for the gait parameters α, ω and δ for each point of the Pareto front in Fig. 4 and Fig. 5 for the lateral undulation and eel-like motion pattern, respectively. Fig. 6 and Fig. 7 present the body shape of the robot for each point of the Pareto front for the lateral undulation and eel-like motion pattern, respectively.

Remark 5: By setting the parameter ω to the maximum value, as is the case for almost all Pareto optimal points, we can reduce the dimension of the search space, n_c, to 2. In this case, the parameter ω is eliminated from the optimization problem and the parameters α and δ are the optimization decision variables. In addition, choosing the maximum possible value for the parameter ω provides motivation for high frequency actuation solutions for future work in the control and design of swimming robots.

5 Conclusions

In this paper, we developed an effective multi-objective optimization scheme to obtain optimal gait parameters for underwater snake robots. The proposed optimization method constitutes a general tool to investigate the motion efficiency of different dynamic models of swimming snake robots controlled by sinusoidal motion patterns. PSO was applied to obtain the Pareto optimal gait parameters in the presence of trade-offs between the the power consumption and the forward velocity. Pareto fronts showed how improving efficiency with respect to the power consumption is related to deteriorating efficiency in terms of the forward velocity along the trade-off curve. Decision makers can consider the Pareto front as an informative tool to specify the preferred Pareto optimal point. Moreover, some interesting insights about the optimal swimming gait parameters, which are significant for the control and design of underwater snake robot, were obtained. In the future, the efficiency of other sinusoidal motion patterns will be investigated. Furthermore, other



(a) 3D plots of optimal gait parameters.

(b) 2D plots for the forward velocity as a function of the optimal gait parameters.

(c) 2D plots for the average power consumption as a function of the optimal gait parameters.

Fig. 5: Optimal solution of eel-like motion pattern.

Table 2: Lateral undulation: Results from multi-optimization study for the underwater snake robot.

w_p	α [deg]	ω [deg/s]	δ [deg]	$\bar{v}$ [m/s]	P_{avg} [W]
0	44.0100	210.0000	15.1400	0.8425	34.2515
0.0500	41.5451	209.9863	16.1863	0.8407	29.9318
0.1000	38.5042	209.9987	17.6406	0.8327	24.9031
0.1500	37.1659	209.9976	18.9259	0.8259	22.3614
0.2000	34.9161	209.6835	20.3226	0.8107	18.9214
0.2500	33.4248	209.9899	22.9651	0.7922	15.6914
0.3000	32.1065	209.9997	26.1707	0.7640	12.4822
0.3500	31.7042	209.9932	28.3385	0.7444	10.8197
0.4000	31.9861	209.9938	30.8753	0.7233	9.3646
0.4500	32.0642	209.9931	33.6360	0.6956	7.8220
0.5000	32.4826	209.9995	35.8916	0.6731	6.7981
0.5500	33.2925	209.9890	38.9056	0.6417	5.6396
0.6000	34.4218	209.9967	41.3291	0.6170	4.8916
0.6500	35.0257	209.9667	44.0756	0.5842	4.0883
0.7000	37.7160	209.9899	47.2997	0.5523	3.4617
0.7500	39.7087	207.8852	50.9907	0.5060	2.7433
0.8000	39.1360	193.1456	53.9248	0.4379	1.9467
0.8500	54.4381	209.6399	75.3818	0.3102	0.8571
0.9000	68.4280	207.9591	89.1905	0.2425	0.4111
0.9500	46.4469	138.4423	89.9360	0.1555	0.1693
1.0000	0	16.5134	59.8411	0	0

Table 3: Eel-like motion: Results from multi-optimization study for the underwater snake robot.

w_p	α [deg]	ω [deg/s]	δ [deg]	$\bar{v}$ [m/s]	P_{avg} [W]
0	59.2350	209.9813	26.1994	0.6038	13.4369
0.0500	58.9029	209.9887	26.5538	0.6037	13.1134
0.1000	56.7018	209.9712	28.9106	0.6004	11.0886
0.1500	54.6536	209.9960	30.3223	0.5960	9.7995
0.2000	53.8984	209.9943	31.4665	0.5922	9.0650
0.2500	52.6137	209.9798	33.4673	0.5838	7.8826
0.3000	52.1308	209.9987	34.8301	0.5776	7.2274
0.3500	51.3303	209.9852	36.6885	0.5675	6.3739
0.4000	51.0358	209.9947	38.4587	0.5576	5.7077
0.4500	51.4150	209.9983	40.4453	0.5469	5.1124
0.5000	52.2181	209.9899	42.9157	0.5327	4.4629
0.5500	51.7512	209.9984	44.8704	0.5178	3.9089
0.6000	53.5402	209.9995	47.6477	0.5007	3.3852
0.6500	56.0090	209.9980	51.2005	0.4769	2.7965
0.7000	57.0378	209.9775	53.9967	0.4552	2.3694
0.7500	62.0106	209.9535	58.4522	0.4245	1.8881
0.8000	65.6760	209.9181	62.8889	0.3915	1.4920
0.8500	73.7492	209.8543	71.2193	0.3337	0.9857
0.9000	89.2687	209.9100	89.9740	0.2334	0.3958
0.9500	75.0664	150.9382	89.7837	0.1663	0.2004
1.0000	1.0695	0.0639	51.8275	0	0

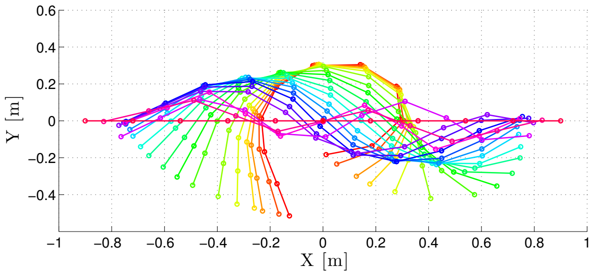


Fig. 6: Body shape for lateral undulation.

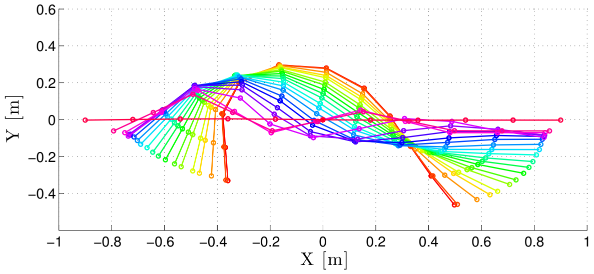


Fig. 7: Body shape for the eel-like motion pattern.

design parameters of underwater snake robots (e.g., the number of the links) will be considered in the optimization variables.

Acknowledgements Research partly funded by VISTA - a basic research program in collaboration between The Norwegian Academy of Science and Letters, and Statoil, and partly supported by the Research Council of Norway through its Centres of Excellence funding scheme, project no. 223254-NTNU AMOS.

References

1. Boyer, F., Porez, M., Khalil, W.: Macro-continuous computed torque algorithm for a three-dimensional eel-like robot. IEEE Transactions on Robotics **22**(4), 763 –775 (2006)
2. Chatterjee, S., Nachstedt, T., Tamosiunaite, M., Wörgötter, F., Enomoto, Y., Ariizumi, R., Matsuno, F., Manoonpong, P.: Learning and chaining of motor primitives for goal-directed locomotion of a snakelike robot with screw-drive units. International Journal of Advanced Robotic Systems **12** (2015)
3. Chee, W.S., Teo, J., Kinabalu, K.: Empirically comparing three multi-objective optimization approaches for the automated evolution of snake-like modular robots. In: Proc. The International Conference on Artificial Intelligence and Pattern Recognition (AIPR), pp. 175–183. Malaysia (2014)
4. Chen, J., Friesen, W.O., Iwasaki, T.: Mechanisms underlying rhythmic locomotion: bodyfluid interaction in undulatory swimming. The Journal of Exp. Biology **214**(4), 561–574 (2011)
5. Di Mario, E., Talebpour, Z., Martinoli, A.: A comparison of PSO and Reinforcement Learning for multi-robot obstacle avoidance. In: Proc. IEEE Congress on Evolutionary Computation (CEC), pp. 149–156. Cancun (2013). DOI 10.1109/CEC.2013.6557565
6. Fossen, T.I.: Handbook of Marine Craft Hydrodynamics and Motion Control. John Wiley & Sons, Ltd (2011)
7. Hirose, S.: Biologically Inspired Robots: Snake-Like Locomotors and Manipulators. Oxford University Press (1993)
8. Kelasidi, E., Pettersen, K.Y., Gravdahl, J.T.: Stability analysis of underwater snake robot locomotion based on averaging theory. In: Proc. IEEE International Conference on Robotics and Biomimetics (ROBIO). Bali, Indonesia (2014). DOI 10.1109/ROBIO.2014.7090392
9. Kelasidi, E., Pettersen, K.Y., Gravdahl, J.T.: Energy efficiency of underwater robots. In: Proc. 10th IFAC Conference on Manoeuvring and Control of Marine Craft (MCMC), pp. 152–159. Copenhagen, Denmark (2015). DOI 10.1016/j.ifacol.2015.10.273
10. Kelasidi, E., Pettersen, K.Y., Gravdahl, J.T.: Energy efficiency of underwater snake robot locomotion. In: Proc. 23th Mediterranean Conference on Control Automation (MED), pp. 1124–1131. Torremolinos, Spain (2015). DOI 10.1109/MED.2015.7158907
11. Kelasidi, E., Pettersen, K.Y., Gravdahl, J.T., Liljebäck, P.: Modeling of underwater snake robots. In: Proc. IEEE International Conference on Robotics and Automation (ICRA), pp. 4540–4547. Hong Kong, China (2014). DOI 10.1109/ICRA.2014.6907522
12. Kelasidi, E., Pettersen, K.Y., Liljebäck, P., Gravdahl, J.T.: Integral line-of-sight for path-following of underwater snake robots. In: Proc. IEEE Multi-Conference on Systems and Control, pp. 1078 – 1085. Juan Les Antibes, France (2014). DOI 10.1109/CCA.2014.6981478
13. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proc. IEEE International Conference on Neural Networks, pp. 1942–1948. Perth, WA (1995). DOI 10.1109/ICNN.1995.488968
14. Kennedy, J., Eberhart, R.C., Shi, Y.: Swarm Intelligence. Morgan Kaufmann Publishers (2001)
15. Kennedy, J., Mendes, R.: Population structure and particle swarm performance. In: Proc. Congress on Evolutionary Computation (CEC), pp. 1671–1676. Honolulu, HI (2002). DOI 10.1109/CEC.2002.1004493
16. Kern, S., Koumoutsakos, P.: Simulations of optimized anguilliform swimming. Journal of Exp. Biology **209**(24), 4841–4857 (2006)
17. Kober, J., Bagnell, J.A., Peters, J.: Reinforcement learning in robotics: A survey. The International Journal of Robotics Research **32**(11), 1238–1274 (2013)
18. Koziel, S., Yang, X.S. (eds.): Computational optimization, methods and algorithms. Springer Science and Business Media (2011)
19. Kuo, P.D., Grierson, D.: Genetic algorithm optimization of escape and normal swimming gaits for a hydrodynamical model of carangiform locomotion. In: Proc. Genetic and Evolutionary Computing Conf., pp. 170–177. Chicago, IL (2003)
20. Laskari, E.C., Parsopoulos, K.E., Vrahatis, M.N.: Particle swarm optimization for minimax problems. In: Proc. Congress on Evolutionary Computation (CEC), pp. 1576–1581. Honolulu, HI (2002). DOI 10.1109/CEC.2002.1004477
21. Lighthill, M.J.: Large-amplitude elongated-body theory of fish locomotion. Proceedings of the Royal Society of London. Series B. Biological Sciences **179**(1055), 125–138 (1971)
22. Liljebäck, P., Pettersen, K.Y., Stavdahl, Ø., Gravdahl, J.T.: Snake Robots: Modelling, Mechatronics, and Control. Springer-Verlag, Advances in Industrial Control (2013)
23. Liljeback, P., Stavdahl, O., Pettersen, K.Y., Gravdahl, J.T.: Mamba - a waterproof snake robot with tactile sensing. In: Proc. International Conference on Intelligent Robots and Systems (IROS), pp. 294–301. Chicago, IL (2014). DOI 10.1109/IROS.2014.6942575
24. McIsaac, K., Ostrowski, J.: Motion planning for anguilliform locomotion. IEEE Transactions on Robotics and Automation **19**(4), 637–625 (2003)
25. Pareto, V.: Manual of political economy. A. M. Kelley (1906)
26. Parsopoulos, K., Vrahatis, M.: Recent approaches to global optimization problems through particle swarm optimization. Natural Computing **1**(2-3), 235–306 (2002)
27. Taylor, G.: Analysis of the swimming of long and narrow animals. Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences **214**(1117), 158–183 (1952)
28. Tesch, M., Schneider, J., Choset, H.: Expensive multiobjective optimization for robotics. In: Proc. IEEE International Conference on Robotics and Automation (ICRA), pp. 973–980. Karlsruhe, Germany (2013)

29. Wetter, M.: GenOpt, Generic Optimization Program, 3.1.0 edn. Simulation Research Group, Building Technologies Department, Lawrence Berkeley National Laboratory (Berkeley, CA 94720, 2011)
30. Wiens, A., Nahon, M.: Optimally efficient swimming in hyper-redundant mechanisms: control, design, and energy recovery. Bioinspir. Biomim. **7**(4) (2012)
31. Zadeh, L.: Optimality and non-scalar-valued performance criteria. IEEE Transactions on Automatic Control **8**(1), 59–60 (1963)