Journal of Intelligent & Robotic Systems (2020) 97:657–672
https://doi.org/10.1007/s10846-019-01008-y

Adaptive Neural Network Control of Underwater Robotic Manipulators Tuned by a Genetic Algorithm

Tony Salloom[1,2] · Xinbo Yu[1,2] · Wei He[1,2] · Okyay Kaynak[1,2]

Received: 18 September 2018 / Accepted: 28 February 2019 / Published online: 20 March 2019

Abstract

This paper describes a novel approach for the control of underwater robots that can handle uncertainties and disturbance problems, which are commonly met in underwater environments. The considered system is an underwater manipulator with n-degrees of freedom. The approximation capability of an adaptive neural network is exploited to estimate uncertainties in system dynamics. Drag and lift forces are considered as an external disturbance, and a disturbance observer approach which has been proved to be effective with on-land robotic systems, is applied to compensate for it. The objective of the controller designed is to track a desired trajectory. To find the optimal gain parameters of this controller, a classical Genetic Algorithm is employed. Extensive simulation studies carried out on a two degrees of freedom manipulator indicate the efficacy of the proposed approach, proving that the disturbance observer originally developed for on-land systems can also be used effectively for underwater robotic systems. Finally, the performance of the proposed controller, tuned by the Genetic Algorithm is compared with that of a controller, tuned manually. The results show that the reliance on a well-known classic Genetic Algorithm for the tuning of the controller parameters not only saves time, but also provides better values of the parameters.

Keywords Control design · Underwater robots · Adaptive neural network · Drag force · Genetic algorithm · Disturbance observer

1 Introduction

Recently, underwater world has been in the forefront for many researchers. Tasks such as exploration of underwater resources, maintaining underwater equipment, investigation of sunken ships and submarines, and many other underwater manipulation tasks have become important research issues. Since the underwater world can be perilous for human beings, underwater robotic manipulators are being increasingly used in a greater number of tasks, and many control strategies have been developed to control robotic manipulators [1]. However, the accomplishment of an underwater manipulating task in an accurate manner requires accurate system dynamics. In fact, dynamic modeling of underwater robotic systems is more complicated than that of on-land robotic systems [2–4]. Additionally, there exist a higher level of uncertainties in underwater robotic systems, due to the extra forces exerted by water, such as added mass and moment of inertia, buoyancy, drag force, etc. Most of them depend on the relative velocity and acceleration between water and manipulator.

In practice, the dynamic models of robotic manipulators, and of all other physical systems, will always suffer from some uncertainties. These are due to inaccurate information on physical parameters, interaction with the surrounding environment, and such effects. Therefore the dynamics of a manipulator that include moment of inertia, gravitational and centrifugal and coriolis matrices may be known only partially [5]. Uncertainties can be of two types; parametric uncertainties and non-parametric uncertainties. Both are studied widely in literature. In [6], authors design a model-free adaptive controller for linear systems considering both parametric and internal-external non-parametric uncertainties. In [7], authors design and analyse an adaptive estimator for an unknown linear

✉ Wei He
weihe@ieee.org

[1] School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing, 100083, China

[2] Institute of Artificial Intelligence, University of Science and Technology Beijing, Beijing, 100083, China

system, by depending on the noisy states and control signal. In [8], authors design and carry out an identification scheme of the model of 6-DOF upper limb of a humanoid robot. In this paper, we use a NN to approximate the parametric uncertainties while a DO is used to approximate the nonparametric uncertainties with added external disturbance.

Neural Networks (NNs) have long been used as a powerful tool in control of nonlinear systems [9–13], due to their abilities, such as the ability to learn [14–17] and the ability to approximate any continuous function with a reasonable error as in [18–22]. The ability of approximation is exploited in the field of control design to handle the unknown part in system model under different operating conditions [5, 23–25]. Authors in [24] consider a case of a full-state feedback with time-varying output constraints, and use the adaptive NN for the same purpose. Authors in [26, 27] design and control a tethered space robot in order to use it for space debris cleaning mission, they use a NN to compensate for system's uncertainties. However, all of the referenced works are meant to handle the dynamics uncertainties of on-land manipulators, while not much can be seen in literature that consider the uncertainty problems in underwater robotic systems. In the few works that are available, a variety of approaches are suggested, such as indirect adaptive control to control the underwater vehicle-manipulator [28], fuzzy control [29], and sliding mode control [30–32]. In [32], a NN is used to reduce the distance between the sliding surface and the current state. The work reported in this paper distinguishes itself from the referenced ones by handling a higher degree of uncertainties caused by underwater environment.

In addition to the problem of uncertainties, robotic systems are subject to external disturbances, which are mostly unknown and unexpected [33, 34], especially in underwater environment. The technique of disturbance observer (DO) has widely been used to compensate this kind of disturbance [24, 35]. However, there exists a number of works in the literature that considers this technique as a candidate to solve the disturbance problem in underwater environment. Authors in [36] design a sliding mode controller for underwater vehicle based on a backstepping technique. They design a DO to compensate for the uncertainties in system dynamics and the external disturbance together. However, they ignore the uncertainties that exist in the moment of inertia matrix and design their DO based on that matrix. In [37], a sliding mode controller is designed for autonomous underwater vehicles, while a DO is designed to compensate for the unknown model parameters and disturbances. In [38], an integral sliding mode controller is designed for underwater robot vehicle based on a state observer. The external disturbance and the unmeasured velocity are considered as extended state. In this paper, we show that the DO described in [24] can be used effectively for underwater disturbances approximation.

To our knowledge, there is no work reported in literature that uses a NN controller with a DO as a method to handle the uncertainties and the disturbances that underwater manipulators may be subject to.

During the literature survey process of our work, we have noted that most research in the field of control design discuss the design process without presenting a clear explanation of how the parameters of the controller designed are tuned, because they can be set manually and get reasonable performance. However, manual-tuning process may take a long time. Furthermore, the resulting performance may be acceptable but still far from the optimal. That has inspired us to use an intelligent method to tune our controller. The use of an evolutionary algorithm, especially a Genetic Algorithm (GA) is a good solution for that. GA basically is a search-based optimization method, which depends on the principle of genetics and natural selection. Due to its properties, GA is used frequently to find the optimal solution of many problems. Path planning is one of its applications[10, 39, 40]. It has been applied in [40] to find the shortest possible and collision free path while considering a safety parameter. GA has been used in [41] to tune the parameters of the proportional integral derivative (PID) controller, which is designed to control a bidirectional inductive power transfer system. It is also used with a Multi-input-Multi-output (MIMO) system in [42] to optimize the controller designed. GA can widely be seen in underwater robotic systems too. Authors in [43] use GA as a task-based design method, where they develop a suitable propulsion configuration for autonomous underwater vehicles (AUVs) and tune the control parameters of the propulsion system. Some research has been done to prove the superiority of GA as in [44].

The robot considered in this work (Fig. 1) is an n-link manipulator with uncertain system dynamics. For the accomplishment of an underwater manipulating task, it is suggested that an adaptive NN-based controller is used, which is an adapted version of the controller designed in [20]. A DO compensates for drag and lift forces as well as external friction. Although, it is possible to employ a NN to approximate them, we chose to use a DO for two reasons:

- To simplify the NN used for approximation, because simpler functions can be approximated satisfactorily using simpler NN.
- The designed DO can also compensate for the external unknown disturbances, which are very likely to occur during underwater manipulating tasks.

In order to guarantee near-optimal performance, the controller parameters are tuned by a GA. The efficiency of the designed controller and the justification for the use

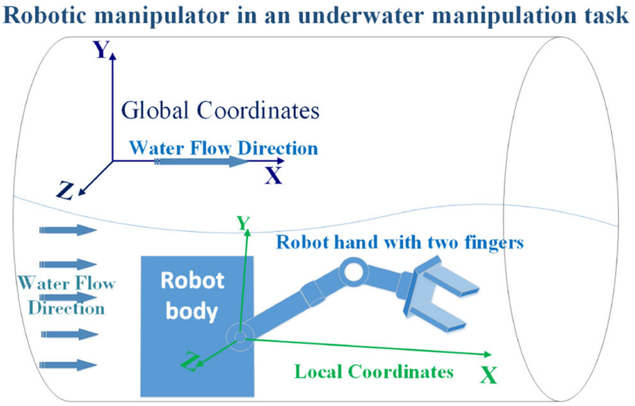


Fig. 1 The n-DOF manipulator considered

of a GA for tuning are demonstrated by MATLAB based simulation studies.

The contributions of this work include:

- An efficient controller is proposed for an n-DOF underwater robotic manipulator. It is based on an adaptive NN and a DO.
- It is proved that the DO used for on-land manipulators also works with underwater robots to compensate for the external forces, such as drag, lift and external friction.
- The parameters of the designed controller are tuned using GA. The advantages of GA-based tuning over manual tuning are shown by simulation studies.

The rest of this paper is organized like this: the problem formulation and system dynamics are explained in Section 2. In Section 3, the controller is designed, and the system stability is proved. The designed GA steps are illustrated in Section 4. Simulation environment and results are presented in Section 5. Finally, an assessment of the work is given in Section 6.

2 Problem Formulation

The control problem of an uncertain n-degrees of freedom robotic manipulator, operating in a water stream is discussed in this paper. The system considered is shown in Fig. 1. It can be represented as a MIMO nonlinear system.

The dynamics of underwater robot manipulators are one of the most complicated dynamics. Once such manipulators are underwater, they face numerous forces that significantly change the dynamics of the system from what they are on land [45]. Below is a brief explanation of those forces:

1. *Buoyancy and earth gravity*: The effect of these on the robotic manipulator are opposite to each other. Buoyancy equals to the water mass displaced by the rigid body. It can be calculated by the following equation [46]:

$$\beta = \rho g V = bg \tag{1}$$

where $b = \rho V$, ρ is the water density, V is the volume of displaced water, which is equal to the volume of immersed rigid body, g is the earth gravity, its value changes with the depth underwater.

2. *Added mass and moment of inertia*: These forces can reach significant levels when an immersed object has to shift (or deflect) some water as it moves through it. It is hard to calculate it. These forces depend on the shape of the moving object. For simplicity, they are considered as a volume of water moving with the body. This assumption adds to other model-system mismatches.
3. *Internal friction force*: This is due to the friction in the joints and can be neglected when compared to other resistant forces.
4. *Drag force and external friction*: These are due to the difference between the speed of the water flow and the robot speed when the manipulator moves in water. The effect of the external friction force caused by water is included in the drag force equation through its coefficient. In both deep or shallow water, it is found that it can be expressed like in Morison's equation [47]:

$$f_{Drag} = \frac{1}{2}\rho A C_d \|v\|^2 \tag{2}$$

where ρ is water density, A is the cross sectional area projected normally to the direction of motion, v is the relative velocity of robot v^r to water speed v^w,

$$v = v^r - v^w \tag{3}$$

C_d in Eq. 2 represents the drag coefficient. It reflects the impact of shape, texture, viscosity (which results in viscous drag or skin friction), compressibility and lift (which causes induced drag). Drag coefficient is very hard to determine, especially for not cylindrical shape. It may be obtained experimentally [48, 49] or determined as a function of Reynold number [50].

The dynamics of underwater robotic system of the n-degrees of freedom manipulator is modeled as in the following equation:

$$M(\phi)\ddot{\phi} + C(\phi, \dot{\phi})\dot{\phi} + G(\phi) = \tau - f_{des}(t) \tag{4}$$

where:

- ϕ is $(n \times 1)$ vector that represents the angular position of n joints.
- $\dot{\phi}$, $\ddot{\phi}$ are $(n \times 1)$ vectors of velocity and acceleration, respectively.
- M is an $(n \times n)$ matrix. It represents the inertia matrix including the added mass and the added moment of inertia. It satisfies the symmetric positive properties.

- C is an $(n \times n)$ matrix. It represents the coriolis and centrifugal matrix of n-DOF manipulator, including the added coriolis and centrifugal forces.
- G is a vector of n elements. It represents the gravity forces and the buoyancy effects.
- f_{des} represents the unknown external disturbance which basically cosed by the hydraulic drag and lift forces in the underwater environment. It also includes the friction due to the water flow.
- τ is the vector of applied joint torques, which actually are the control inputs. It contains n elements.

3 Control Design

The uncertainties in the considered system are parametric uncertainties which exist in any realistic system, due to the manufacturing process and associated with the system parameters such as mass density, geometric parameters, etc. And nonparametric uncertainties due to the lack of knowledge regarding the system. In this work, we use an adaptive NN to approximate the parametric uncertainties. Drag and lift forces are considered as an unknown disturbance from the external environment, and estimated by a DO. The DO is used to estimate the non-parametric uncertainties too.

Figure 2 shows the control strategy.

The system can be described in terms of the state variables as follows:

$$\dot{x}_1 = x_2 \tag{5}$$

$$\dot{x_2} = M^{-1}(x_1)\,[\tau - C(x_1, x_2)x_2 - G(x_1) - f_{dis}] \tag{6}$$

where $x_1 = \phi$ and $x_2 = \dot{\phi}$ are the state variables and f_{dis} is the external disturbance. In this paper, the state variables x_1 and x_2 are assumed to be known in full, (i.e. the case of full state feedback). The required task is tracking the desired trajectory $x_d = \phi_d = [\phi_{d1}, \phi_{d2}, \ldots, \phi_{dn}]^T$. The tracking error will be:

$$e_1(t) = x_1(t) - x_d(t) \tag{7}$$

We design another control variable $\alpha(t)$ related to the tracking error and the velocity. It is designed as in the following equation:

$$\alpha(t) = -K_1 e_1(t) + \dot{x}_d(t) \tag{8}$$

where K_1 is the gain matrix, which is asymmetric positive matrix. We define the second error as follows:

$$e_2(t) = x_2(t) - \alpha(t) = \dot{x}_1(t) - \alpha(t) \tag{9}$$

and the time derivative of $e_2(t)$ is

$$\dot{e}_2 = M^{-1}(x_1)\,[\tau - C(x_1, x_2)x_2 - G(x_1) - f_{dis}] - \dot{\alpha} \tag{10}$$

The unknown parts of the system dynamics are $M(x_1)$, $C(x_1, x_2)$ and $G(x_1)$. Let us assume that M and C matrices consist of two parts, a known part $M_c(x_1)$ and $C_c(x_1, x_2)$, respectively, and an unknown part $\Delta M(x_1)$ and $\Delta C(x_1, x_2)$, respectively. Then $M(x_1) = M_c(x_1) + \Delta M(x_1)$ and $M_c(x_1)$ should be skew symmetric and positive definite matrix.

$C(x_1, x_2) = C_c(x_1, x_2) + \Delta C(x_1, x_2)$, and $C_c(x_1, x_2)$ is designed to satisfy that $\dot{M}_c(x_1) - 2C_c(x_1, x_2)$ is a skew symmetric matrix. In the rest of this paper, we will use the following denotation: M_c represents $M_c(x_1)$, ΔM represents $\Delta M(x_1)$, C_c represents $C_c(x_1, x_2)$, and ΔC represents $\Delta C(x_1, x_2)$. Thus, Eq. 10 can be written as:

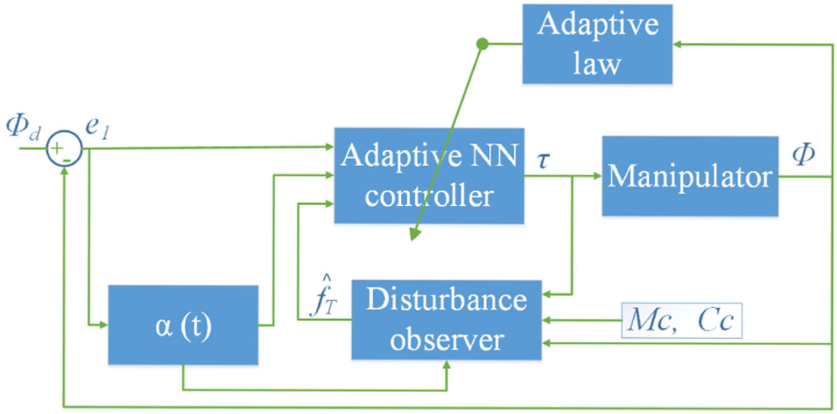
$$\begin{aligned}\dot{e}_2 = M_c^{-1}(x_1)\,[&\tau - M(x_1)\dot{\alpha} - \Delta M(x_1)\dot{e}_2 - G(x_1)\\ &- f_{dis} - C_c(x_1, x_2)x_2 - \Delta C x_2]\end{aligned} \tag{11}$$

Then the NN $W^T\Theta(h)$ is used to approximate the uncertainties as follows:

$$W^T\Theta(h) = M(x_1)\dot{\alpha} + \Delta M \dot{e}_2 + \Delta C(x_1, x_2)x_2 + G(x_1) - \delta \tag{12}$$

where δ is the approximation error, $h = [\phi^T, \dot{\phi}^T, \alpha^T, \dot{\alpha}^T]$ is the input to the NN, and $\Theta(h)$ is the radial basis function

Fig. 2 The control strategy



which is used as an activation function of the NN. It is shown in the following equation.

$$\Theta_i(h) = \exp\left[\frac{-(h-o_i)^T(h-o_i)}{\mu_i^2}\right] \tag{13}$$

where o_i is the center of the function, and μ is the variance. Let the NN $\hat{W}^T\Theta(h)$ be the estimation of $W^T\Theta(h)$. The adaptive law of the NN proposed in Eq. 12 is designed as:

$$\dot{\hat{W}} = -\psi_i\left[\Theta_i(h)e_{2,i} + \gamma_i|e_{2,i}|\hat{W}_i\right],\ i = 1, 2, ..., N \tag{14}$$

where N represents the number of nodes in the hidden layer of the NN, ψ_i is a positive gain, γ_i is a small positive constant.

Proposition 1 [24, 51] *Considering the adaptive law* (14), $\dot{\hat{W}}_i$ *is bounded. In other words, there is a positive constant* $\epsilon_i > 0$, *where* $\|\Theta_i(h)\| \le \epsilon_i$, *then* $\|\hat{W}\| \le \frac{\epsilon_i}{\gamma_i}$, *as proved in* [52].

Substitution of Eqs. 12 in 11 results in the following:

$$\dot{e}_2 = M_c^{-1}\left[\tau - C_c x_2 - W^T\Theta(h) - \delta - f_{dis}\right] \tag{15}$$

Let us consider this error δ as an extra source of disturbance and combine it together with the external source, the total disturbance is:

$$f_T = f_{dis} + \delta \tag{16}$$

Lemma 1 [24]*: Let* $F(x)$ *be a continually differentiable function over* $[0, \infty)$*. If* $F(x)$ *is bounded then* $\dot{F}(x)$ *is bounded over* $[0, \infty)$*, which means there is a positive constant* $\chi > 0$ *where* $|F(x)| \le \chi, \forall x \in [0, \infty)$ *then* $|\dot{F}(x)|$ *is bounded* $\forall x \in [0, \infty)$.

The drag and external friction forces are bounded, because they are caused by the difference between water speed and the robot speed and both velocities can not be infinite. In addition, these forces are function of The designed NN approximation error δ is also bounded based on Proposition 1 and Eq. 12. Thus, the disturbance is bounded. In addition, the drag and external friction forces are a function of the square of the difference between the velocities of the manipulator and the water flow, as in Eq. 2. Since both velocities are continuous and differentiable, then Drag force is continuous and differentiable, consequently, the disturbance is continuous and differentiable. According to Lemma 1, there is a positive constant ξ where $\|\dot{f}_T\| \le \xi$. In order to compensate for disturbance, the DO, which is proposed in [24], is applied. Let us define the disturbance estimation error as follows:

$$e_3 = f_T + \Phi(e_2) \tag{17}$$

where $\Phi(e_2)$ is a linear function of e_2 which can be designed easily to ensure that

$$\frac{\partial\Phi(e_2)}{\partial e_2} = B \tag{18}$$

where B is a positive constant. The estimated disturbance is:

$$\hat{f}_T = \hat{e}_3 - \Phi(e_2) \tag{19}$$

where $\hat{e}_3$ is the estimation of e_3. Since the velocity is available as a feedback signal, then e_2 is known. Hence, the disturbance can be estimated if there is a convenient way to establish $\hat{e}_3$. Let us design $\hat{e}_3$ as follows:

$$\dot{\hat{e}}_3 = BM_c^{-1}(\tau - C_c x_2 - \hat{f}_T) \tag{20}$$

It is clear that:

$$\dot{\tilde{f}}_T = \dot{\tilde{e}}_3 = BM_c^{-1}\left(-\tilde{f}_T + W\Theta(h)\right) - \dot{f}_T \tag{21}$$

The input torque is:

$$\tau = -e_1 - K_2e_2 + C_c\alpha + \hat{W}^T\Theta(h) + \hat{f}_T \tag{22}$$

Lyapunov theory has proved itself as an effective method to prove the stability of nonlinear systems, it has been used widely for that purpose [53–56]. The stability of this system also can be proved by using the Lyapunov second theory. Let us consider the following Lyapunov function:

$$V = \frac{1}{2}e_1^Te_1 + \frac{1}{2}e_2^TM_ce_2 + \frac{1}{2}\tilde{f}_T^T\tilde{f}_T + \frac{1}{2}\sum_{i=1}^{N}\tilde{W}_i^T\psi_i^{-1}\tilde{W}_i \tag{23}$$

where $\tilde{W}_i = \hat{W}_i - W_i$ is the error in the adapted NN weights, $\tilde{f}_T = \hat{f}_T - f_T$ is the disturbance estimation error. For the underwater robotic manipulator system defined by the dynamics (4), with uncertain system dynamics under bounded initial conditions and the control law in Eq. 14, the error signals e_1, e_2, $\tilde{f}_T$ and $\tilde{W}$ are bounded over compact sets Ω_{e_1}, Ω_{e_2}, Ω_{f_T} and Ω_W, respectively. The compact sets are defined as follows:

$$\Omega_{e_1} := \left\{e_1 \in \mathbb{R}^n | \|e_1\| \le \sqrt{S}\right\} \tag{24}$$

$$\Omega_{e_2} := \left\{e_2 \in \mathbb{R}^n | \|e_2\| \le \sqrt{\frac{S}{\lambda_{\min}(M)}}\right\} \tag{25}$$

$$\Omega_{f_T} := \left\{\tilde{f}_T \in \mathbb{R}^n | \|\tilde{f}_T\| \le \sqrt{S}\right\} \tag{26}$$

$$\Omega_W := \left\{\tilde{W} \in \mathbb{R}^{N\times n} | \|\tilde{W}\| \le \sqrt{\frac{S}{\lambda_{\min}(\psi^{-1})}}\right\} \tag{27}$$

In above $S = 2\left(V(0) - \frac{b}{a}\right)$, where a and b are two positive numbers defined by Eq. 54 in Appendix. $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ represent the maximum and minimum eigenvalues of matrix A, respectively.

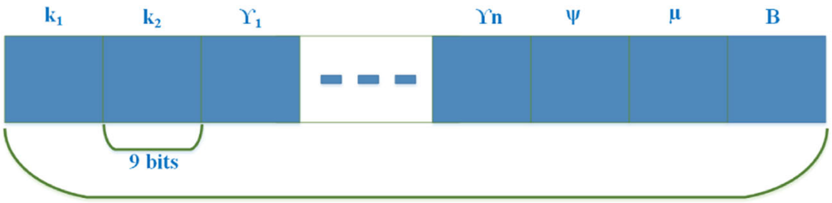


Fig. 3 Representation of the parameters to be optimized in the chromosomes of GA

Thus, the handled system described by the dynamics in Eq. 4 with full-state feedback under the designed controller is stable for every bounded initial conditions. the proof clarified in Appendix.

4 Tuning the Parameters of the Controller by a Genetic Algorithm

An optimal controller is extremely desirable [57] in control systems. In this work, GA is used to optimize the proposed controller. The implementation of the controller proposed in Eq. 22, requires the tuning of the adaptive NN parameters proposed in Eq. 14, in addition to the DO parameter B and the control variable α. In other words, the variables K_1, K_2, ψ, γ_i, μ and B need to be tuned. In order to guarantee a high performance with minimum tracking error, a method should be used to arrive at the optimal parameters as closely as possible. The use of a GA is known to be effective in this domain. The details of the GA designed are described in what follows, assuming that the basics are already known by the reader.

Solution Representation The solution is a GA chromosome that contains the scalers Gk_1, Gk_2, $G\psi$, $G\gamma_i$, $G\mu$ and GB. They are used to set the variables mentioned above. Every scaler is represented by a string of binary bits as shown in Fig. 3. The length of this string should be chosen according to the solution space. In our case, we choose 9 bits for every scaler. The genotype is created by encoding every variable by the use of gray coding method. Gray coding is useful to avoid the undesired results of mutation and crossover process.

After finding optimal values of those scalers, they are used to set the parameters of the related control law as follows: $K_1 = Gk_1 I_{n\times n}$, $K_2 = Gk_2 I_{n\times n}$, $\psi = G\psi I_{N\times N}$, $\gamma_i = G\gamma_i$, $\mu = G\mu$ and $B = GB$.

The steps of the tuning procedure are shown in the GA flow chart in Fig. 4.

Initial Population The initial population is created randomly as integer numbers between [0, 512] to guarantee the required positivity, then in-coded using gray code. However, the value of γ_i should be less than one, so the randomly created numbers are divided by 513 to get suitable values.

Fitness Function The goal of this algorithm is to minimize the tracking error of all links. Since the tracking error of ϕ_n must be small as possible (because link n affects the end-effector position directly), we consider it twice in the fitness function. Thus the fitness function of every possible solution j is set to be like:

$$fit_j = 2e_1^n + \sum_{i=1}^{i=n-1} (e_1^i) + \sum_{i=1}^{i=n} \tilde{f}_T^{i} \tag{28}$$

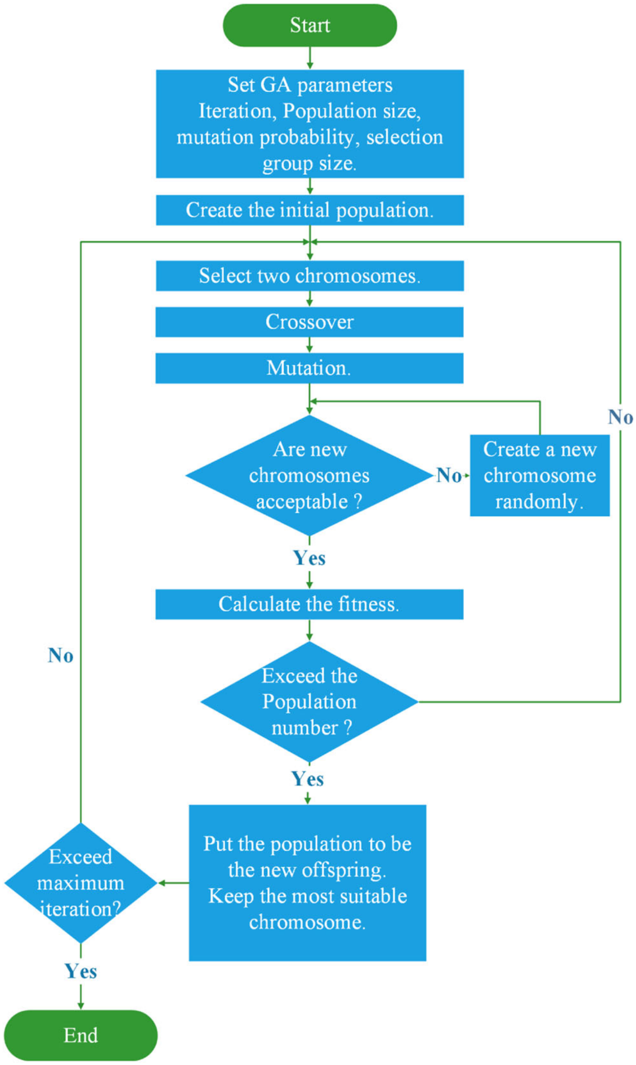

Start
Set GA parameters Iteration, Population size, mutation probability, selection group size.
Create the initial population.
Select two chromosomes.
Crossover
Mutation.
Are new chromosomes acceptable ?
No
Create a new chromosome randomly.
Yes
Calculate the fitness.
Exceed the Population number ?
No
Yes
Put the population to be the new offspring. Keep the most suitable chromosome.
Exceed maximum iteration?
No
Yes
End

Fig. 4 GA flow chart

where e_1^i is the largest tracking error of link i calculated during the last 10 seconds of the simulation, $\tilde{f}_T^i$ is the approximation errors of the DO for link i. Thus the solution is: $Sol = \min(fit_j), \quad j = 1, 2, \ldots, population\_size$.

Selection of Parents The next phase of GA is to select the parents of the next generation members. The tournament selection method is applied here. Two groups of chromosomes are created. The members of every group are chosen from the current population at random. Then, the chromosome with the best fitness value in the group is chosen to be a parent (elitism). Such a method encourages the highly fit individuals to be chosen. The new offspring is yielded by applying crossover and mutation operations on the selected parents.

Crossover Process The crossover process involves transferring some parts between the two parents to create two new chromosomes. We choose to use the multi-points crossover method. Three cros-sover points are chosen such that every chromosome are divided into 4 parts, and the crossover process swaps the first and the third quarters between the parents to result in two new chromosomes. Every new chromosome is tested against the control design conditions to guarantee that we do not get a singularity of matrix M during the tracking mission. If it violates a condition, it is rejected, and a new chromosome is created randomly. This creation and testing process is repeated until the newly created chromosome satisfies all conditions. The selection and the crossover processes are repeated over the whole population.

Mutation For every chromosome in the new population, a random number between [0,1] is created. If this number is smaller than the chosen mutation probability, then a random bit is flipped. The resulting chromosome is also tested against the control design conditions.

Termination method The steps described above are repeated until a predefined iteration number is reached. The approach known as elitism is used, i.e. the best chromosome of a population is kept until a chromosome with a better fitness value is created. Finally the best solution is used in the designed controller. In practice, there is more than one way to terminate the process; the definition of a satisfactory error threshold is one of them, the algorithm stops when the desired error threshold is reached. However, in our particular case of application, no specific error levels can be defined.

5 Simulation Studies

We consider the 2-DOF manipulator moving in the xy plane. The manipulator is built of two links jointed by two rotary joints. Both joints rotate around z axis. The position of the wrist is derived by two rotations. Let m_i and l_i be the mass and the length of link i. The center of mass is located at center of the link at l_{ci} distance from the previous joint. Both links are cylindrical with radios R_i and length l_i. The added mass m_{ai} of link i can be calculated based on the following formula:

$$m_{ai} = \rho\pi R_i^2 l_i \tag{29}$$

The total mass of link i can be calculated as in [46]: $m_i = m_{ri} + m_{ai}$. Buoyancy can be calculated by Eq. 1.
We identify ϕ as:

$$\phi = \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} \Rightarrow \dot{\phi} = \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{bmatrix}$$

Using Lagrange equation, the dynamics of the 2-DOF manipulator can be written as in Eq. 4, where:

$$M(\phi) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \tag{30}$$

$$C(\phi, \dot{\phi}) = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & 0 \end{bmatrix} \tag{31}$$

$$G(\phi) = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix} \tag{32}$$

where:

$$\begin{aligned}
M_{11} &= m_1 l_{c1}^2 + m_2(l_1^2 + l_{c2}^2 + 2 l_1 l_{c2} \cos\phi_2) + 0.25 m_1 l_1^2 + 0.25 m_2 l_2^2, \\
M_{12} &= M_{21} = m_2(l_{c2}^2 + l_1 l_{c2} \cos\phi_2) + 0.25 m_2 l_2^2, \\
M_{22} &= m_2 l_{c2}^2 + 0.25 m_2 l_2^2, \\
C_{11} &= -m_2 l_1 l_{c2} \dot{\phi}_2 \sin\phi_2, \\
C_{12} &= -m_2 l_1 l_{c2} (\dot{\phi}_1 + \dot{\phi}_2) \sin\phi_2, \\
C_{21} &= m_2 l_1 l_{c2} \dot{\phi}_1 \sin\phi_2, \\
G_1 &= g((m_1 - b_1) l_{c2} + (m_2 - b_2) l_1) \cos\phi_1 + (m_2 - b_2) l_{c2} g \\
\cos &\quad (\phi_1 + \phi_2), \\
G_2 &= (m_1 - b_1) l_{c2} g \cos(\phi_1 + \phi_2).
\end{aligned}$$

We design $M_c = (\sin(t) + 2) I_{2\times 2}$ and $C_c = (0.5\cos(t) + 1) I_{2\times 2}$. Such a design guarantees that M_c is positive definite matrix and $\dot{M}_c - 2C_c$ is a skew symmetric matrix. For simplicity in simulation, we assume that water flows along the x axis in the local coordinate system. The water current is assumed to be not rotational but lateral and flows in one direction. In order to make this work more useful, the water

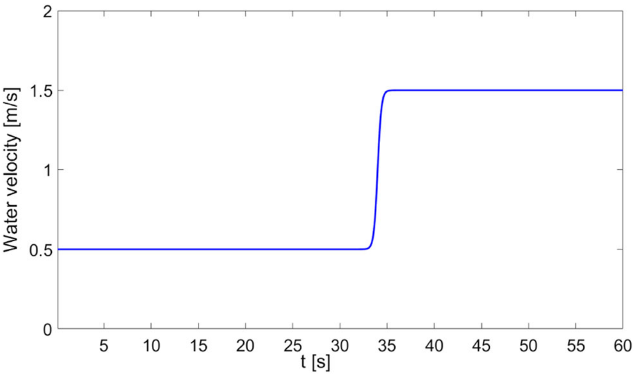


Fig. 5 Water velocity

velocity is assumed to undergo a step change during the simulation time, as shown in Fig. 5.

It is assumed to be flowing at a velocity of $v_w = 0.5$ (m/s) when the simulation starts. In order to simulate nature, we assume that around the 34th second, the robot faces a sudden flow of water with a high speed; around 1.5 (m/s). The water flow velocity follows the following model: $v_w^x = 0.5 + \frac{1}{1+e^{5(-t+34)}}$.

The kinematics and the Jacobian matrices of link 1 and link 2 can be found as follows:

$$P_1 = \begin{bmatrix} P_{11} \\ P_{12} \end{bmatrix} = \begin{bmatrix} l_{c1} \cos \phi_1 \\ l_{c1} \sin \phi_1 \end{bmatrix} \tag{33}$$

$$J_1 = \begin{bmatrix} J_{111} & 0 \\ J_{121} & 0 \end{bmatrix} = \begin{bmatrix} -l_{c1} \sin \phi_1 & 0 \\ l_{c1} \cos \phi_1 & 0 \end{bmatrix} \tag{34}$$

$$P_2 = \begin{bmatrix} P_{21} \\ P_{22} \end{bmatrix} = \begin{bmatrix} l_1 \cos \phi_1 + l_{c2} \cos(\phi_1 + \phi_2) \\ l_1 \sin \phi_1 + l_{c2} \sin(\phi_1 + \phi_2) \end{bmatrix} \tag{35}$$

$$J_2 = \begin{bmatrix} J_{211} & J_{212} \\ J_{221} & J_{222} \end{bmatrix}$$

$$= \begin{bmatrix} -l_{c1} \sin \phi_1 + l_{c2} \sin(\phi_1 + \phi_2) & -l_{c2} \sin(\phi_1 + \phi_2) \\ l_{c1} \cos \phi_1 + l_{c2} \cos(\phi_1 + \phi_2) & l_{c2} \cos(\phi_1 + \phi_2) \end{bmatrix} \tag{36}$$

The linear velocity of link 1 and 2 can be computed as follows:

$$v_{r1} = \begin{bmatrix} v_{r1}^x \\ v_{r1}^y \end{bmatrix} = \begin{bmatrix} J_{111} \dot{\phi}_1 \\ J_{121} \dot{\phi}_1 \end{bmatrix} \tag{37}$$

$$v_{r2} = \begin{bmatrix} v_{r2}^x \\ v_{r2}^y \end{bmatrix} = \begin{bmatrix} J_{211} \dot{\phi}_1 + J_{212} \dot{\phi}_2 \\ J_{221} \dot{\phi}_1 + J_{222} \dot{\phi}_2 \end{bmatrix} \tag{38}$$

The drag torque can be calculated as follows:

$$D(\phi, \dot{\phi}) = \frac{1}{2} \rho C_d \begin{bmatrix} d_{11} + d_{12} \\ l_1 R_1 (d_{21} + d_{22}) \end{bmatrix} \tag{39}$$

where:
$d_{11} = l_1 R_1 (J_{111} (v_{r1}^x - v_w^x)^2 + J_{121} (v_{r1}^y - v_w^y)^2)$,
$d_{12} = l_2 R_2 (J_{211} v_{r2}^x) + J_{221} (v_{r2}^y - v_w^y)^2$,
$d_{21} = J_{212} (v_{r2}^x - v_w^x)^2$,
$d_{22} = J_{222} (v_{r2}^y - v_w^y)^2)$.

Table 1 Parameters of the robotic manipulator [20]

Parameter	Explanation	Value
m_1	Mass of link 1	2 kg
m_2	Mass of link 2	0.85 kg
l_1	Length of link 1	0.35 m
l_2	Length of link 2	0.31 m
R_1	Radius of link 1	0.1 m
R_2	Radius of link 2	0.1 m
m_{a1}	Added Mass of link 1	$\rho \pi R_1^2 l_1$
m_{a2}	Added Mass of link 2	$\rho \pi R_2^2 l_2$
V_1	Volume of link 1	$\pi R_1^2 l_1$
V_2	Volume of link 2	$\pi R_2^2 l_2$

The drag coefficient is set at $C_d = 0.82$. The parameters of the manipulator are considered as in [20], and listed in Table 1.

The desired trajectory is chosen to be a rotational motion as $\phi_{d1} = 3 \sin(0.5t)$, $\phi_{d2} = 3 \cos(0.5t)$. The number of nodes is chosen experimentally as $N = 128$. We have considered the number of nodes used in [20, 24], where a NN is used for the same purpose, we have tried the same numbers in those references, going for the smallest N because, the bigger N is the more complicated is the structure of the NN, which may generate input delay due to heavy computation, degrading the control. On the other hand, a smaller N causes a higher approximation error and, consequently, a higher tracking error.

The initial weights are set to $\hat{W}_{1,i} = \hat{W}_{2,i} = 0, (i = 1, 2, \ldots, 128)$ in the adaptive law. The simulation is run for 60 seconds.

The particular individualities of the GA are as follows:

- Every chromosome contains seven scalers as shown in Fig. 3.
- Every scaler is represented by 9 bits.

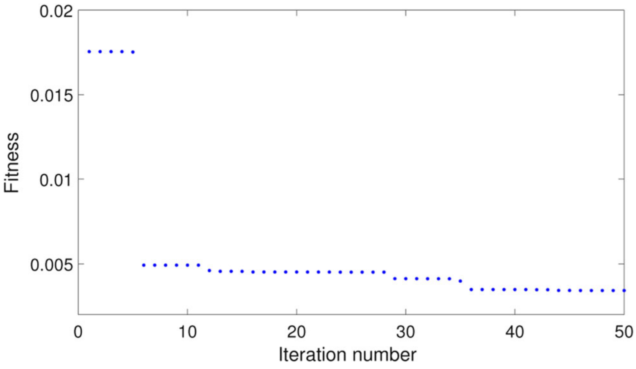


Fig. 6 Convergence of the fitness value to the optimal one

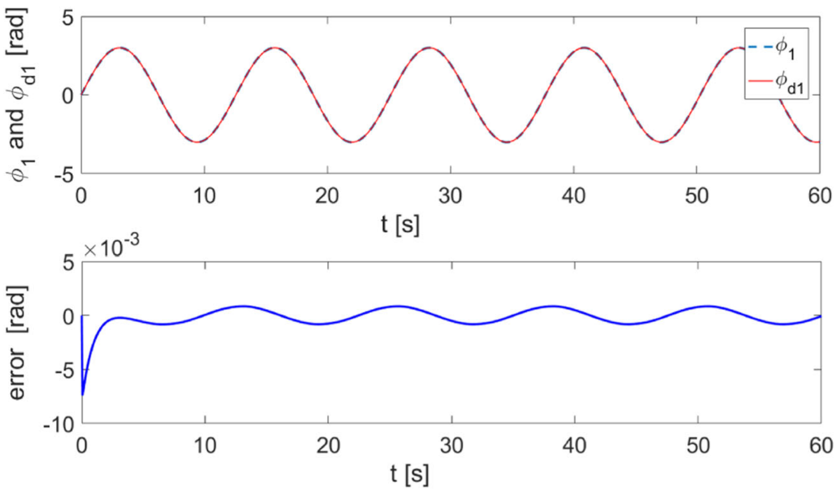


Fig. 7 Actual and desired trajectories and the tracking error of link 1

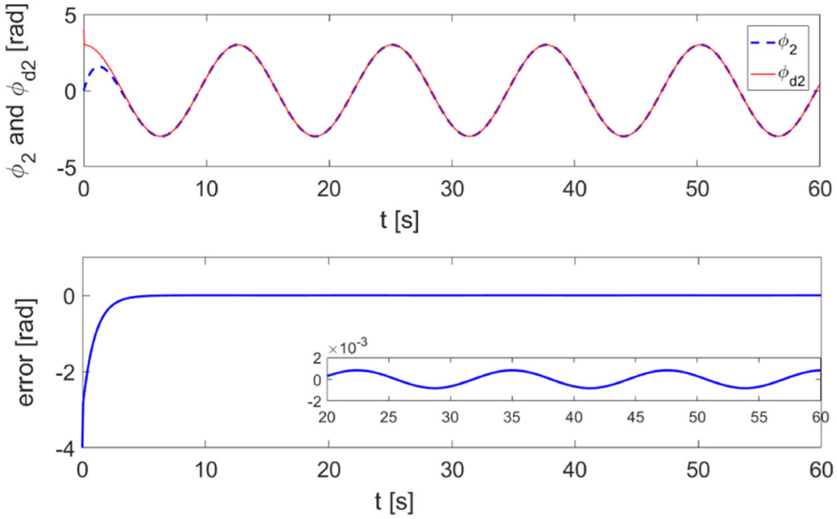


Fig. 8 Actual and desired trajectories and the tracking error of link 2

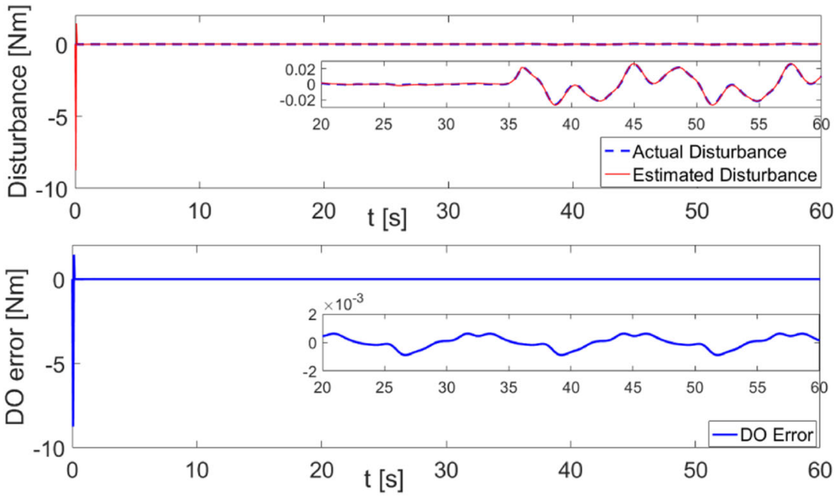


Fig. 9 External disturbance and the DO approximation error of link 1

Fig. 10 External disturbance and the DO approximation error of link 2

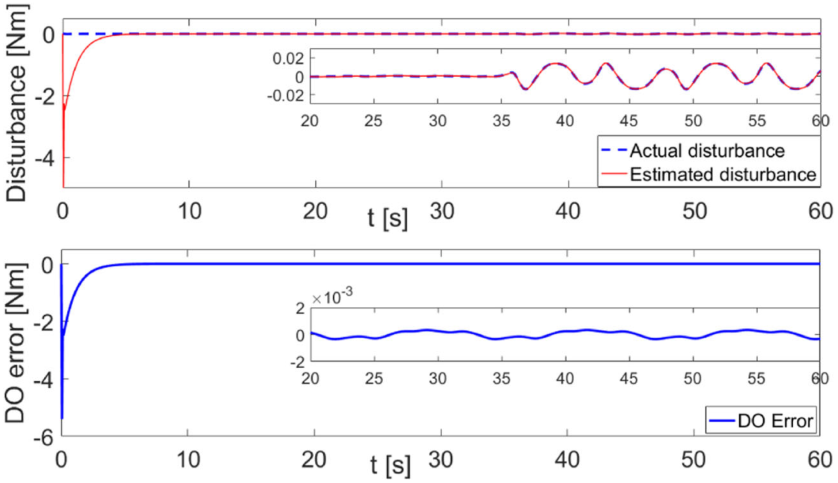


Fig. 11 Control inputs of link 1 and 2

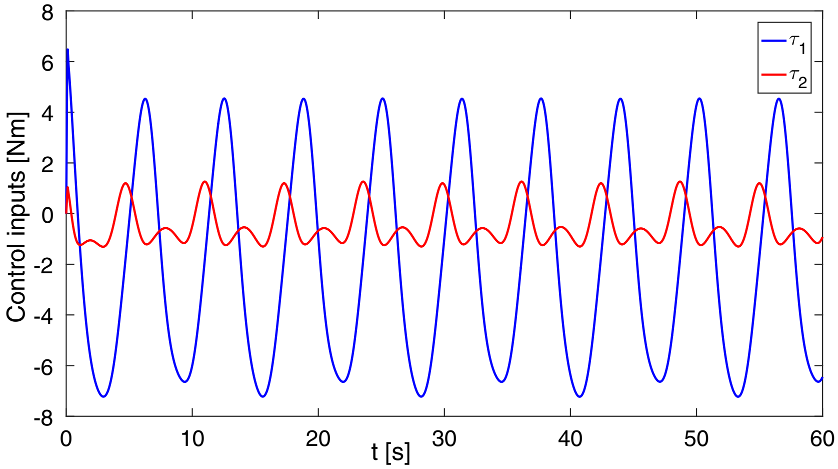
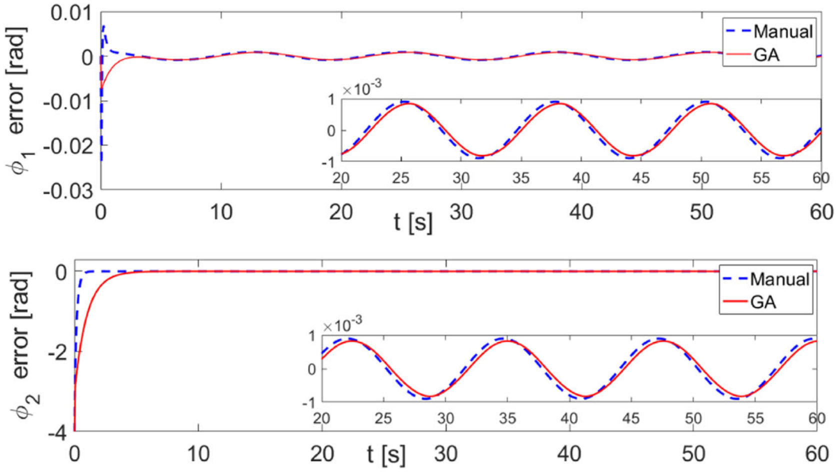


Fig. 12 Tracking errors when the parameters of the controller are tuned by GA and manually



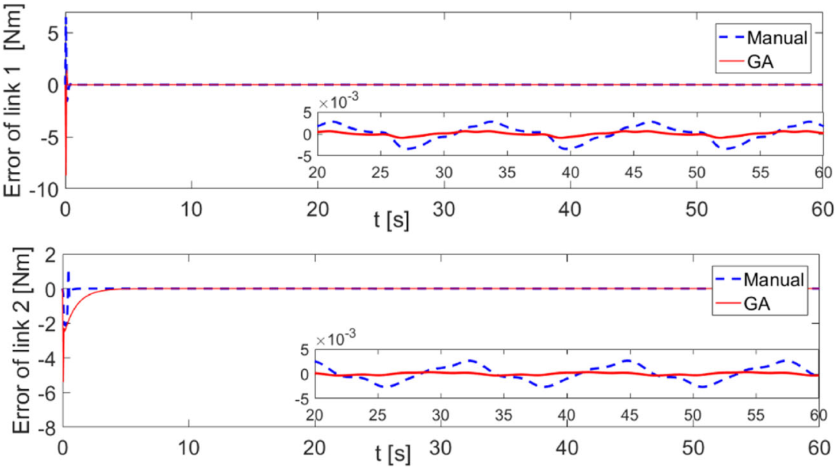


Fig. 13 Disturbance estimation errors when the parameters of the controller are tuned by GA and manually

- The population size set as: $population\_size = 16$.
- Number of iterations: $iteration = 50\, times$.
- Mutation probability set as: $mutation = 0.25$.

The GA parameters values are set experimentally. However, iteration number and population size should be chosen as small as possible in order to save computing time. The selection phase is accomplished using the tournament selection method, where a group of 3 chromosomes are chosen from the current population. The fittest chromosome is the first parent. The same process is done to choose the second parent. To find the fitness value of every chromosome, a simulation using MATLAB is run for 60 seconds. The highest error value during the last 10 seconds of the simulation is used to calculate the fitness value of every chromosome.

$$fit_j = \max_{t\in[50,60]}\left(e_1^1(t)\right) + 2\max_{t\in[50,60]}\left(e_1^2(t)\right) + \max_{t\in[50,60]}\left(\tilde{f}_T^1(t)\right) + \max_{t\in[50,60]}\left(\tilde{f}_T^2(t)\right) \tag{40}$$

The optimal solution should satisfy the following:

$$fit_{optimal} = \min_{j=1..50}(fit_j) \tag{41}$$

where $j = 1, 2, \ldots, 50$ represents the iteration number.

Figure 6 shows that the fitness value converges to an optimal value. Although the computational cost of this method may be high, it is the best way to find the optimal values of the control variables, because it tests every possible solution and evaluates its fitness. In addition, it will be used only one time at the beginning, so time is not very important here. The designed GA gives the following solution: $GK_1 = 1$ then $K_1 = 1I_{2\times 2}$, $GK_2 = 19$ then $K_2 = 19I_{2\times 2}$, $\gamma_1 = G\gamma_1 = 0.5625$, $\gamma_2 = G\gamma_2 = 0.0566$ and $G\psi = 470$ then $\psi_i = 470I_{128\times 128}$. $\mu = 47$. $B = GB = 27$. then $\Phi(e_2) = 27e_2$.

Figures 7 and 8 show the tracking error of link 1 and link 2, respectively.

The maximum tracking error for link 1 in the last 10 seconds of simulation is $\max_{t\in[50,60]} e_1^1(t) = 8.14\times$

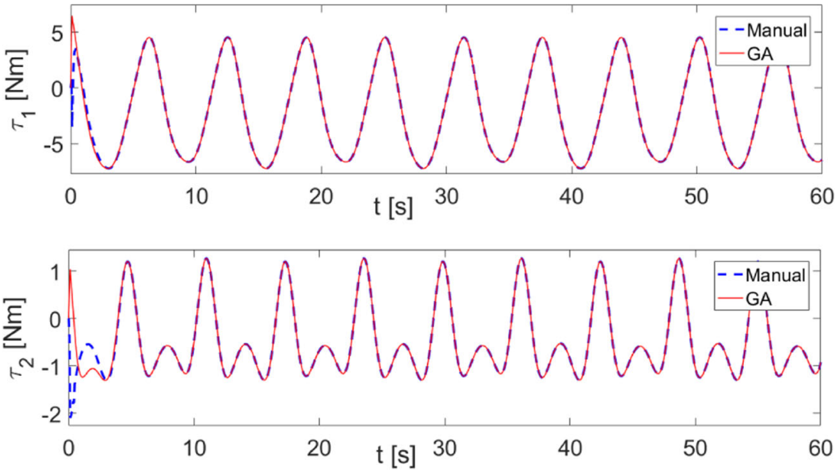


Fig. 14 Input torques when the parameters of the controller are tuned by GA and manually

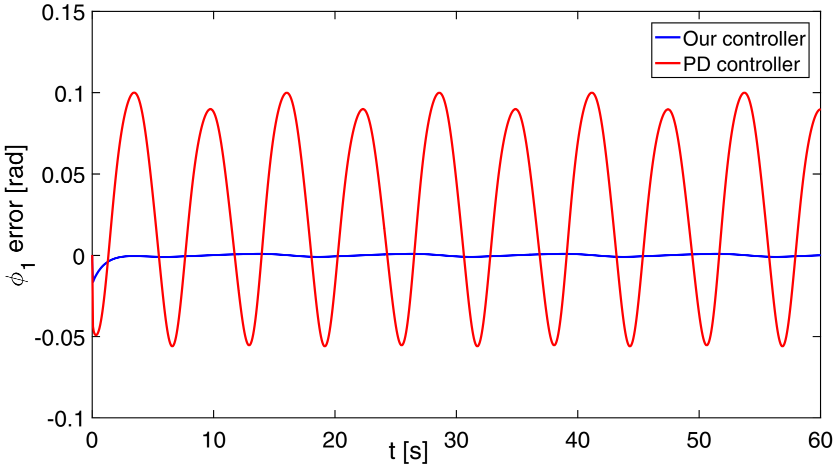


Fig. 15 Tracking errors of the PD controller and the proposed controller for link 1

$10^{-4} rad$, for link 2, the maximum tracking error is $\max_{t\in[50,60]} e_1^2(t) = 8.24 \times 10^{-4} rad$. It is clear that the tracking errors are bounded at very small values. It indicates that the drag forces are compensated successfully by the designed DO. It is clear that DO can track the changes in the drag force with a small error, as it is seen in Figs. 9 and 10. As can be seen from Fig. 11, the input torques do not exceed 7.5 Nm for Joint 1 and 1.5 Nm for Joint 2, which are not beyond the limits of the joint actuators.

In order to show the benefits of using a GA to tune the parameters of the controller, a comparison study between a controller tuned by the GA and one that is tuned manually is done. During manual tuning process, if the designer is lucky to come close to the optimal values of the parameters, he may end up with the following values for the parameters: $K_1 = 5I_{2\times 2}$, $K_2 = 20I_{2\times 2}$, $\gamma_1 = 0.5$, $\gamma_2 = 0.05$ and $\psi = 400I_{128\times 128}$. The $\mu = 40$. $B = 20$. then $\Phi(e_2) = 20e_2$.

Figure 12 illustrates the difference in the tracking errors when GA is used and when the controller is tuned manually. It is clear that although the difference is small, the use of a GA results in better tracking errors. Tracking errors change smoothly, which has good impacts on the motors. Figure 13 shows the difference in the DO performance. It is very clear that the errors in disturbance estimation errors are smaller when GA is used. On the other hand, the input torques are almost identical except at the beginning, as it appears in Fig. 14.

The classical PD control can solve the problem of uncertainties with a small tracking error only if K_p is chosen high enough (around 1500) to compensate for the errors in the estimation of the dynamics, a high value of K_p results in a better disturbance rejection. However the gain that can be used in a PD controller has limitations due to manipulator structure flexibility and actuator time delays. Therefor, the previous gain is not reachable. We design the PD controller gains as follows: $K_p = 70I_{2\times 2}$ and $K_v = 30I_{2\times 2}$. The tracking errors of the PD controller is not even close to our controller errors, as shown in Figs. 15 and 16.

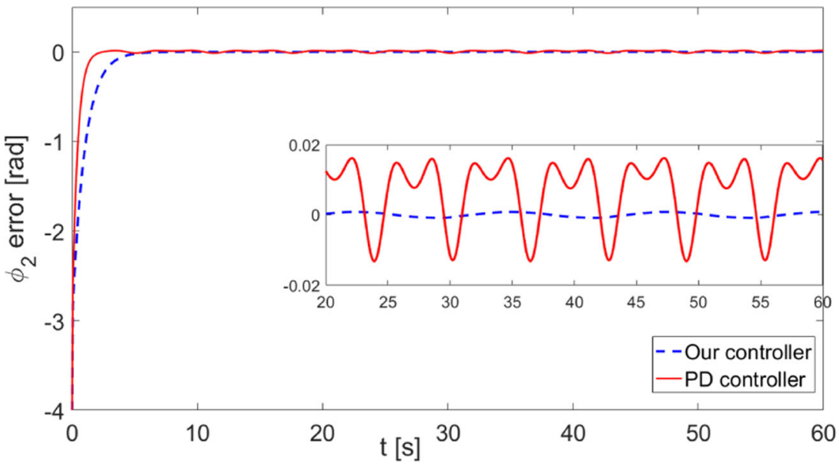


Fig. 16 Tracking errors of the PD controller and the proposed controller for link 2

6 Conclusion

In this paper, an underwater robotic manipulator control system with full-state feedback is developed for trajectory tracking task. The considered system has n-DOF, and it is subjected to hydrodynamic forces, caused by the difference between the water current velocity and the manipulator velocity. The controller is designed based on two approaches, the adaptive NN approach and the DO approach. Since this system suffers of a high level of uncertainties, an adaptive NN is used to compensate for the uncertain parts of the system dynamics as in [20]. The effects of the water flow are considered as an external disturbance. A DO is applied to approximate it, as in [24]. In order to tune the parameters of the proposed controller, an evolutionary computation based approach in the form of a GA is used. The parameters of the controller are tuned using the created GA. The efficiency of the controller proposed is checked by MATLAB based simulation studies, which has been run for 60 seconds. It is verified that the control system tuned by the GA can achieve the assigned tracking task effectively. The tracking error stays within a very small error band. It is thus verified that NN can solve the uncertainties problem and guarantee good performance with a small approximation error.

To show the efficiency of our controller, we compared it with PD controller. The results obtained show that the tracking errors of our controller are about twenty times less than tracking errors of PD controller.

The performance of the GA-tuned controller is compared with one that is tuned manually, in terms of the achieved tracking errors and the disturbance estimation errors. The comparison results have shown that GA can provide the near optimal values for controller parameters and this leads to smaller errors. The only disadvantage of using GA is the slow convergence rate, which we intend to alleviate in future work.

This research has also shown that the DO technique used with on-land robotic manipulators can be used equally effectively in the design of a DO for underwater manipulators.

In this work, we have assumed that we have sufficiently comprehensive information about the manipulator and its operational environment to enable us to design a controller that will ensure satisfactory performance. However, in reality, the parameters of the underwater environment may vary considerably. It is our intention to verify the robustness of the performance of the controller experimentally under such variations. It is our future work to build a laboratory environment for such experimental studies.

Acknowledgments This work was supported by the National Natural Science Foundation of China under Grant 61873298 the Beijing Natural Science Foundation under Grant 4172041.

Appendix : The proof of the system stability

To prove the stability of the considered system under the designed controller, we consider the Lyapunov function in Eq. 23 and take its time derivative to get:

$$\dot V = -e_1^T K_1 e_1 + e_1^T e_2 + \frac{1}{2} e_2^T \dot M_c e_2 + \frac{1}{2} e_2^T M_c \dot e_2 + \tilde f_T^T \dot{\tilde f}_T + \sum_{i=1}^{N} \tilde W_i^T \psi_i^{-1} \dot{\tilde W}_i \tag{42}$$

Substituting Eq. 15 and 22 in Eq. 42 results in:

$$\dot V = -e_1^T K_1 e_1 + e_1^T e_2 + \frac{1}{2} e_2^T \dot M_c e_2 + e_2^T \Big[-e_1 - K_2 e_2 + C_c \alpha + \hat W^T \Theta(h) + \hat f_T - C_c x_2 - W^T \Theta(h) - f_T\Big] + \tilde f_T^T \dot{\tilde f}_T + \sum_{i=1}^{N} \tilde W_i^T \psi_i^{-1} \dot{\tilde W}_i \tag{43}$$

$$\dot V = -e_1^T K_1 e_1 - e_2^T K_2 e_2 + \frac{1}{2} e_2^T (\dot M_c - 2C_c) e_2 + e_2^T \left[\tilde W^T \Theta(h) + \tilde f_T\right] + \tilde f_T^T \dot{\tilde f}_T + \sum_{i=1}^{N} \tilde W_i^T \psi_i^{-1} \dot{\tilde W}_i \tag{44}$$

$\frac{1}{2} e_2^T (\dot M_c - 2C_c) e_2 = 0$ because $\dot M_c - 2C_c$ is a skew symmetric matrix. Substituting Eq. 21 in 44 yields:

$$\dot V = -e_1^T K_1 e_1 - e_2^T K_2 e_2 + e_2^T \left[\tilde W^T \Theta(h) + \tilde f_T\right] - \tilde f_T^T B M_c^{-1} \tilde f_T + \tilde f_T^T B M_c^{-1} W \Theta(h) - \tilde f_T^T \dot f_T + \sum_{i=1}^{N} \tilde W_i^T \psi_i^{-1} \dot{\tilde W}_i \tag{45}$$

We have $\tilde W_i = \hat W_i - W_i$, then we can write

$$\sum_{i=1}^{N} \tilde W_i^T \psi_i^{-1} \dot{\tilde W}_i = -\sum_{i=1}^{N} \tilde W_i^T [\Theta_i(h) e_{2,i} + \gamma_i |e_2| \hat W_i] + \sum_{i=1}^{N} \tilde W_i^T \psi_i^{-1} \dot W_i \tag{46}$$

Substituting in Eq. 45 results in:

$$\dot V \le - e_1^T K_1 e_1 - e_2^T K_2 e_2 + e_2^T \tilde f_T - \tilde f_T^T B M_c^{-1} \tilde f_T - \tilde f_T^T \dot f_T + \tilde f_T^T B M_c^{-1} W \Theta(h) - \sum_{i=1}^{N} \tilde W_i^T \gamma_i |e_2| \hat W_i \tag{47}$$

We have :

$$-e_2^T \tilde f_T \le \frac{1}{2} \|e_2\|^2 + \frac{1}{2} \|\tilde f_T\|^2 \tag{48}$$

$$-\tilde f_T^T \dot f_T \le \frac{1}{2} \tilde f_T^T \tilde f_T + \frac{1}{2} \xi^2 \tag{49}$$

$$\tilde{f}_T^T BM_c^{-1} W\Theta(h) \le \frac{1}{2}\|BM_c\|^2\|\tilde{f}_T\|^2 + \frac{1}{2}\sum_{i=1}^{N}\|W_i\|^2\|\Theta_i(h)\|^2 \quad (50)$$

$$-\sum_{i=1}^{N}\tilde{W}_i^T\gamma_i|e_2|\hat{W}_i \le \frac{1}{2}e_2^Te_2+\frac{1}{8}\sum_{i=1}^{N}\gamma_i^2\left(\|\tilde{W}_i\|^2-\|W_i\|^2\right)^2 \quad (51)$$

According to Proposition 1, we can write:

$$\|\tilde{W}_i\| \le \frac{\epsilon_i}{\gamma_i}+\|W_i\| = d \quad (52)$$

For the activation function of the RBF NN, there is a constant $\epsilon > 0$ where $\|\Theta_i(h)\| \le \epsilon, i = 1, 2, \ldots, N$. Substituting Eqs. 48–52 in Eq. 47 results in:

$$\begin{aligned}\dot{V} \le &- e_1^TK_1e_1 - e_2^T(K_2 - I_{n\times n})e_2 + \frac{1}{2}\xi^2\\ &- \tilde{f}_T^T\left(-\frac{\|BM_c^{-1}\|^2+2}{2}I_{n\times n} + BM_c^{-1}\right)\tilde{f}_T\\ &- \sum_{i=1}^{n}\frac{\gamma_i^2}{4}\|W_i\|^2\|\tilde{W}_i\|^2 + \sum_{i=1}^{n}\frac{\gamma_i^2}{8}\left(\|W_i\|^4+d^4\right)\\ \le &- aV + b\end{aligned} \quad (53)$$

where

$$\begin{aligned}b =& \sum_{i=0}^{n}\frac{\gamma_i+\epsilon^2}{2}\|W_i\|^2 + \sum_{i=0}^{n}\frac{\gamma_i^2}{8}\left(\|W_i\|^4+d^4\right)+\frac{1}{2}\xi^2\\ a =& \min\Bigg(2\lambda_{\min}(K_1), \frac{2\lambda_{\min}(K_2-I_{n\times n})}{\lambda_{\max}(M)},\\ &2\lambda_{\min}\left(BM_c^{-1}-\left(1+\frac{1}{2}\|BM_c^{-1}\|^2\right)I_{n\times n}\right),\\ &\min_{i=1,2,..,n}\left(\frac{\gamma_i^2\|W_i\|^2}{2\lambda_{\max}(\psi_i^{-1})}\right)\Bigg)\end{aligned} \quad (54)$$

In order to guarantee a positive value of a, the gain matrices K_1, K_2 and $\Phi(e_2)$ are designed to guarantee the following conditions:

$$\begin{aligned}&\lambda_{\min}(K_1) > 0, \quad \lambda_{\min}\left(BM_c^{-1}-\left(1+\frac{1}{2}\|BM_c^{-1}\|^2\right)I_{n\times n}\right) > 0,\\ &\lambda_{\min}\left(K_2 - I_{n\times n}\right) > 0\end{aligned} \quad (55)$$

Multiply both sides of Eq. 53 with e^{at} to get:

$$\frac{d}{dt}\left(Ve^{at}\right) \le be^{at} \quad (56)$$

Integrating Eq. 56 over the interval $[0, t]$ yields:

$$V \le V(0) + \frac{b}{a} \quad (57)$$

Since all terms of Lyapunov function (23) are positive, for e_1 we can write:

$$\frac{1}{2}\|e_1\|^2 \le V(0)+\frac{b}{a} \to \|e_1\| \le 2(\sqrt{V(0)+\frac{b}{a}}) = \sqrt{S} \quad (58)$$

The other error signals boundedness can be proved in the same way. Thus the stability of this system is proved.

References

1. Yang, C., Wu, H., Li, Z., He, W., Wang, N., Su, C.Y.: Mind control of a robotic arm with visual fusion technology. IEEE Trans. Ind. Inf. **14**(9), 3822–3830 (2018)
2. Huang, H., Tang, Q., Li, H., Liang, L., Li, W., Pang, Y.: Vehicle-manipulator system dynamic modeling and control for underwater autonomous manipulation. Multibody Sys.Dyn. **41**(2), 125–147 (2017)
3. Wang, H., Wang, C., Chen, W., Liang, X., Liu, Y.: Three dimensional dynamics for cable-driven soft manipulator. IEEE/ASME Trans. Mechatron. **22**(1), 18–28 (2017)
4. Wang, H., Zhang, R., Chen, W., Liang, X., Pfeifer, R.: Shape detection algorithm for soft manipulator based on fiber bragg gratings. IEEE/ASME Trans. Mechatron. **21**(6), 2977–2982 (2016)
5. Xiao, B., Yin, S., Kaynak, O.: Tracking control of robotic manipulators with uncertain kinematics and dynamics. IEEE Trans. Ind. Electron. **63**(10), 6439–6449 (2016)
6. Tutsoy, O., Barkana, D.E., Tugal, H.: Design of a completely model free adaptive control in the presence of parametric, non-parametric uncertainties and random control signal delay. ISA Trans. **76**, 67–77 (2018)
7. Tutsoy, O.: Design and comparison base analysis of adaptive estimator for completely unknown linear systems in the presence of OE noise and constant input time delay. Asian J. Control **18**, 1020–1029 (2016)
8. He, W., Ge, W., Li, Y., Liu, Y.J., Yang, C., Sun, C.: Model identification and control design for a humanoid robot. IEEE Trans. Syst. Man Cybern. Syst. Hum. **47**(1), 45–57 (2017)
9. Chen, C.L.P., Wen, G.-X., Liu, Y.-J., Liu, Z.: Observer-based adaptive backstepping consensus tracking control for high-order nonlinear semi-strict-feedback multiagent systems. IEEE Trans. Cybern. **46**(7), 1591–1601 (2016)
10. He, W., Li, Z., Chen, C.L.P.: A survey of human-centered intelligent robots: issues and challenges. IEEE/CAA J. Autom. Sin. **4**(4), 602–609 (2017)
11. Yang, C., Li, Z., Cui, R., Xu, B.: Neural network-based motion control of an underactuated wheeled inverted pendulum model. IEEE Trans. Neural Net. Learn. Syst. **25**(11), 2004–2016 (2014)
12. Li, Z., Huang, Z., He, W., Su, C.-Y.: Adaptive impedance control for an upper limb robotic exoskeleton using biological signals. IEEE Trans. Ind. Electron. **64**(2), 1664–1674 (2017)
13. Zhang, S., Dong, Y., Ouyang, Y., Yin, Z., Peng, K.: Adaptive neural control for robotic manipulators with output constraints and uncertainties. IEEE Trans. Neural Net. Learn. Syst. **29**(11), 5554–5564 (2018)
14. Liu, D., Xu, Y., Wei, Q., Liu, X.: Residential energy scheduling for variable weather solar energy based on adaptive dynamic programming. IEEE/CAA J. Autom. Sin. **5**(1), 36–46 (2018)
15. Dai, S., Wang, C., Wang, M.: Dynamic learning from adaptive neural network control of a class of nonaffine nonlinear systems. IEEE Trans. Neural Net. Learn. Syst. **25**(1), 111–123 (2014)

16. Dai, S., Wang, M., Wang, C.: Neural learning control of marine surface vessels with guaranteed transient tracking performance. IEEE Trans. Ind. Electron. **63**(3), 1717–1727 (2016)
17. Wang, L., Liu, Z., Chen, C.L.P., Zhang, Y., Lee, S., Chen, X.: Energy-efficient SVM learning control system for biped walking robots. IEEE Trans. Neural Net. Learn. Syst. **24**(5), 831–837 (2013)
18. Liu, Z., Lai, G., Zhang, Y., Chen, X., Chen, C.L.P.: Adaptive neural control for a class of nonlinear time-varying delay systems with unknown hysteresis. IEEE Trans. Neural Net. Learn. Sys. **25**(12), 2129–2140 (2014)
19. Sun, C., He, W., Hong, J.: Neural network control of a flexible robotic manipulator using the lumped spring-mass model. IEEE Trans. Syst. Man Cybern. Syst. Hum. **47**(8), 1–12 (2016)
20. He, W., Ge, S.S., Li, Y., Chew, E., Ng, Y.S.: Neural network control of a rehabilitation robot by state and output feedback. J. Intell. Robot. Syst. **80**(1), 15–31 (2015)
21. Wang, F.Y., Zheng, N.N., Cao, D., Martinez, C.M., Li, L., Liu, T.: Parallel driving in CPSS: A unified approach for transport automation and vehicle intelligence. IEEE/CAA J. Autom. Sin. **4**(4), 577–587 (2017)
22. Yang, C., Wang, X., Cheng, L., Ma, H.: Neural-learning-based telerobot control with guaranteed performance. IEEE Trans. Cybern. **47**(10), 3148–3159 (2017)
23. Li, Y., Ge, S.S.: Human–robot collaboration based on motion intention estimation. IEEE/ASME Trans. Mechatron. **19**(3), 1007–1014 (2014)
24. He, W., Huang, H., Ge, S.S.: Adaptive neural network control of a robotic manipulator with time-varying output constraints. IEEE Trans. Cybern. **47**(10), 3136–3147 (2017)
25. Li, Z., Su, C.-Y., Li, G., Su, H.: Fuzzy approximation-based adaptive backstepping control of an exoskeleton for human upper limbs. IEEE Trans. Fuzzy Syst. **23**(3), 555–566 (2015)
26. Huang, P., Wang, D., Meng, Z., Zhang, F., Liu, Z.: Impact dynamic modeling and adaptive target capturing control for tethered space robots with uncertainties. IEEE/ASME Trans. Mechatron. **21**(5), 2260–2271 (2016)
27. Huang, P., Zhang, F., Cai, J., Wang, D.: Dexterous tethered space robot: design, measurement, control and experiment. IEEE Trans. Aerosp. Electron. Syst. **3**, 53 (2017)
28. Mohan, S., Kim, J.: Indirect adaptive control of an autonomous underwater vehicle-manipulator system for underwater manipulation tasks. Ocean Eng. **54**, 233–243 (2012)
29. Dos Santos, C.H.F., De Pieri, E.R.: Functional machine with takagi-sugeno inference to coordinated movement in underwater vehicle-manipulator systems. IEEE Trans. Fuzzy Syst. **21**(6), 1105–1114 (2013)
30. Zhang, M.J., Chu, Z.Z.: Adaptive sliding mode control based on local recurrent neural networks for underwater robot. Ocean Eng. **45**, 56–62 (2012)
31. Xu, J., Wang, M., Qiao, L.: Dynamical sliding mode control for the trajectory tracking of underactuated unmanned underwater vehicles. Ocean Eng. **105**, 54–63 (2015)
32. Lee, M., Choi, H.S.: A robust neural controller for underwater robot manipulators. IEEE Trans. Neural Netw. **11**(6), 1465–1470 (2000)
33. Zhang, Y., Sun, J., Liang, H., Li, H.: Event-triggered adaptive tracking control for multiagent systems with unknown disturbances. IEEE Trans. Cybern. 1–12 (2018)
34. Chen, M.: Disturbance attenuation tracking control for wheeled mobile robots with skidding and slipping. IEEE Trans. Ind. Electron. **64**(4), 3359–3368 (2017)
35. Li, Z., Su, C.Y., Wang, L., Chen, Z., Chai, T.: Nonlinear disturbance observer-based control design for a robotic exoskeleton incorporating fuzzy approximation. IEEE Trans. Ind. Electron. **62**(9), 5763–5775 (2015)
36. Liu, S., Liu, Y., Wang, N.: Nonlinear disturbance observer-based backstepping finite-time sliding mode tracking control of underwater vehicles with system uncertainties and external disturbances. Nonlinear Dyn. **88**(1), 465–476 (2017)
37. Cui, R., Zhang, X., Cui, D.: Adaptive sliding-mode attitude control for autonomous underwater vehicles with input nonlinearities. Ocean Eng. **123**, 45–54 (2016)
38. Cui, R., Chen, L., Yang, C., Chen, M.: Extended state observer-based integral sliding mode control for an underwater robot with Unknown disturbances and uncertain nonlinearities. IEEE Trans. Ind. Electron. **64**(8), 6785–6795 (2017)
39. Tsai, C.C., Huang, H.C., Chan, C.K.: Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation. IEEE Trans. Ind. Electron. **58**(10), 4813–4821 (2011)
40. Sahu, D., Mishra, A.K.: Mobile robot path planning by genetic algorithm with safety parameter. International Journal of Engineering Science and Computing **7**(8), 14723–14727 (2017)
41. Neath, M.J., Swain, A.K., Madawala, U.K., Thrimawithana, D.J.: An optimal PID controller for a bidirectional inductive power transfer system using multiobjective genetic algorithm. IEEE Trans. Power Electron. **29**(3), 1523–1531 (2014)
42. Dimeo, R., Lee, K.: Boiler-turbine control system design using a genetic algorithm. IEEE Trans. Energy Convers. **10**(4), 752–759 (1995)
43. Chocron, O., Vega, E.P., Benbouzid, M.: Dynamic reconfiguration of autonomous underwater vehicles propulsion system using genetic optimization. Ocean Eng. **156**(2017), 564–579 (2018)
44. Roberge, V., Tarbouchi, M., Labonte, G.: Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. IEEE Trans. Ind. Inf. **9**(1), 132–141 (2013)
45. Siciliano, B., Khatib, O.: Springer Handbook of Robotics. Springer International Publishing, Cham (2016)
46. Kolodziejczyk, W.: The method of determination of transient hydrodynamic coefficients for a single DOF underwater manipulator. Ocean Eng. **153**, 122–131 (2018)
47. il Seo, S., suk Mun, H., ho Lee, J., ha Kim, J.: Simplified analysis for estimation of the behavior of a submerged floating tunnel in waves and experimental verification. Mar. Struct. **44**, 142–158 (2015)
48. Dioguardi, F., Mele, D.: A new shape dependent drag correlation formula for non-spherical rough particles. Experiments and results. Powder Technol. **277**, 222–230 (2015)
49. Chen, Z., Shatara, S., Tan, X.: Modeling of biomimetic robotic fish propelled by an ionic polymermetal composite caudal fin. IEEE/ASME Trans. Mechatron. **15**(3), 448–459 (2010)
50. Maza, M., Adler, K., Ramos, D., Garcia, A.M., Nepf, H.: Velocity and drag evolution from the leading edge of a model mangrove forest. J. Geophys. Res. Oceans **122**(11), 9144–9159 (2017)
51. Meng, W., Yang, Q., Si, J., Sun, Y.: Adaptive neural control of a class of output-constrained nonaffine systems. IEEE Trans. Cyber. **46**(1), 85–95 (2016)
52. Na, J., Ren, X., Zheng, D.: Adaptive control for nonlinear pure-feedback systems with high-order sliding mode observer. IEEE Trans. Neural Net. Learn. Syst. **24**(3), 370–382 (2013)
53. Li, Y., Tee, K.P., Chan, W.L., Yan, R., Chua, Y., Limbu, D.K.: Continuous role adaptation for human robot shared control. IEEE Trans. Robot. **31**(3), 672–681 (2015)
54. Zhou, Q., Wang, L., Wu, C., Li, H., Du, H.: Adaptive fuzzy control for nonstrict-feedback systems with input saturation and output constraint. IEEE Trans. Syst. Man Cybern. Syst. Hum. **47**(1), 1–12 (2017)
55. Li, H., Wang, L., Du, H., Boulkroune, A.: Adaptive fuzzy backstepping tracking control for strict-feedback systems with input delay. IEEE Trans. Fuzzy Syst. **25**(3), 642–652 (2017)

56. Li, Z., Huang, B., Ajoudani, A., Yang, C., Su, C.-Y., Bicchi, A.: Asymmetric bimanual control of dual-Arm exoskeletons for human-cooperative manipulations. IEEE Trans. Robot. **34**(1), 264–271 (2018)
57. Wu, G., Sun, J., Chen, J.: Optimal linear quadratic regulator of switched systems. IEEE Trans. Autom. Control **PP**(8), 1 (2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Tony Salloom received his B.Eng. in Electronics and Computer Engineering from Aleppo University, Aleppo, Syria, in 2008, and his M.Eng. in Information and Communication Engineering from University of Science and Technology Beijing, Beijing, China, in 2016. He was a database engineer at Syrian Telecommunication Company, Syria, from 2009 to 2013. He is currently pursuing the Ph.D. degree with the School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing, China. His current research interests include neural networks, robotics and intelligent control systems.

Xinbo Yu received his B.Eng. in Control Technology and Instruments from University of Science and Technology Beijing, Beijing, China, in 2013, and his M.Eng. in Mechanical Engineering from Guilin University of Electronic Technology, Guilin, China, in 2016. Now he is currently pursuing the Ph.D. degree with the School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing, China. His current research interests include robotics, intelligent control and human-robot interaction.

Wei He (S'09-M'12-SM16) received his B.Eng. and his M.Eng. degrees from College of Automation Science and Engineering, South China University of Technology (SCUT), China, in 2006 and 2008, respectively, and his PhD degree from Department of Electrical & Computer Engineering, the National University of Singapore (NUS), Singapore, in 2011. He is currently working as a full professor in School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing, China. He has co-authored 2 book published in Springer and published over 100 international journal and conference papers. He has been awarded a Newton Advanced Fellowship from the Royal Society, UK. He is a recipient of the IEEE SMC Society Andrew P. Sage Best Transactions Paper Award in 2017. He serves as an Associate Editor of IEEE Transactions on Control Systems Technology, IEEE Transactions on Neural Networks and Learning Systems, IEEE Transactions on Systems, Man, and Cybernetics: Systems and IEEE Access IEEE/CAA Journal of Automatica Sinica, Neurocomputing, and an Editor of IEEE/CAA Journal of Automatica Sinica, Assemply Automation, and Journal of Intelligent & Robotic Systems. He is the member of the IFAC TC on Distributed Parameter Systems, IFAC TC on Computational Intelligence in Control and IEEE CSS TC on Distributed Parameter Systems. His current research interests include robotics, distributed parameter systems and intelligent control systems.

Okyay Kaynak (F'03) received the B.Sc. (first-class honors) and the Ph.D. degrees in electronic and electrical engineering from the University of Birmingham, Birmingham, U.K., in 1969 and 1972, respectively. From 1972 to 1979, he held various positions within the industry. In 1979, he joined the Department of Electrical and Electronics Engineering, Bogazici University, Istanbul, Turkey, where he is presently a Professor Emeritus, holding the UNESCO Chair on Mechatronics and also a "1000 Talents Program" Professor at the University of Science and Technology Beijing, China. He has hold long-term (near to or more than a year) Visiting Professor/Scholar positions at various institutions in Japan, Germany, U.S.A., Singapore and China. His current research interests include the fields of intelligent control and CPS. He has authored three books and edited five and authored or coauthored more than 400 papers that have appeared in various journals, books, and conference proceedings. He is one of the 2017 Highly Cited Researchers. Dr. Kaynak has been on many committees of the IEEE and was the President of IEEE Industrial Electronics Society during 2002-2003.