Ocean Engineering 266 (2022) 112994

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

Constrained fault-tolerant thrust allocation of ship DP system based on a novel quantum-behaved squirrel search algorithm ☆

Zhihao Yu, Jialu Du *

School of Information Science and Technology, Dalian Maritime University, Dalian, Liaoning, 116026, China

ARTICLE INFO

Keywords:
Dynamic positioning
Thrust allocation
Fault-tolerant
Optimization
Squirrel search algorithm

ABSTRACT

The thrust allocation (TA) of ship dynamic positioning (DP) systems is to determine the desired thrust force and direction of each thruster of a ship such that all the thrusters jointly produce the generalized control forces commanded by DP controllers. Thruster faults inevitably occur because thrusters operate in the complex ocean environment for a long time. In addition, due to physical limitations of thrusters, the desired thrust force and direction of thrusters both have magnitude and rate constraints. In the presence of loss-of-effectiveness (LOE) faults as well as magnitude and rate constraints of thrusters of ships, the TA of their DP systems can be formulated as a constrained fault-tolerant optimization problem. This paper creates a novel quantum-behaved squirrel search algorithm (QSSA) by combining the squirrel search algorithm with a quantum Delta potential well model, and whereby solves this constrained fault-tolerant TA optimization problem. Simulation studies on a scale model supply ship verify the effectiveness of the QSSA-based constrained fault-tolerant TA of the DP ship under thruster LOE faults.

1. Introduction

Ships equipped with dynamic positioning (DP) systems can automatically maintain their positions and headings under ocean environmental disturbances with the assistance of their own thrusters (Sørensen, 2011). As the marine resource exploration and exploitation shifts from shallow sea to deep ocean, ship DP systems are gradually gaining more attentions in the field of ocean engineering as pipe-laying, drilling, hydrographic surveying and ocean observation (Fossen, 2011). The ship DP control system mainly consists of the DP controller, the thrust allocation (TA), the propulsion system of ships and the measurement system. The schematic of the ship DP control system is shown in Fig. 1. The TA is to map the generalized control force from the DP controller into the desired thrust force and direction of the thrusters of the propulsion system, which is one of the vital parts of the ship DP control system.

For the ship mounted with thrusters in fixed direction alone or in combination with rudders, Fossen and Sagatun (1991) and Fossen (1994) formulated the TA problem of its DP system as a least-squares optimization problem and then solved the problem using Lagrange Multipliers, which allowed the desired thrust forces of thrusters to be calculated simply using pseudo-inverses. The above TA problem neglects the physical limitations of thrusters; however, the physical limitations cause the magnitude and rate of both their thrust forces and directions to be constrained in practice. As a result, the commanded generalized control forces cannot be generated effectively by these thrusters without the consideration of the constraints, which decreases the performance of the DP control system. Considering the magnitude and rate constraints on the thrust forces, Johansen et al. (2005) formulated the TA problem of ships with thrusters in fixed direction as a quadratic programming problem with linear constraints and provided an explicit piecewise linear representation of its optimal solution, which was computed numerically using multi-parametric quadratic programming (mp-QP). Rudder control inputs are termed as rudder angles, and the rudder forces are limited to sectors due to magnitude and rate constraints on rudder angles. For the linear constrained TA problem of ships equipped with main propellers and rudders, Lindegaard and Fossen (2003) proposed an analytic 2-norm optimal method ensuring the continuity of the solutions, where the sector constraint of only one rudder was considered. Using a control-Lyapunov approach, Johansen (2004) developed an optimal dynamic TA algorithm constructed as a dynamic update law for a general class of nonlinear systems, which relaxes the computational complexity considerably compared to the direct nonlinear programming approaches in Lindegaard and Fossen (2003). Johansen et al. (2008) extended

☆ This work was supported in part by the Dalian Science and Technology Innovation Fund Program, China under Grant 2020JJ26GX020; in part by the National Natural Science Foundation of China under Grant 51079013.

* Corresponding author.
E-mail addresses: yzh@dlmu.com.cn (Z. Yu), dujl66@163.com (J. Du).

https://doi.org/10.1016/j.oceaneng.2022.112994
Received 29 March 2022; Received in revised form 10 October 2022; Accepted 22 October 2022
Available online 15 November 2022

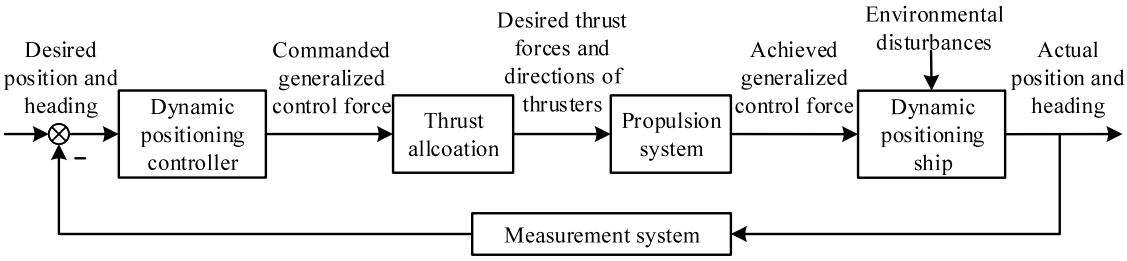


Fig. 1. Schematics of the ship DP control system.

the method in Lindegaard and Fossen (2003) to the general case with any number of rudders while considering the magnitude and rate constraints of thrusters, and the linear constrained TA problem was solved by the mp-QP algorithm. Depending on the number of propeller revolutions per minute, Li et al. (2016) divided the constraints on rudder angles into four parts resulting in four TA optimization sub-problems, and the TA optimization sub-problems were solved using the particle swarm optimization (PSO). Redistributed pseudo-inverse methods (see Shi et al., 2010; Virnig and Bodden, 1994) were developed for the above linear constrained TA problems. The first step of the redistributed pseudo-inverse method is to solve the unconstrained TA problem using pseudo-inverses, and then the obtained thrust forces of thrusters are saturated to satisfy the constraints. And the unsaturated thrust forces are re-computed by solving a reduced unconstrained TA problem using a reduced pseudo-inverse.

The TA problem of ships with azimuth thrusters is generally a non-convex optimization problem subject to a nonlinear constraint on the azimuth angles of the azimuth thrusters. The thruster configuration matrix in this nonlinear constraint becomes singular when the azimuth thrusters are at the same azimuth angle, which compels the thrusters to generate thrust force with unacceptably large magnitudes and decreases the maneuverability of DP ships. Sørdalen (1997) handled the singular configuration using a modification of the singular-value decomposition, and solved the constrained TA problem utilizing pseudo-inverses with the extended thrust representation. To avoid the singular configuration, Johansen et al. (2004) considered an additional objective of singularity avoidance as well as the magnitude and rate constraints of azimuth thrusters, and the nonlinear constrained TA problem with a non-convex term of singularity avoidance was solved using a sequential quadratic programming (SQP) method. Scibilia and Skjetne (2012) handled the singularity problem by opportunely defining the future expected thrust force, which allows to remove the non-convex term from the cost function in Johansen et al. (2004). With the consideration of magnitude and rate constraints on both RPM and azimuth angles, Ruth et al. (2007) presented a TA method based on a constrained quadratic cost function capable of handling rotating azimuth thrusters. The hydrodynamic interaction effects on thrusters, including thruster-hull interactions and thruster-thruster interactions, significantly affect the delivered thrust force (Cozijn and Hallmann, 2012, 2013), and therefore they should be considered during DP operations. Taking into account the hydrodynamic interaction as well as the magnitude and rate constraints of thrusters, Arditti et al. (2015) presented an efficiency function dependent on the azimuth angle in the formulation of the nonlinear constrained TA optimization problem, and the problem was solved by a modified SQP method. Further, in Arditti et al. (2019), a complete representation of the hydrodynamic interactions in thrust generation was presented. For thruster ventilation events caused by extreme weather conditions, Ruth et al. (2009) proposed the antispin TA strategy to reduce thrust losses and the possibility of multiple thruster ventilation events. With regard to the nonlinear constrained TA problem of ships equipped with azimuth thrusters, the SQP methods (see Li and Yang, 2019; Arditti et al., 2015; Johansen et al., 2004) have become the most widely employed TA algorithm in practice due to its easy implementation although several other iterative numerical optimization algorithms developed in Wang et al. (2013), Yadav et al. (2014), Mauro and Nabergoj (2016), Wu et al. (2016) and Khare and Agrawal (2019) can provide more accurate solutions than SQP.

The aforementioned literature did not take thruster faults into account. Due to thrusters operate in the complex ocean environment for a long time, they inevitably undergo faults including loss-of-effectiveness (LOE), outrage and stuck (Wang and Yang, 2016). As for the ship's DP control system, the issue of thruster fault tolerance is typically handled within the TA algorithm (Johansen and Fossen, 2013). In particular, considering thruster LOE faults in the linear constrained TA problem, several TA algorithms have been reported. Omerdic and Roberts (2004) utilized efficiency factors to represent the LOE faults and integrated pseudo-inverses weighted by the efficiency factors into a thruster fault diagnosis and accommodation system. Sarkar et al. (2002) incorporated a dynamic state feedback technique into the pseudo-inverse based TA method weighted by efficiency factors, and thus generating reference thrust forces within the magnitude constraint of thrusters. Soylu et al. (2008) proposed a TA method based on minimizing the infinity-norm of thrust forces weighted by efficiency factors, where a recurrent neural network was proposed to obtain real-time computation rates. Casavola and Garone (2010) presented an optimization-based TA method integrated with a parameter estimation scheme for the unknown efficiency factors. Cristofaro and Johansen (2014) employed pseudo-inverses weighted by unknown time-varying efficiency factors, which were estimated by unknown input observers. The thrust allocation problem of ships is a special case of the more general control allocation (CA) problem, which also includes CA in flight control, spacecrafts and cars (Ruth, 2008). The CA methods provided in Alwi and Edwards (2008), Shen et al. (2017) and Argha et al. (2019) for the linear constrained fault-tolerant CA problem of aircrafts or spacecrafts can be extended to that of ships.

Offshore supply ships as DP ships are widely employed not only to help in exploration and drilling of oil but also for providing necessary supplies to the construction and marine vehicles located at the deep ocean. Due to the vital role of offshore supply vessels in the field of ocean engineering, the safe operation of them in the presence of thruster faults is required to be guaranteed. For offshore supply ships mounted with azimuth thrusters and tunnel thrusters subject to LOE faults as well as the magnitude and rate constraints, this paper focuses on solving the constrained fault-tolerant TA problem of them meanwhile considering additional objectives of optimizing power consumption, minimizing tear-and-wear and singularity avoidance. For such a TA problem with nonlinearly formulated objectives and constraints, Johansen and Fossen (2013) recommends the iterative numerical optimization algorithm, e.g. the genetic algorithm (GA) (Mauro and Nabergoj, 2016) and the PSO (Wang et al., 2013). However, the convergence rate of those TA algorithms is not fast enough when solving the nonlinear constrained fault-tolerant TA problem.

The squirrel search algorithm (SSA) (Jain et al., 2019), a recently proposed iterative numerical optimization algorithm, has been proven through numerical experiments to have a faster convergence rate than many other algorithms including GA and PSO. It is widely adopted for various engineering optimization problems (Basu, 2019; Zheng and

Luo, 2019; Khare and Agrawal, 2019; Cao et al., 2021; Liu et al., 2021; Sakthivel et al., 2021). The constrained fault-tolerant thrust allocation problem addressed in this paper is a non-convex optimization problem with a nonlinear constraint. This problem has lots of local minima that may trap algorithms into sub-optimal solutions, which makes the algorithms take extra iterations to jump out of the local minima. When SSA is tried to solve the nonlinear constrained fault-tolerant TA problem, it always takes too many iterations as well as too much time for the online optimization of TA. In order to satisfy the real-time requirement of TA, suboptimal solutions have to be used as optimization results when the SSA is not converge yet, and the accuracy of TA is sacrificed. Therefore, the convergence rate of SSA needs to be further improved. Fortunately, the quantum-inspired GA (Narayanan and Moore, 1996) provides an effective approach utilizing concepts and principles of quantum mechanisms to improve the convergence rate. In the recent decade, quantum Delta potential well models of various iterative optimization algorithms have been constructed (see Li et al., 2012; Haibin and Cong, 2015; Hannan et al., 2018; Mahdi et al., 2019; Alvarez-Alvarado et al., 2021), whereby the convergence rate of these algorithms are enhanced. According to the mechanisms of quantum-inspired algorithms, their search agents can appear anywhere during iterations so that the population diversity is enriched (Hao et al., 2010). As a consequence, the quantum-inspired algorithms can obtain high probabilities of escaping from the local optimal solution and achieving the global optimal solution, thus attaining a faster convergence rate.

Motivated by the aforementioned contents, for solving the TA optimization problem subject to LOE faults as well as thruster magnitude and rate constraints, this paper creates a novel quantum-behaved squirrel search algorithm (QSSA) by combining SSA with a quantum Delta potential well model. QSSA solves the nonlinear constrained fault-tolerant TA optimization problem effectively meanwhile ensuring the safety of the DP ship under thruster LOE faults and satisfying the real-time requirement as well. The performance of QSSA for the nonlinear fault-tolerant TA problem is better than that of SSA, GA, PSO and SQP with higher accuracy, lower total power consumption and less tear-and-wear of thrusters. What is more, the thruster LOE fault tolerance is considered in the nonlinear constrained TA problem involving azimuth thrusters, which is more practical; however, the TA or CA algorithms developed in Omerdic and Roberts (2004), Sarkar et al. (2002), Soylu et al. (2008), Casavola and Garone (2010), Cristofaro and Johansen (2014), Alwi and Edwards (2008), Shen et al. (2017) and Argha et al. (2019) can only be adopted for the unconstrained or the linear constrained fault-tolerant TA problems of ships.

The remaining parts are organized as follows. The constrained fault-tolerant TA optimization problem is formulated in Section 2. Section 3 provides the design of QSSA and the numerical experiment result of solving a set of benchmark functions. Section 4 provides simulation studies on a scale model supply ship. Section 5 contains a conclusion.

2. Formulations of the constrained fault-tolerant TA optimization problem

2.1. Thruster configuration

As for the ship in 3 dimensions of freedom of the horizontal plane, the generalized control force $\boldsymbol{\tau}_c \in \mathbb{R}^3$ commanded by the DP controller consists of the commanded surge force F_x, sway force F_y and yaw moment F_n, i.e. $\boldsymbol{\tau}_c = [F_x, F_y, F_n]^{\mathrm{T}}$. $\boldsymbol{\tau}_c$ is distributed by the TA algorithm as individual thrust forces produced by m thrusters of the ship. The magnitude of the thrust forces is denoted by a thrust force vector $\boldsymbol{f} = [f_1, f_2, \dots, f_m] \in \mathbb{R}^m$. The direction of the thruster in the horizontal plane is denoted by an azimuth angle vector $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_m] \in \mathbb{R}^m$. The thrust force vector $\boldsymbol{f}$ is related to the propeller RPM denoted by $\boldsymbol{u} = [u_1, u_2, \dots, u_m]^{\mathrm{T}}$, which is conveniently expressed as

$$f_i = K_i u_i |u_i|, i = 1, 2, \dots m \tag{1}$$

where K_i is the thrust coefficient.

All the thrusters jointly produce the actual generalized control force denoted by $\boldsymbol{\tau} \in \mathbb{R}^3$ that should equal the commanded generalized control force $\boldsymbol{\tau}_c$. And $\boldsymbol{\tau}$ is related to the thrust force $\boldsymbol{f}$ and the azimuth angle $\boldsymbol{\alpha}$:

$$\boldsymbol{\tau} = \boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{f} \tag{2}$$

where $\boldsymbol{T}(\boldsymbol{\alpha}) = [\boldsymbol{T}_1(\alpha_1), \dots, \boldsymbol{T}_m(\alpha_m)]$ is the thrust configuration matrix. For each thruster

$$\boldsymbol{T}_i(\alpha_i) = \begin{bmatrix} \cos\alpha_i \\ \sin\alpha_i \\ l_{x,i}\sin\alpha_i - l_{y,i}\cos\alpha_i \end{bmatrix}, i = 1, 2, \dots, m \tag{3}$$

The installing location of the ith thruster in the horizontal plane is denoted by $(l_{x,i}, l_{y,i})$.

2.2. Modeling of thruster LOE faults

The thruster LOE faults are generally represented as unknown time-varying efficiency factors (Cristofaro and Johansen, 2014). For each thruster, the efficiency factor is $0 \leqslant \eta_i \leqslant 1, i = 1, 2, \dots, m$. For $\eta_i \equiv 1$, it means the thruster is fault-free. $0 \leqslant \eta_i < 1$ means the thruster is suffering from the partial LOE fault, and $\eta_i = 0$ the complete LOE fault especially. The efficiency factors need to be determined before performing the TA procedure during each sampling interval, and they can be estimated using the methods provided in Casavola and Garone (2010). And the estimation of efficiency factors are introduced in Eq. (2) as

$$\boldsymbol{\tau} = \boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{E}\boldsymbol{f} \tag{4}$$

where $\boldsymbol{E} = \mathrm{diag}(\eta_1, \eta_2, \dots, \eta_m)$.

A vector $\boldsymbol{s} \in \mathbb{R}^m$ containing slack variables represents the error between commanded and achieved generalized control forces:

$$\begin{aligned} \boldsymbol{s} &= \boldsymbol{\tau}_c - \boldsymbol{\tau} \\ &= \boldsymbol{\tau}_c - \boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{E}\boldsymbol{f} \end{aligned} \tag{5}$$

2.3. Magnitude and rate constraints of thrusters

The thrust force $\boldsymbol{f}$ is constrained between its lower limit $\underline{\boldsymbol{f}} \in \mathbb{R}^m$ and upper limit $\overline{\boldsymbol{f}} \in \mathbb{R}^m$. Due to the hydrodynamic interaction between azimuth thrusters closed to each other (thruster–thruster interactions), feasible sectors are set so as to avoid thrust loss. Thus, the azimuth angle $\boldsymbol{\alpha}$ of azimuth thrusters is generally constrained between $\underline{\boldsymbol{\alpha}} \in \mathbb{R}^m$ and $\overline{\boldsymbol{\alpha}} \in \mathbb{R}^m$. The rate constraints on the thrust force $\boldsymbol{f}$ and the azimuth angle $\boldsymbol{\alpha}$ are respectively modeled using their maximum increments per second denoted by $\Delta\boldsymbol{f}$ and $\Delta\boldsymbol{\alpha}$. Synthetically, the magnitude and rate constraints on $\boldsymbol{f}$ and $\boldsymbol{\alpha}$ of each sampling interval are given as the followed form (Ruth, 2008):

$$\begin{cases} f_i^- \leqslant f_i \leqslant f_i^+, i = 1, 2, \dots, m \\ \boldsymbol{f}^- = \max(\underline{\boldsymbol{f}}, \boldsymbol{f}_0 - \Delta\boldsymbol{f}h) \\ \boldsymbol{f}^+ = \min(\overline{\boldsymbol{f}}, \boldsymbol{f}_0 + \Delta\boldsymbol{f}h) \\ \alpha_i^- \leqslant \alpha_i \leqslant \alpha_i^+, i = 1, 2, \dots, m \\ \boldsymbol{\alpha}^- = \max(\underline{\boldsymbol{\alpha}}, \boldsymbol{\alpha}_0 - \Delta\boldsymbol{\alpha}h) \\ \boldsymbol{\alpha}^+ = \min(\overline{\boldsymbol{\alpha}}, \boldsymbol{\alpha}_0 + \Delta\boldsymbol{\alpha}h) \end{cases} \tag{6}$$

where $\boldsymbol{f}^- \in \mathbb{R}^m$ and $\boldsymbol{f}^+ \in \mathbb{R}^m$ are the lower and upper limits of the thrust force that all thrusters can produce during a sampling interval. And $\boldsymbol{\alpha}^- \in \mathbb{R}^m$ and $\boldsymbol{\alpha}^+ \in \mathbb{R}^m$ are the lower and upper limits of the azimuth angle during a sampling interval. $\boldsymbol{f}_0$ and $\boldsymbol{\alpha}_0$ are the thrust force and the azimuth angle of the last sampling instant. Scalar h is the sampling interval.

2.4. Objectives

Considering LOE faults as well as magnitude and rate constraints of thrusters, the constrained fault-tolerant TA optimization model is constructed as follow:

$$\min_{\boldsymbol{f},\boldsymbol{s},\boldsymbol{\alpha}\in\Omega} \quad \sum_{i=1}^{m} \boldsymbol{w}_{f,i}|f_i|^{3/2} + \boldsymbol{s}^{\mathrm{T}}\boldsymbol{W}_s\boldsymbol{s} + (\boldsymbol{\alpha}-\boldsymbol{\alpha}_0)^{\mathrm{T}}\boldsymbol{W}_\alpha(\boldsymbol{\alpha}-\boldsymbol{\alpha}_0) + \frac{w_n}{\varepsilon + \det(\boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{T}^{\mathrm{T}}(\boldsymbol{\alpha}))} \tag{7}$$

$$\text{s.t.} \quad \boldsymbol{s} = \boldsymbol{\tau}_c - \boldsymbol{T}(\boldsymbol{\alpha})\boldsymbol{E}\boldsymbol{f} \tag{8}$$

$$f_i^- \leqslant f_i \leqslant f_i^+, i = 1,2,\ldots,m \tag{9}$$

$$\alpha_i^- \leqslant \alpha_i \leqslant \alpha_i^+, i = 1,2,\ldots,m. \tag{10}$$

In the objective function Eq. (7), the first term represents the total power consumption combining the power consumption of m individual thrusters. The second term penalizes the error between commanded and achieved generalized control forces. The third term penalizes the change of the azimuth angle in order to minimize tear-and-wear of azimuth thrusters. Vector $\boldsymbol{w}_f \in \mathbb{R}^m$, diagonal matrices $\boldsymbol{W}_s \in \mathbb{R}^{m\times m}$ and $\boldsymbol{W}_\alpha \in \mathbb{R}^{m\times m}$ contain weights. The last term is to avoid singular configuration of thrusters, where $w_n > 0$ is scalar weight and $\varepsilon > 0$ is a small constant avoiding division by zero. A large w_n ensures high maneuverability of DP ships at the cost of higher power consumption of thrusters and vice versa. The search space Ω refers to the region formed by the constraints Eqs. (8)–(10), which contains all feasible solutions of this TA optimization problem.

Remark 1. The control inputs to the actuators of the propulsion system are the propeller RPM and the azimuth angles. Generally, the thrust forces $\boldsymbol{f}$ are defined as the decision variables of the thrust allocation problem, and the physical limitations of thrusters are given using thrust forces (Johansen et al., 2004; Li and Yang, 2019; Ruth, 2008; Arditti et al., 2019). To make the formulation of the TA problem simple and easy to understand, this paper takes the thrust force as the decision variables rather than the propeller RPM. By solving the TA optimization problem, we have the optimal solution that consists of the optimal thrust force and azimuth angle. And then the optimal thrust force can be converted into the optimal propeller RPM using

$$u_i = \mathrm{sign}(f_i)\sqrt{K_i^{-1}|f_i|}, i = 1,2,\ldots,m. \tag{11}$$

The negative value of f_i here means the thrust force produced by reversely rotating propellers.

3. Quantum-behaved squirrel search algorithm (QSSA)

3.1. Brief introduction of SSA

The inspiration of SSA is the foraging activity of a flying squirrel colony. The squirrels fly among trees in forest to search for food sources. The flying squirrel colony denoted by $\boldsymbol{FS} \in \mathbb{R}^{n\times d}$ represents the location of n squirrels in a d-dimension search space:

$$\boldsymbol{FS} = \begin{bmatrix} \boldsymbol{FS}_1 \\ \vdots \\ \boldsymbol{FS}_n \end{bmatrix} = \begin{bmatrix} FS_{1,1} & \ldots & FS_{1,d} \\ \vdots & \vdots & \vdots \\ FS_{n,1} & \ldots & FS_{n,d} \end{bmatrix}_{n\times d}. \tag{12}$$

The flying squirrel colony is randomly initialized using

$$\boldsymbol{FS}_i = \underline{\boldsymbol{FS}} + \boldsymbol{R}_i \odot (\overline{\boldsymbol{FS}} - \underline{\boldsymbol{FS}}), i = 1,2,\ldots,n \tag{13}$$

where $\boldsymbol{R}_i \in \mathbb{R}^{1\times d}$ is a matrix containing uniform random numbers in range $[0,1]$, and $\odot$ is an operator of the element-wise multiplication. $\underline{\boldsymbol{FS}}, \overline{\boldsymbol{FS}} \in \mathbb{R}^{1\times d}$ respectively contain lower and upper bounds.

It is generally assumed for SSA that only three types of trees are available in forest: 1 hickory tree with the optimal food source, 3 acorn trees with normal food sources and $n-4$ normal trees without any food source (Jain et al., 2019). The location of the squirrel on the hickory tree marks a currently optimal solution of an optimization problem, and this squirrel will not move until other squirrels find another hickory tree with better food source. Every flying squirrel individually searches for food and optimally utilizes the available food resources by exhibiting a dynamic foraging behavior.

In this case, squirrels on acorn trees denoted by $\boldsymbol{FS}_{at}$ may glide towards the hickory tree with predator presence probability P_{dp}, or glide towards randomly selected trees. The location of the squirrels on acorn trees updates using

$$\boldsymbol{FS}_{at}(k+1) = \begin{cases} \boldsymbol{FS}_{at}(k) + d_g G_c(\boldsymbol{FS}_{ht}(k) - \boldsymbol{FS}_{at}(k)), & R_1 \geqslant P_{dp} \\ \underline{\boldsymbol{FS}} + \boldsymbol{R}_{at} \odot (\overline{\boldsymbol{FS}} - \underline{\boldsymbol{FS}}), & \text{otherwise} \end{cases} \tag{14}$$

where k is the current iteration value. $\boldsymbol{R}_{at} \in \mathbb{R}^{3\times d}$ and R_1 contain uniformly distributed random numbers in range $[0,1]$. G_c is a constant set as 1.9, and d_g is the gliding distance of squirrels. According to Jain et al. (2019), d_g is approximately calculated using $d_g = \frac{h_g}{\tan(\phi)}$, where h_g is the loss in height occurred after gliding and $\phi = \arctan(\frac{C_D}{C_L})$ is the gliding angle to the horizontal plane. C_D is the drag coefficient fixed as 0.60. C_L is the lift coefficient and randomly chosen in the range $(0.675, 1.5)$, which can be calculated using $(1.5-0.675)r_d + 0.675$. Thus, d_g is approximately calculated as

$$d_g = \frac{8}{0.6sf}(0.825r_d + 0.675) = \frac{11r_d + 9}{sf} \tag{15}$$

where r_d is a uniformly distributed random scalar in range $[0,1]$. sf is called scaling factor which is a suitable non-zero constant in range $[16,37]$ in order to achieve the desired level of accuracy without affecting stability of algorithm.

When $R_1 \geqslant P_{dp}$, the movement of a squirrel located on an acorn tree $\boldsymbol{FS}_{at}$ is represented by a randomly scaled difference vector $d_g G_c(\boldsymbol{FS}_{ht}(k) - \boldsymbol{FS}_{at}(k))$ which is a new search direction towards and near to the global optimal solution. This movement enhances the exploitation phase of SSA as the squirrel searches for more accurate optimal solutions nearby the current optimal solution. When $R_1 < P_{dp}$, i.e. predator presence is felt by a squirrel, then it may move to some random safe location represented by a random point of the search space $\underline{\boldsymbol{FS}} + \boldsymbol{R}_{at} \odot (\overline{\boldsymbol{FS}} - \underline{\boldsymbol{FS}})$. This random movement of the squirrels allows SSA to more completely explore the search space and avoid getting stuck at local minimal points, which enhances the exploration phase SSA.

Similarly, the squirrels on normal trees denoted by $\boldsymbol{FS}_{nt}$ may glide to the hickory tree, the acorn trees or randomly selected trees. For each squirrel on the normal tree, it glides towards the selected tree with a probability P_{dp}. Otherwise, the squirrel glides towards a tree randomly selected in the search space. The location of the squirrels on normal trees updates using

$$\boldsymbol{FS}_{nt}(k+1) = \begin{cases} \boldsymbol{FS}_{nt}(k) + d_g G_c(\boldsymbol{FS}_{at}(k) - \boldsymbol{FS}_{nt}(k)), & R_4 \geqslant P_s \text{ and } R_2 \geqslant P_{dp} \\ \boldsymbol{FS}_{nt}(k) + d_g G_c(\boldsymbol{FS}_{ht}(k) - \boldsymbol{FS}_{nt}(k)), & R_4 < P_s \text{ and } R_3 \geqslant P_{dp} \\ \underline{\boldsymbol{FS}} + \boldsymbol{R}_{nt} \odot (\overline{\boldsymbol{FS}} - \underline{\boldsymbol{FS}}), & \text{otherwise} \end{cases} \tag{16}$$

where $\boldsymbol{R}_{nt} \in \mathbb{R}^{3\times d}$, R_2, R_3 and R_4 contain uniformly distributed random numbers in range $[0,1]$. P_s is the probability that squirrels select the hickory tree or the acorn trees.

When $R_4 \geqslant P_s$ and $R_2 \geqslant P_{dp}$, a squirrel located on a normal tree $\boldsymbol{FS}_{nt}$ glides to a new search direction represented by a randomly scaled difference vector $d_g G_c(\boldsymbol{FS}_{at}(k) - \boldsymbol{FS}_{nt}(k))$. When $R_4 < P_s$ and $R_3 \geqslant P_{dp}$, the squirrel glides to another new search direction represented by a randomly scaled difference vector $d_g G_c(\boldsymbol{FS}_{ht}(k) - \boldsymbol{FS}_{nt}(k))$. Similar to that in Eq. (14), these movements make squirrels search near the hickory tree, which enhance the exploitation phase of SSA. When $R_2 < P_{dp}$ or $R_3 < P_{dp}$, the squirrel glides to a random location, which is the same as that Eq. (14) and enhances the exploration phase of SSA.

When winter is coming, squirrels search for food source more aggressively than usual to avoid starving to death. The mark of winter is that seasonal constants $S_c^i (i = 1, 2, 3)$ and the minimal value of seasonal constants $S_{\min}$ satisfy the seasonal monitoring condition: $S_c^i < S_{\min}$. These constants update using (Jain et al., 2019)

$$S_c^i = \sqrt{\sum_{q=1}^{d} (\boldsymbol{FS}_{at,q}^i - \boldsymbol{FS}_{ht,q})^2} \tag{17}$$

$$S_{\min} = \frac{10^{-6}}{365^{\frac{2.5k}{g}}} \tag{18}$$

where k is the current iteration value. g is the maximum number of iterations which is a positive integer. The value $S_{\min}$ affects the exploration and exploitation capabilities of SSA. Larger value of $S_{\min}$ promotes exploration while smaller values of $S_{\min}$ enhance the exploitation capability of algorithm. In winter, all the squirrels on normal trees starve, and the location of them updates using

$$\boldsymbol{FS}_{nt}^{new} = \underline{\boldsymbol{FS}} + (\overline{\boldsymbol{FS}} - \underline{\boldsymbol{FS}}) \odot \text{Lévy}(n-4). \tag{19}$$

Lévy(x) is the function used to generate an x by d matrix containing random numbers that obey the Lévy distribution:

$$\text{Lévy}(x) = \frac{0.01 r_a(x)}{|r_b(x)|^{\frac{1}{\beta}}} \left(\frac{\Gamma(1+\beta) \sin\frac{\pi\beta}{2}}{\Gamma\left(\frac{1+\beta}{2}\right) 2^{\frac{\beta-1}{2}} \beta} \right)^{\frac{1}{\beta}} \tag{20}$$

where constant $\beta = 1.5$, and $r_a(x) \in \mathbb{R}^{x \times d}$ and $r_b(x) \in \mathbb{R}^{x \times d}$ are matrices of uniformly distributed random numbers in range $[0, 1]$. orange this behavior enhances the exploration capability of SSA. In fact, the relocation of such squirrels in Eq. (19) is to randomly generate solutions using the Lévy flight represented by Lévy$(n-4)$, which encourages better and more efficient search space exploration than the uniformly generated random solutions in Eqs. (14) and (16).

Remark 2. In the fault-tolerant TA problem formulation in Section 2, the fourth term in Eq. (7) is non-convex and the constraint Eq. (8) is nonlinear. Thus, the fault-tolerant TA problem is a non-convex optimization problem subject to a nonlinear constraint. SSA is an iterative numerical optimization algorithm that can better solve non-convex and nonlinear optimization problems compared with most of widely employed algorithms, such as PSO, GA and so on (Jain et al., 2019). Therefore, in the next subsection, we will combine SSA with a Delta potential well model and utilize the modified SSA to solve the fault-tolerant TA problem in Section 2.

3.2. Quantum behavior

In quantum time-space framework, the location $\boldsymbol{X} = [x, y, z]^{\mathrm{T}}$ of an object in 3-dimensional space is modeled by wave function $\Psi(\boldsymbol{X}, t)$. The object can appear anywhere at the same time. In the quantum space, the probability of the object appearing at a point is proportional to the strength module of the wave function on the point, that is

$$|\Psi(\boldsymbol{X}, t)|^2 \mathrm{d}x\mathrm{d}y\mathrm{d}z = Q\mathrm{d}x\mathrm{d}y\mathrm{d}z \tag{21}$$

satisfying

$$\int_{-\infty}^{+\infty} |\Psi(\boldsymbol{X}, t)|^2 \mathrm{d}x\mathrm{d}y\mathrm{d}z = \int_{-\infty}^{+\infty} Q\mathrm{d}x\mathrm{d}y\mathrm{d}z = 1 \tag{22}$$

where $Q\mathrm{d}x\mathrm{d}y\mathrm{d}z$ is the probability of the object appearing at the spot $\boldsymbol{X}$ at time t.

According to the quantum physics, the state function of an object is described as the time-dependent *Schrödinger equation*:

$$i\hbar \frac{\partial}{\partial t} \Psi(\boldsymbol{X}, t) = \left(-\frac{\hbar^2}{2m} \nabla^2 + V(\boldsymbol{X}) \right) \Psi(\boldsymbol{X}, t) \tag{23}$$

where $\hbar$ is Planck Constant, m is the mess of the object and $V(\boldsymbol{X})$ is the potential field of the object.

In most of the population-based optimization algorithms, their search agents are driven to move towards the global optimal point. In this paper, a Delta potential well model of SSA is constructed to guide the search behavior of the squirrel. For simplicity, we consider a squirrel in one-dimensional space firstly. The optimal point p is set as the center of the Delta potential well, which attracts the search agents. The location of each flying squirrel is formulated as the wave function $\Psi(x, t)$ to replace the original foraging behavior of squirrels with the quantum foraging behavior. The squirrel at point x in the quantum space is attracted by the potential well at point p, and the potential energy at p is

$$V(x) = -\gamma\delta(x - p) \tag{24}$$

where γ is the depth of the Delta potential well. Subsequently, by solving the simultaneous equations: Eqs. (23)–(24), we have the probability density function

$$Q(x - p) = |\Psi(x - p)|^2 = \frac{1}{L} \exp\left(-\frac{2|x - p|}{L} \right) \tag{25}$$

where L is the feature length of the Delta potential well. Using Monte Carlo Method to solve the equation below

$$\frac{1}{L} \exp\left(-\frac{2|x - p|}{L} \right) = \frac{1}{L} U \tag{26}$$

the achieved solution is

$$x = p \pm \frac{L}{2} \ln\frac{1}{U} \tag{27}$$

where U is a uniformly distributed random number in range $[0, 1]$ independently generated for each dimension. The solution represents the location of a search agent in the search space.

The variables p and L in Eq. (27) are time-varying, and the iteration of QSSA is regarded as the discrete time. Thus, Eq. (27) can be rewritten as

$$\begin{cases} x(k+1) = p(k) \pm \frac{L(k)}{2} \ln\frac{1}{U} \\ L(k) = 2\lambda(k)|p(k) - x(k)| \end{cases} \tag{28}$$

where k means the kth iteration of the algorithm. $\lambda(k)$ is used to scale the feature length $L(k)$ and updates using

$$\lambda(k) = \omega_1 + (\omega_2 - \omega_1)\frac{k}{g} \tag{29}$$

where ω_1 and ω_2 are fixed between $(0, 1)$ and g is the maximum iteration number.

Flying squirrels on different trees behave differently, so the potential well center affecting on them is generated independently. According to Eq. (14), squirrels on acorn nut trees move towards the hickory nut tree, thus $\boldsymbol{FS}_{ht}$ is the center of the potential well. The feature length $\boldsymbol{L}_{at} \in \mathbb{R}^{1 \times d}$ is related to the distance between $\boldsymbol{p}_{at} \in \mathbb{R}^{1 \times d}$ and $\boldsymbol{FS}_{at}$:

$$\boldsymbol{p}_{at}(k) = \boldsymbol{FS}_{ht}(k) \tag{30}$$

$$\boldsymbol{L}_{at}(k) = 2\lambda(k) \left| \boldsymbol{p}_{at}(k) - \boldsymbol{FS}_{at}(k) \right| \tag{31}$$

where $|\cdot|$ is an element-wise operator of taking the absolute value. The Eq. (14) is rewritten as

$$\boldsymbol{FS}_{at}(k+1) = \boldsymbol{p}_{at} \pm \lambda(k) \left| \boldsymbol{p}_{at}(k) - \boldsymbol{FS}_{at}(k) \right| \odot \ln\frac{1}{\boldsymbol{U}_{3 \times d}} \tag{32}$$

Similarly, squirrels on normal trees are approaching towards the hickory nut tree or the acorn nut trees. Therefore, the center of the potential well $\boldsymbol{p}_{nt}$ is calculated using

$$\begin{cases} \boldsymbol{p}_{nt}(k) = \phi(k)\frac{1}{3}\sum_{i=1}^{3} \boldsymbol{FS}_{at}^i(k) + (1 - \phi(k))\boldsymbol{FS}_{ht}(k) \\ \phi(k) = \frac{c_1 r_1(k)}{c_1 r_1(k) + c_2 r_2(k)} \end{cases} \tag{33}$$

where c_1 and c_2 are scalars fixed in range $[0, 1]$. r_1 and r_2 are random numbers in range $[0, 1]$, which are generated independently during each

iteration. Obviously, the feature length $\boldsymbol{L}_{nt}$ is related to the distance between $\boldsymbol{p}_{nt}$ and $\boldsymbol{FS}_{nt}$:

$$\boldsymbol{L}_{nt}(k) = 2\lambda(k)|\boldsymbol{p}_{nt}(k) - \boldsymbol{FS}_{nt}| \tag{34}$$

The Eq. (16) is rewritten as

$$\boldsymbol{FS}_{nt}(k+1) = \boldsymbol{p}_{nt} \pm \lambda(k)|\boldsymbol{p}_{nt}(k) - \boldsymbol{FS}_{nt}| \odot \ln \frac{1}{\boldsymbol{U}_{(n-4)\times d}} \tag{35}$$

QSSA is created by combining SSA and the Delta potential well model, and its pseudo-code is described as Algorithm 1. The stop criterion in Algorithm 1 is a permissible small threshold for the difference between the last two consecutive results. If the difference between the last two consecutive results is smaller than the threshold value, the iteration stops, and then the algorithm outputs the current optimal solution. The threshold value is fixed as 10^{-6} in the numerical experiments of Section 3.3.

Algorithm 1: Pseudo-code of QSSA

```
Set algorithm parameters: n, g, sf, Pdp, Ps, ω1, ω2, c1, and c2;
Generate initial location of n flying squirrels using Eq (13);
while stop criterion is FALSE do
    Evaluate the fitness of each squirrel according to its location;
    Determine the flying squirrels on hickory, acorn, and normal
      trees respectively;
    for each flying squirrel on acorn nut trees do
        Update location using Eq. (32);
    end
    for each flying squirrel on normal trees do
        Update location using Eq. (35);
    end
    Calculate seasonal constant Sc using Eq. (17);
    for each flying squirrel do
        if seasonal monitoring condition is TRUE then
            Randomly generate flying squirrels using Eq. (19);
        end
    end
end
Output the location of the squirrel on the hickory nut tree as the
  optimal solution;
```

3.3. Numerical experiments of QSSA

The QSSA is evaluated by carrying out a set of numerical experiments that is solving 26 commonly used benchmark functions (Jamil and Yang, 2013; Liang et al., 2013) formulated in Table 5 of Appendix. In Table 5, f^* denotes the known global minimum of the benchmark functions. The numerical experiment of each benchmark function runs 30 times independently, and results are provided in Table 6 and Fig. 14. In the Table 6, *Best* and *Worst* are respectively the best result and the worst result among all the achieved solutions. *Mean* and *SD* respectively refer to the average value and standard deviation of the best solution over the 30 runs. The Fig. 14 shows the performance comparison between QSSA and SSA for 26 different benchmark functions, where the results of QSSA are plotted using solid lines and those of SSA dashed lines. The numerical experiments verify that QSSA beats SSA almost in solving all the benchmark functions with remarkable convergence rate.

4. Simulation and discussion

This paper chooses a 1:30 scale model of a platform supply ship for simulation. The model supply ship are mounted with 6 thrusters: 2 tunnel thrusters and 4 azimuth thrusters. The thruster layout in the horizontal plane and notations are shown in Fig. 2. The figure marks the installing positions of each thruster and the gravity center O of the supply ship. The installing positions of the thrusters are detailed in Table 1, and the magnitude and rate constraints of the thrusters during the operation of the DP system are provided in Table 2.

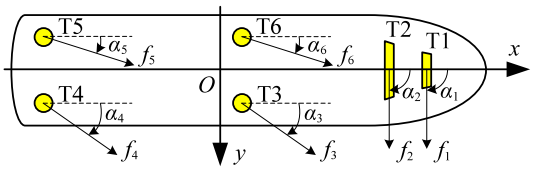


Fig. 2. Layout of thrusters and notations.

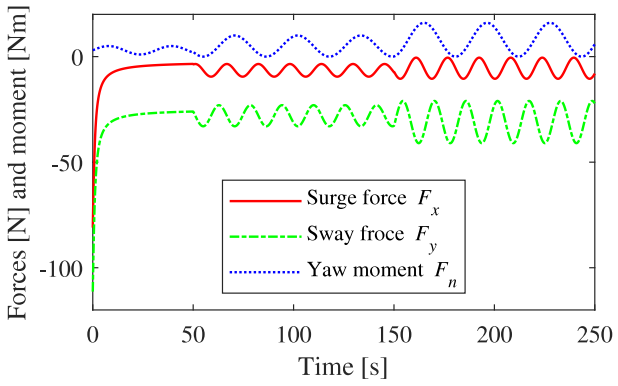


Fig. 3. Commanded generalized control force.

Table 1
Installing position of thrusters.

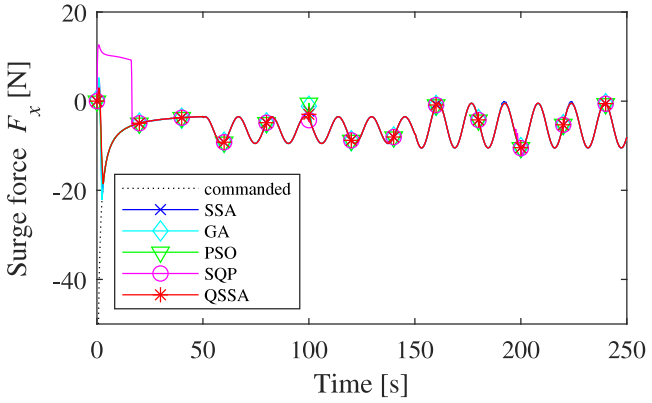
Index	Thruster type	Installing position	
		l_x [m]	l_y [m]
T1	Tunnel thruster	1.37	0
T2	Tunnel thruster	1.25	0
T3	Azimuth thruster	0.5	0.18
T4	Azimuth thruster	−1.68	0.18
T5	Azimuth thruster	−1.68	−0.18
T6	Azimuth thruster	0.5	−0.18

Table 2
Constraints of thrusters.

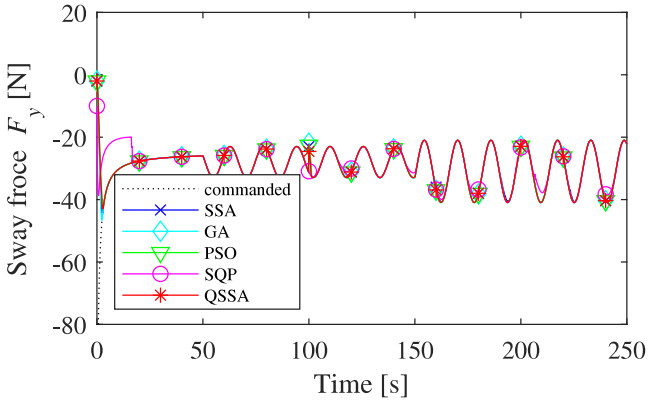
Index	$\underline{f}$ [N]	$\overline{f}$ [N]	Δf [N]	$\underline{\alpha}$ [deg]	$\overline{\alpha}$ [deg]	$\Delta\alpha$ [deg]
T1	−14.72	14.72	5	–	–	–
T2	−14.72	14.72	5	–	–	–
T3	5×10^{-5}	49.05	10	−252.60	72.60	8
T4	5×10^{-5}	49.05	10	−252.60	72.60	8
T5	5×10^{-5}	49.05	10	−72.60	252.60	8
T6	5×10^{-5}	49.05	10	−72.60	252.60	8

Table 3
Parameter selection of QSSA, SSA and PSO.

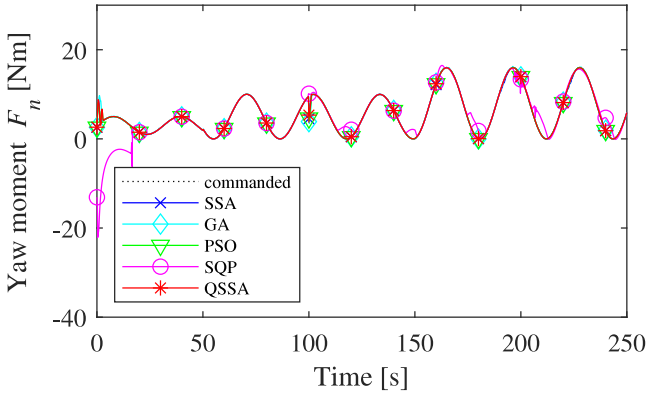
Parameter	QSSA	SSA	PSO
n	30	30	50
g	100	200	200
ω_1, ω_2	0.6,0.9	–	–
c_1, c_2	0.6,0.9	–	–
P_{dp}, P_s	0.2,0.5	0.2,0.5	–
sf	32.8	17.6	–



(a) Achieved surge force.

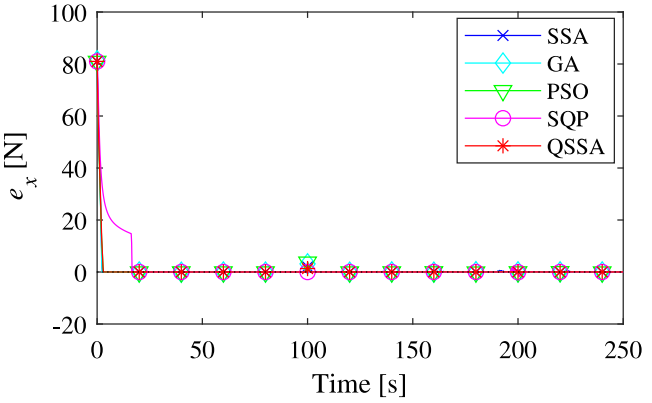


(b) Achieved sway force.

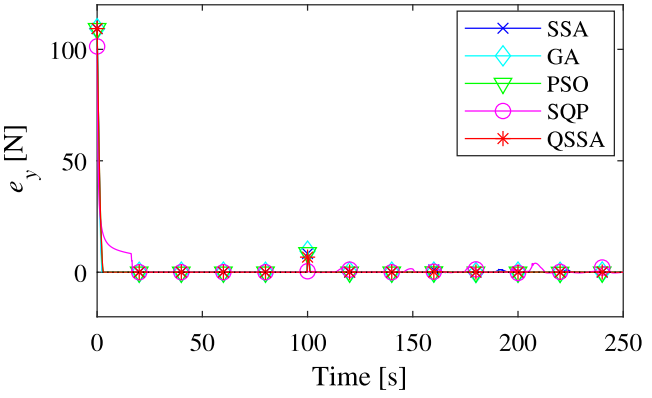


(c) Achieved yaw moment.

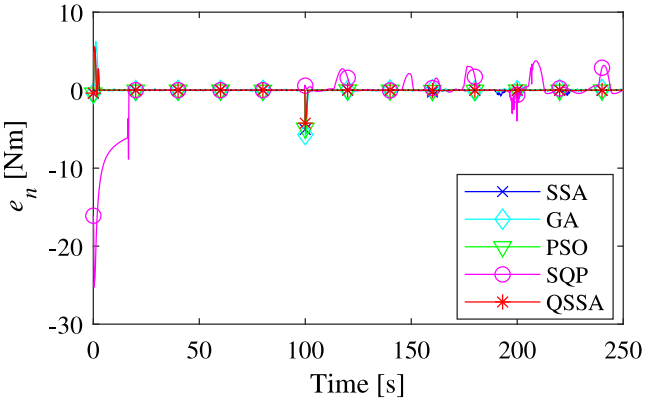
Fig. 4. Achieved generalized control force.



(a) Errors of surge force.



(b) Errors of sway force.



(c) Errors of yaw moment.

Fig. 5. Errors between commanded and achieved generalized control forces.

Table 4
Average values of the KPIs of 30 runs of simulations.

Algorithms	Allocation error J_e	Power consumption J_p	Tear-and-wear J_α
QSSA	**1.3256E+03**	**1.4315E+05**	**4.6578E+03**
SSA	1.3473E+03	1.5645E+05	8.9558E+03
PSO	1.3871E+03	1.5929E+05	1.8899E+04
GA	1.3299E+03	1.7064E+05	2.0719E+04
SQP	4.9348E+03	2.3632E+05	5.9303E+03

The simulation assumes the model ship is operating at the commanded generalized forces that can overcome certain external disturbances. And the TA task during the DP operation is to map the commanded generalized control force $\boldsymbol{\tau}_c$ into thruster forces. The commanded generalized control force $\boldsymbol{\tau}_c = [F_x, F_y, F_n]^{\mathrm{T}}$ commanded by the DP controller is assumed known and given as follows:

$$F_x = \begin{cases} 50\arctan t - 81, & 0 \leqslant t \leqslant 50 \\ 3\sin(0.4t) - 6.5, & 50 < t \leqslant 150 \\ 5\sin(0.4t) - 5.5, & 150 < t \leqslant 250 \end{cases} \tag{36}$$

$$F_y = \begin{cases} 55\arctan t - 111.31, & 0 \leqslant t \leqslant 50 \\ 5\cos(0.4t) - 28, & 50 < t \leqslant 150 \\ 10\cos(0.4t + \frac{\pi}{3}) - 31, & 150 < t \leqslant 250 \end{cases} \tag{37}$$

Table 5
Function benchmarks for the numerical experiments.

Benchmark functions	d	Constraints on x_j	Global minimum f^*
$f_1 = \sum_{j=1}^{d}(x_j + 0.5)^2$	30	$[-5.12, 5.12]$	0
$f_2 = \sum_{j=1}^{n} x_j^2$	30	$[-100, 100]$	0
$f_3 = \sum_{j=1}^{d} j x_j^2$	30	$[-10, 10]$	0
$f_4 = \sum_{j=1}^{d} x_j^4 + \text{rand}[0, 1)$	30	$[-1.28, 1.28]$	0
$f_5 = (1.5 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_2^2)^2 + (2.625 - x_1 + x_1 x_2^3)^2$	2	$[-4.5, 4.5]$	0
$f_6 = -\cos x_1 \cos x_2 \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2)$	2	$[-100, 100]$	-1
$f_7 = 0.26(x_1^2 + x_2^2) - 0.48 x_1 x_2$	2	$[-10, 10]$	0
$f_8 = 100(x_1^2 - x_2^2) + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1(x_2 - 1)^2 + (x_4 - 1)^2 + 19.8(x_2 - 1)(x_4 - 1)$	4	$[-10, 10]$	0
$f_9 = \sum_{j=1}^{d} x_j^2 + (\sum_{j=1}^{d} 0.5 j x_j)^2 + (\sum_{j=1}^{d} 0.5 j x_j)^4$	10	$[-5, 10]$	0
$f_{10} = \sum_{j=1}^{d} \lvert x_j \rvert + \prod_{j=1}^{d} \lvert x_j \rvert$	30	$[-10, 10]$	0
$f_{11} = \sum_{j=1}^{d} (\sum_{k=1}^{j} x_k)$	30	$[-100, 100]$	0
$f_{12} = (x_1 - 1)^2 + \sum_{j=2}^{d} j(2x_j^2 - x_j - 1)^2$	30	$[-10, 10]$	0
$f_{13} = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$	2	$[-100, 100]$	0
$f_{14} = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	2	$[-10, 10]$	0
$f_{15} = -\sum_{j=1}^{d} \sin x_j \sin^{20}\frac{j x_j}{\pi}$	2	$[-0, \pi]$	-1.8013
$f_{16} = -\sum_{j=1}^{d} \sin x_j \sin^{20}\frac{j x_j}{\pi}$	5	$[-0, \pi]$	-4.6877
$f_{17} = -\sum_{j=1}^{d} \sin x_j \sin^{20}\frac{j x_j}{\pi}$	10	$[-0, \pi]$	-9.6602
$f_{18} = \sum_{j=1}^{d} \left(x_j^2 - 10\cos(2\pi x_j) + 10\right)$	30	$[-5.12, 5.12]$	0
$f_{19} = 0.5 + \frac{\sin^2\sqrt{x_1^2 + x_2^2} - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$	2	$[-100, 100]$	0
$f_{20} = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]$	-1.03163
$f_{21} = x_1^2 + 2x_2^2 + 0.3\cos(3\pi x_1)\cos(4\pi x_2) + 0.3$	2	$[-100, 100]$	0
$f_{22} = x_1^2 + 2x_2^2 + 0.3\cos(3\pi x_1 + 4\pi x_2) + 0.3$	2	$[-100, 100]$	0
$f_{23} = \prod_{j=1}^{d} \left(\sum_{k=1}^{5} k\cos((k+1)x_1 + 1)^2\right)^2$	2	$[-10, 10]$	-186.7309
$f_{24} = \sum_{j=1}^{d} \left(100(x_{j+1} - x_j^2)^2 + (x_j - 1)^2\right)$	30	$[-30, 30]$	0
$f_{25} = \frac{1}{4000}\left(\sum_{j=1}^{d}(x_j^2 - 100)^2\right) - \left(\prod_{j=1}^{d}\cos\frac{x_j - 100}{\sqrt{j}}\right) + 1$	30	$[-600, 600]$	0
$f_{26} = -20\exp\left(0.2\sqrt{\frac{1}{d}\sum_{j=1}^{d} x_j^2}\right) - \exp\left(-\frac{1}{d}\sum_{j=1}^{d}\cos(2\pi x_j)\right) + 20 + \exp(1)$	30	$[-32, 32]$	0

$$F_n = \begin{cases} 2\sin(0.2t) + 3, & 0 \leqslant t \leqslant 50 \\ 5\sin(0.2t) + 5, & 50 < t \leqslant 150 \\ 8\sin(0.2t) + 8, & 150 < t \leqslant 250 \end{cases} \tag{38}$$

where t is time. As for the model ship, the sampling time of the DP control system, including the TA procedure, is 0.20 [s]. Also, the plots of the commanded surge force F_x, sway force F_y and yaw moment F_n are shown in Fig. 3.

Thruster LOE faults occur in three different thrusters at different time. Firstly, at $t = 100$ [s], the efficiency of thruster T1 drops down to 30%, i.e. efficiency factor $\eta_1 = 0.3$; and thruster T3 suffers from complete LOE faults, i.e. efficiency factor $\eta_3 = 0$. Then, at $t = 200$ [s], the efficiency of thruster T6 becomes 70%, i.e. efficiency factor $\eta_6 = 0.7$.

The weights in the objective function Eq. (7) of the constrained fault-tolerant TA optimization model provided in Section 2 are set as follows: $\boldsymbol{W}_s = \text{diag}(1000, 1000, 1000)$, $\boldsymbol{w}_f = [10, 10, 10]^{\mathrm{T}}$, $\boldsymbol{W}_\alpha = \text{diag}(10, 10, 10)$ and $w_n = 100$. For all thrusters, initial conditions of the thrust force and the azimuth angle are set to $[0, 0, 0, 0, 0, 0]^{\mathrm{T}}$ [N] and $[90, 90, 0, 0, 0, 0]^{\mathrm{T}}$ [deg] respectively. The magnitude and rate constraints of the thrusters are given in Table 2.

In order to evaluate the performance of QSSA for solving the constrained fault-tolerant TA optimization problem, this section provides the simulation result of QSSA as well as other popular approaches as comparison, including SQP (Johansen et al., 2004), PSO (Wang et al., 2013) and GA (Mauro and Nabergoj, 2016). The parameters of QSSA, SSA and PSO are properly selected and listed in Table 3, where the other parameters of PSO except n and g are chosen the same as those in Wang et al. (2013). SQP and GA are both implemented using the MATLAB optimization toolbox. The simulation results include the achieved generalized control force denoted by $\boldsymbol{\tau}^* = [F_x^*, F_y^*, F_n^*]^{\mathrm{T}}$, which is shown in Fig. 4. The relative errors between the achieved and the commanded generalized control force are recorded and plotted in Fig. 5, which are denoted by e_y, e_x and e_n.

$$\begin{cases} e_x = F_x^* - F_x \\ e_y = F_y^* - F_y \\ e_n = F_n^* - F_n \end{cases} \tag{39}$$

Furthermore, the allocated thrust force and azimuth angle of each thruster during 250 [s] are plotted in Figs. 6–11.

To obtain a quantitative measure of the performance of the involved algorithms, 3 key performance indicators (KPIs) are developed in Eq. (40) based on the allocation errors, the power consumption and the thruster tear-and-wear respectively.

$$\begin{cases} J_e = \int_{t=0}^{250} \{\lvert e_x \rvert + \lvert e_y \rvert + \lvert e_n \rvert\} \mathrm{d}t \\ J_p = \int_{t=0}^{250} \{\sum_{i=1}^{m} \lvert f_i \rvert^{\frac{3}{2}}\} \mathrm{d}t \\ J_\alpha = \int_{t=0}^{250} \{\sum_{i=1}^{m} \lvert \alpha_i - \alpha_{0,i} \rvert\} \mathrm{d}t \end{cases} \tag{40}$$

Table 6
Numerical experiment results of QSSA and SSA for 26 benchmark functions.

		SSA	QSSA			SSA	QSSA
f_1	Best	1.2265E−22	**0**	f_{14}	Best	**0**	**0**
	Worst	1.1541E−21	**0**		Worst	**0**	**0**
	Mean	3.6701E−22	**0**		Mean	**0**	**0**
	SD	2.1142E−22	**0**		SD	**0**	**0**
f_2	Best	2.0799E−22	**1.2197E−88**	f_{15}	Best	**−1.8013**	**−1.8013**
	Worst	3.3837E−21	**1.9978E−76**		Worst	**−1.8013**	**−1.8013**
	Mean	7.6632E−22	**7.0822E−78**		Mean	**−1.8013**	**−1.8013**
	SD	5.6678E−22	**3.6463E−77**		SD	**9.0336E−16**	**9.0336E−16**
f_3	Best	2.3446E−10	**6.3755E−92**	f_{16}	Best	−4.6877	**−4.8935**
	Worst	6.3152E−07	**7.5002E−79**		Worst	−3.4950	**−4.6459**
	Mean	1.2101E−07	**2.6117E−80**		Mean	−4.3203	**−4.7314**
	SD	1.3799E−07	**1.3686E−79**		SD	3.4487E−01	**7.5150E−02**
f_4	Best	6.1611E−02	**3.0128E−02**	f_{17}	Best	−6.1330	**−9.7849**
	Worst	3.5709E−01	**1.6839E−01**		Worst	−4.0445	**−7.9213**
	Mean	1.3931E−01	**7.6938E−02**		Mean	−5.1515	**−9.2660**
	SD	5.5861E−02	**3.3238E−02**		SD	**5.4298E−01**	4.2480E−01
f_5	Best	**0**	**0**	f_{18}	Best	1.4735E+01	**9.9683E−01**
	Worst	**0**	**0**		Worst	5.6243E+01	**1.3929E+01**
	Mean	**0**	**0**		Mean	3.1248E+01	**6.6337E+00**
	SD	**0**	**0**		SD	1.1735E+01	**2.9249E+00**
f_6	Best	−1	−1	f_{19}	Best	**0**	**0**
	Worst	−1	−1		Worst	**0**	**0**
	Mean	−1	−1		Mean	**0**	**0**
	SD	**0**	**0**		SD	**0**	**0**
f_7	Best	1.4938E−68	**1.0565E−176**	f_{20}	Best	**−1.0316**	**−1.0316**
	Worst	4.1670E−53	**5.5170E−152**		Worst	**−1.0316**	**−1.0316**
	Mean	2.5048E−54	**3.4462E−153**		Mean	**−1.0316**	**−1.0316**
	SD	8.9732E−54	**1.0941E−152**		SD	6.7752E−16	**6.6486E−16**
f_8	Best	3.9582E−22	**0**	f_{21}	Best	**0**	**0**
	Worst	3.8878E+00	**0**		Worst	**0**	**0**
	Mean	1.1663E+00	**0**		Mean	**0**	**0**
	SD	1.8120E+00	**0**		SD	**0**	**0**
f_9	Best	8.7391E−23	**7.5308E−40**	f_{22}	Best	**0**	**0**
	Worst	8.4877E−21	**1.0171E−23**		Worst	**0**	**0**
	Mean	1.3360E−21	**3.8038E−25**		Mean	**0**	**0**
	SD	1.8447E−21	**1.8606E−24**		SD	**0**	**0**
f_{10}	Best	2.1992E−01	**6.0514E−50**	f_{23}	Best	**−186.7309**	**−186.7309**
	Worst	7.7436E−01	**1.3530E−45**		Worst	**−186.7309**	**−186.7309**
	Mean	4.2491E−01	**1.0881E−46**		Mean	**−186.7309**	**−186.7309**
	SD	1.5806E−01	**2.8113E−46**		SD	1.0995E−13	**3.6944E−14**
f_{11}	Best	**2.3780E−07**	3.4753E+00	f_{24}	Best	1.6053E+01	**4.5007E−12**
	Worst	**5.3959E−06**	9.4390E+01		Worst	2.8689E+01	**6.4667E+00**
	Mean	**1.4954E−06**	3.6338E+01		Mean	2.4621E+01	**1.4371E+00**
	SD	**1.2287E−06**	2.5998E+01		SD	2.5170E+00	**2.1258E+00**
f_{12}	Best	4.1687E−04	**9.1025E−21**	f_{25}	Best	1.5543E−15	**0**
	Worst	3.0573E+00	**9.7514E−19**		Worst	7.1125E−02	**3.9202E−02**
	Mean	5.2071E−01	**2.5555E−19**		Mean	1.5989E−02	**1.4921E−02**
	SD	7.3975E−01	**1.7554E−19**		SD	1.6427E−02	**1.2051E−02**
f_{13}	Best	**0**	**0**	f_{26}	Best	3.5741E+00	**1.5099E−14**
	Worst	**0**	**0**		Worst	8.3668E+00	**3.9968E−14**
	Mean	**0**	**0**		Mean	5.4369E+00	**2.5757E−14**
	SD	**0**	**0**		SD	1.1629E+00	**6.3962E−15**

The smaller the values of these KPIs are, the better the related performance of the algorithms is. After 30 runs of simulations, the average values of the KPIs are obtained in Table 4.

Remark 3. When solving the constrained fault-tolerant TA optimization problem, the adopted iterative optimization algorithms (QSSA, SSA, SQP, PSO and GA) usually require quite a lot of iterations to attain the global optimal solution, which consumes too much time for online computation. And the execution time of the TA algorithm is not allowed to exceed the sampling interval of the ship DP control system. Hence, it is necessary to reduce the maximum iteration number of the TA algorithm so as to meet the real-time requirement, however, at the cost of attaining the suboptimal solution with lower accuracy. The parameters of all these algorithms are selected by balancing execution time and accuracy.

Observing the generalized control forces achieved by different TA algorithms in Fig. 4, it is shown that all of the algorithms can effectively solve the constrained fault-tolerant TA problem with similar performance. However, the allocation errors shown in Fig. 5 and the KPI J_e in Table 4 tell difference among the discussed algorithms. The average value of the KPI J_e of QSSA is the smallest, which means the accuracy of QSSA is the highest. Therefore, QSSA can better solve the constrained fault-tolerant TA problem than SSA GA, PSO and SQP. Moreover, the allocation error of SQP, represented as circle marked line in Fig. 5, become the largest among all algorithms after thruster LOE faults occur at $t = 100$ [s]. And after another thruster LOE fault occurs at $t = 200$ [s], the allocation errors become much larger. This shows that the SQP cannot find proper optimal solutions when facing the thruster LOE fault, while the other intelligent algorithm can, including QSSA.

From Figs. 6–12 and Table 4, it is observed that the power consumption and tear-and-wear of thrusters by using QSSA are both the lowest among those of all discussed algorithms. This means energy consumption of the thruster system will be decreased, and the lifetime of thrusters will be extended. Therefore, economic benefits are increased by utilizing QSSA for the constrained fault-tolerant TA problem.

Remark 4. As to the scale model ship, the sampling interval of the DP control system is 0.2 [s], which leads that the TA algorithm has to

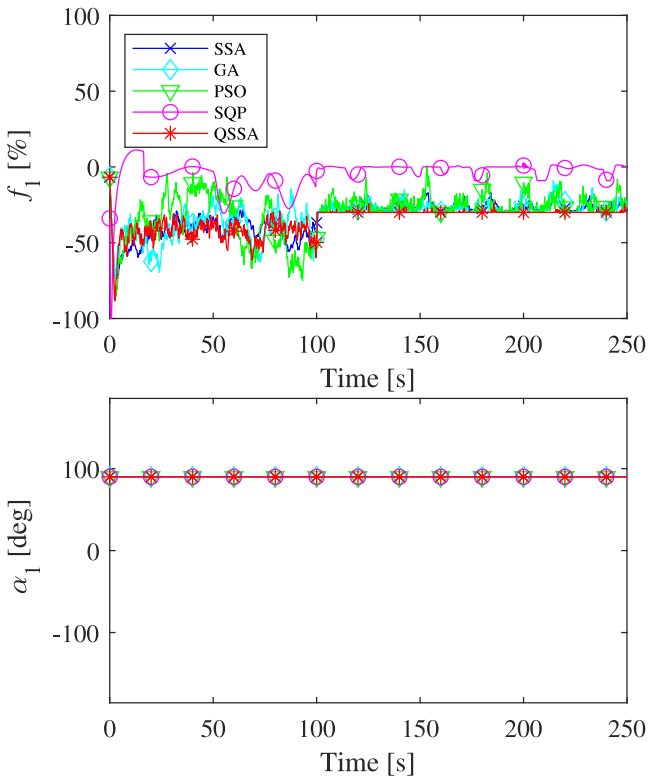


Fig. 6. Thrust force and azimuth angle (fixed at 90 [deg]) of thruster T1, whose efficiency becomes 30% at $t = 100$ [s].

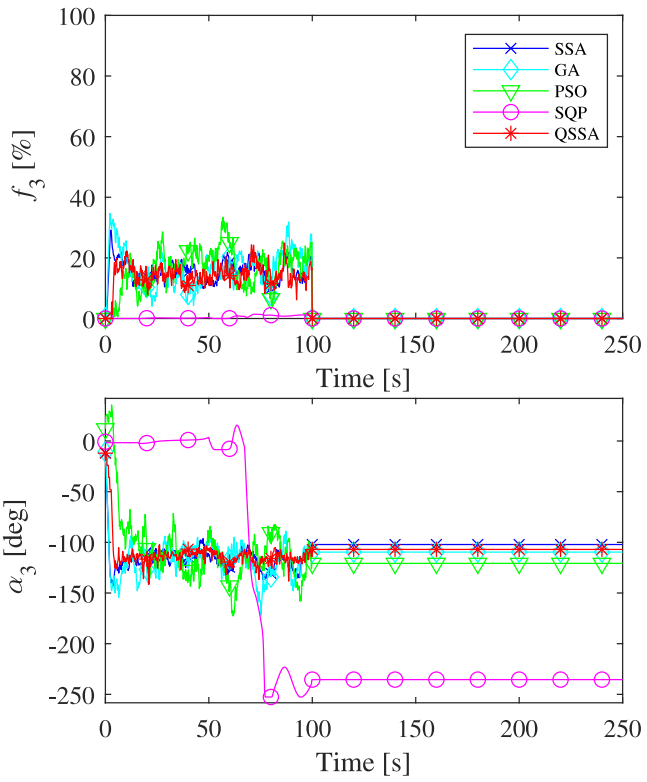


Fig. 8. Thrust force and azimuth angle of thruster T3 suffering from complete LOE fault from $t = 100$ [s].

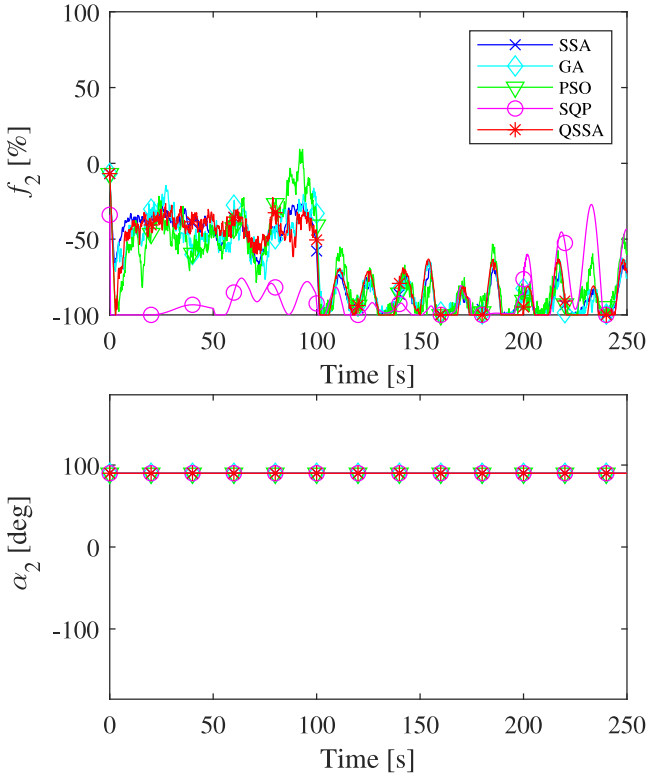


Fig. 7. Thrust force and azimuth angle (fixed at 90 [deg]) of thruster T2.

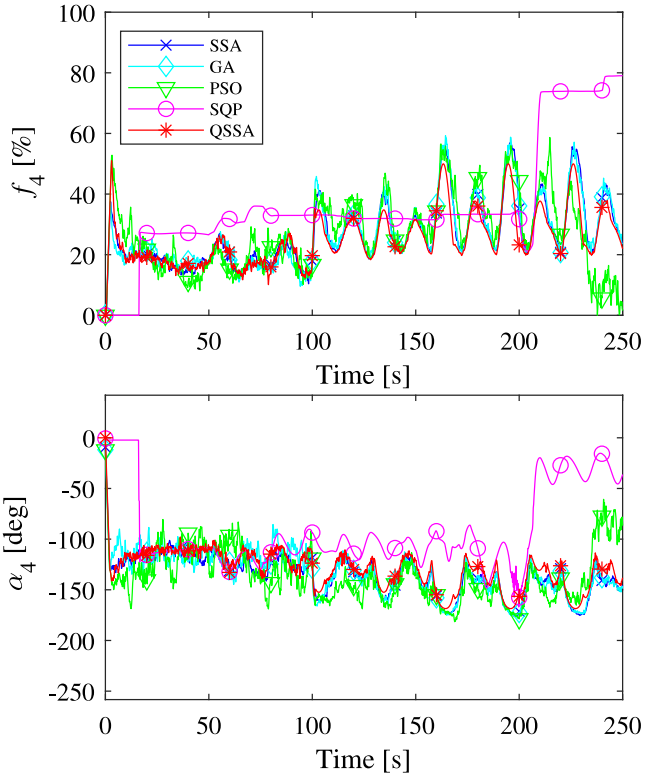


Fig. 9. Thrust force and azimuth angle of thruster T4.

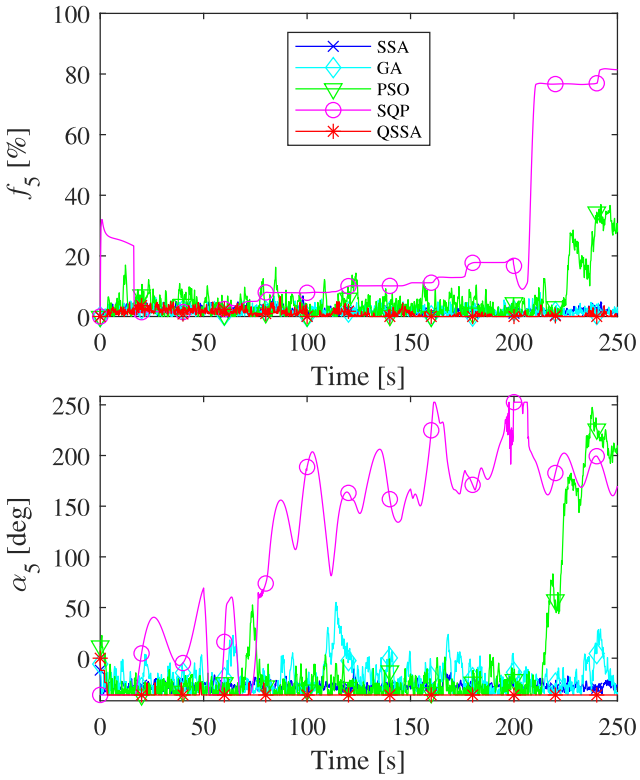


Fig. 10. Thrust force and azimuth angle of thruster T5.

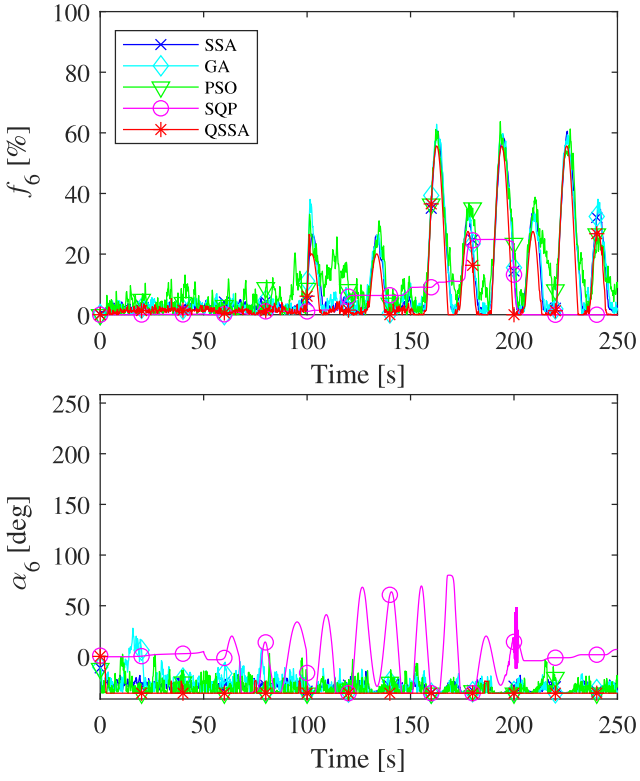


Fig. 11. Thrust force and azimuth angle of thruster T6, whose efficiency becomes 70% at $t = 200$ [s].

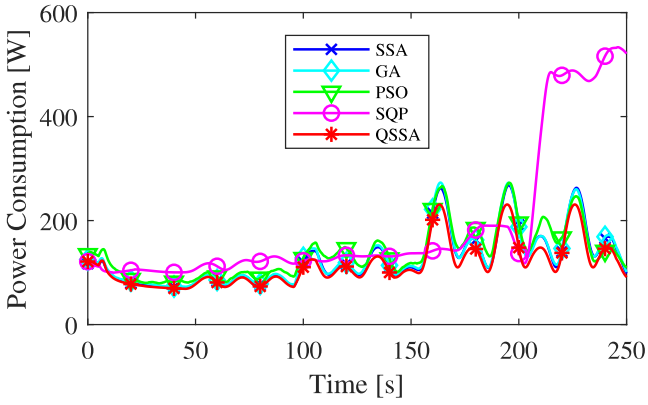


Fig. 12. Total power consumption using SSA, GA, PSO, SQP and QSSA respectively.

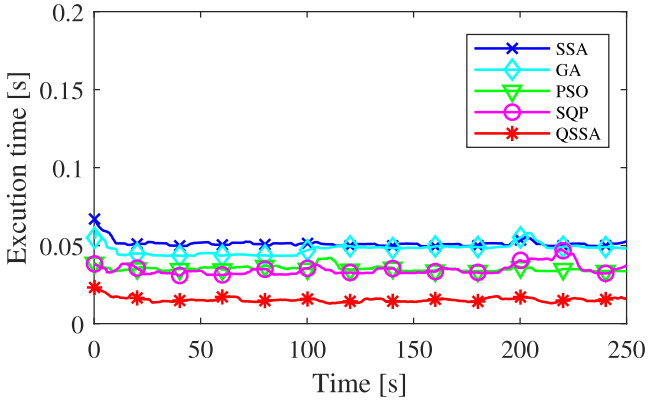


Fig. 13. Recorded execution time during each sampling interval of the ship DP control system (every 0.2 [s]).

calculate a feasible solution within 0.2 [s] so as to meet the real-time requirement. The simulation is performed using the MATLAB software on a machine with CPU at 2.3 GHz, 4.0 GB RAM and the 64-bit Windows 10 operating system. The execution time of the TA procedures using SSA, GA, PSO, SQP and QSSA during every sampling interval is recorded and given in Fig. 13, which is 0.05129 [s], 0.04775 [s], 0.03519 [s], 0.03470 [s] and 0.03318 [s] on average respectively. It turns out that the algorithms all satisfy the real-time requirement in the laboratory condition; and the advantages of QSSA is its lower average execution time and better performance compared with SSA, GA, PSO and SQP.

5. Conclusion

This paper has proposed the novel QSSA for the TA optimization problem subject to LOE faults as well as magnitude and rate constraints of thrusters. The numerical experiment of solving 26 different benchmark functions has indicated that QSSA obtains remarkable convergence rate by utilizing the Delta potential well model to guide the search behavior of the squirrel. The simulation on the scale model supply ship has verified that QSSA can provide more accurate and reliable solutions for the constrained fault-tolerant TA optimization problem than SSA, GA, PSO and SQP while ensuring the safety of the ship under thruster LOE faults. The developed QSSA can be found effective for solving the optimization problem of other engineering fields.

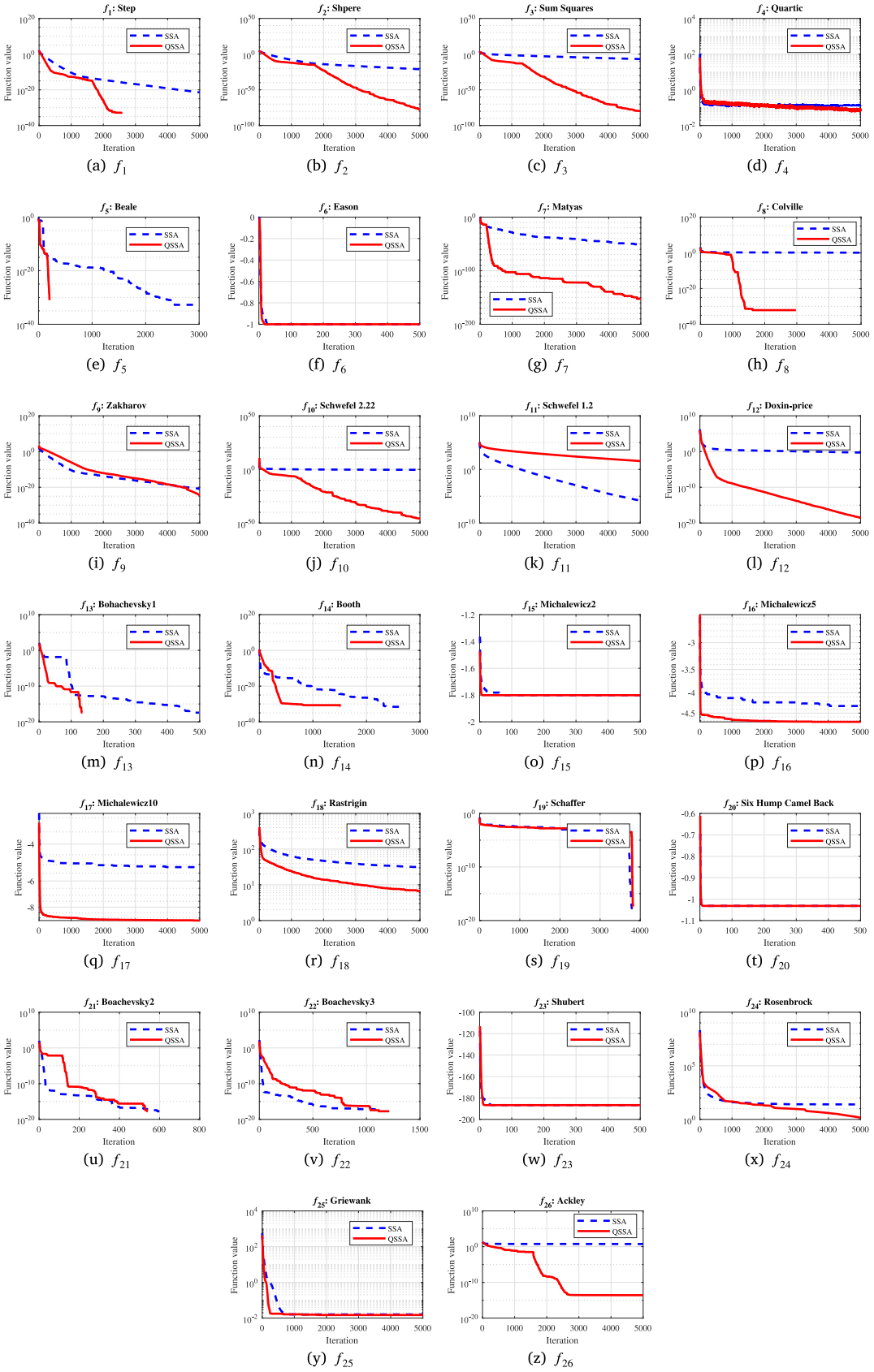


Fig. 14. Convergence comparison between QSSA and SSA in solving the benchmark functions..

CRediT authorship contribution statement

Zhihao Yu: Methodology, Software, Investigation, Visualization, Writing – original draft, Writing – review & editing. **Jialu Du:** Conceptualization, Supervision, Writing – review & editing, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Appendix. Benchmark formulation and results of numerical experiments.

See Tables 5, 6 and Fig. 14.

References

Alvarez-Alvarado, M.S., Alban-Chacon, F.E., Lamilla-Rubio, E.A., Rodriguez-Gallegos, C.D., Velasquez, W., 2021. Three novel quantum-inspired swarm optimization algorithms using different bounded potential fields. Sci. Rep. 11 (1), 11655.

Alwi, H., Edwards, C., 2008. Fault tolerant control using sliding modes with on-line control allocation. Automatica 44 (7), 1859–1866.

Arditti, F., Cozijn, H., Van Daalen, E., Tannuri, E.A., 2019. Robust thrust allocation algorithm considering hydrodynamic interactions and actuator physical limitations. J. Mar. Sci. Technol. 24 (4), 1057–1070.

Arditti, F., Souza, F.L., Martins, T.C., Tannuri, E.A., 2015. Thrust allocation algorithm with efficiency function dependent on the azimuth angle of the actuators. Ocean Eng. 105, 206–216.

Argha, A., Su, S.W., Celler, B.G., 2019. Control allocation-based fault tolerant control. Automatica 103, 408–417.

Basu, M., 2019. Squirrel search algorithm for multi-region combined heat and power economic dispatch incorporating renewable energy sources. Energy 182, 296–305.

Cao, H., Zheng, H., Hu, G., 2021. The optimal multi-degree reduction of Ball Bézier curves using an improved squirrel search algorithm. Eng. Comput. 1–24.

Casavola, A., Garone, E., 2010. Fault-tolerant adaptive control allocation schemes for overactuated systems. Internat. J. Robust Nonlinear Control 20 (17), 1958–1980.

Cozijn, J.L., Hallmann, R., 2012. The wake flow behind azimuthing thrusters: Measurements in open water, under a plate and under a barge. In: ASME 2012 31st International Conference on Ocean, Offshore and Arctic Engineering, Volume 1: Offshore Technology. pp. 485–494.

Cozijn, J.L., Hallmann, R., 2013. Thruster-interaction effects on a DP semi-submersible and a drill ship: Measurement and analysis of the thruster wake flow. In: International Conference on Offshore Mechanics and Arctic Engineering, Volume 1: Offshore Technology. V001T01A060.

Cristofaro, A., Johansen, T.A., 2014. Fault tolerant control allocation using unknown input observers. Automatica 50 (7), 1891–1897.

Fossen, T.I., 1994. Guidance and Control of Ocean Vehicles. John Wiley & Sons. Ltd..

Fossen, T.I., 2011. Handbook of Marine Craft Hydrodynamics and Motion Control. John Wiley & Sons. Ltd..

Fossen, T.I., Sagatun, S.I., 1991. Adaptive control of nonlinear systems: A case study of underwater robotic systems. J. Robot. Syst. 8 (3), 393–412.

Haibin, D., Cong, L., 2015. Quantum-behaved brain storm optimization approach to solving loney's solenoid problem. IEEE Trans. Magn. 51 (1), 1–7.

Hannan, M.A., Ali, J.A., Mohamed, A., Amirulddin, U.A.U., Tan, N.M.L., Uddin, M.N., 2018. Quantum-behaved lightning search algorithm to improve indirect field-oriented fuzzy-PI control for IM drive. IEEE Trans. Ind. Appl. 54 (4), 3793–3805.

Hao, G., Wenbo, X., Jun, S., Yulan, T., 2010. Multilevel thresholding for image segmentation through an improved quantum-behaved particle swarm algorithm. IEEE Trans. Instrum. Meas. 59 (4), 934–946.

Jain, M., Singh, V., Rani, A., 2019. A novel nature-inspired algorithm for optimization: Squirrel search algorithm. Swarm Evol. Comput. 44, 148–175.

Jamil, M., Yang, X.S., 2013. A literature survey of benchmark functions for global optimisation problems. Int. J. Math. Model. Numer. Optim. 4 (2), 150–194.

Johansen, T.A., 2004. Optimizing nonlinear control allocation. In: 2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601), Vol. 4. pp. 3435–3440.

Johansen, T.A., Fossen, T.I., 2013. Control allocation—A survey. Automatica 49 (5), 1087–1103.

Johansen, T.A., Fossen, T.I., Berge, S.P., 2004. Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming. IEEE Trans. Control Syst. Technol. 12 (1), 211–216.

Johansen, T.A., Fossen, T.I., Tøndel, P., 2005. Efficient optimal constrained control allocation via multiparametric programming. J. Guid. Control Dyn. 28 (3), 506–515.

Johansen, T.A., Fuglseth, T.P., Tøndel, P., Fossen, T.I., 2008. Optimal constrained control allocation in marine surface vessels with rudders. Control Eng. Pract. 16 (4), 457–464.

Khare, A., Agrawal, S., 2019. Scheduling hybrid flowshop with sequence-dependent setup times and due windows to minimize total weighted earliness and tardiness. Comput. Ind. Eng. 135, 780–792.

Li, X., Du, J., Li, G., 2016. Thrust allocation optimization in dynamic positioning vessels with main propeller-rudders. J. Donghua Univ. (Engl. Ed.).

Li, Y., Xiang, R., Jiao, L., Liu, R., 2012. An improved cooperative quantum-behaved particle swarm optimization. Soft Comput. 16 (6), 1061–1069.

Li, X.B., Yang, L.C., 2019. Study of constrained nonlinear thrust allocation in ship application based on optimization and SOM. Ocean Eng. 191, 106491.

Liang, J.J., Qu, B.Y., Suganthan, P.N., 2013. Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization, Vol. 635. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore, p. 490.

Lindegaard, K.P., Fossen, T.I., 2003. Fuel-efficient rudder and propeller control allocation for marine craft: experiments with a model ship. IEEE Trans. Control Syst. Technol. 11 (6), 850–862.

Liu, X., Abbas, M., Hu, G., BiBi, S., 2021. Degree reduction of Q-Bézier curves via squirrel search algorithm. Mathematics 9 (18).

Mahdi, F.P., Vasant, P., Abdullah-Al-Wadud, M., Kallimani, V., Watada, J., 2019. Quantum-behaved bat algorithm for many-objective combined economic emission dispatch problem using cubic criterion function. Neural Comput. Appl. 31 (10), 5857–5869.

Mauro, F., Nabergoj, R., 2016. Advantages and disadvantages of thruster allocation procedures in preliminary dynamic positioning predictions. Ocean Eng. 123, 96–102.

Narayanan, A., Moore, M., 1996. Quantum-inspired genetic algorithms. In: Proceedings of IEEE International Conference on Evolutionary Computation. pp. 61–66.

Omerdic, E., Roberts, G., 2004. Thruster fault diagnosis and accommodation for open-frame underwater vehicles. Control Eng. Pract. 12 (12), 1575–1598.

Ruth, E., 2008. Propulsion Control and Thrust Allocation on Marine Vessels (Thesis). Norwegian University of Science and Technology.

Ruth, E., Smogeli, O.N., Perez, T., Sorensen, A.J., 2009. Antispin thrust allocation for marine vessels. IEEE Trans. Control Syst. Technol. 17 (6), 1257–1269.

Ruth, E., Sørensen, A.J., Perez, T., 2007. Thrust allocation with linear constrained quadratic cost function. IFAC Proc. Vol. 40 (17), 337–342.

Sakthivel, V.P., Suman, M., Sathya, P.D., 2021. Combined economic and emission power dispatch problems through multi-objective squirrel search algorithm. Appl. Soft Comput. 100.

Sarkar, N., Podder, T., Antonelli, G., 2002. Fault-accommodating thruster force allocation of an AUV considering thruster redundancy and saturation. IEEE Trans. Robot. Autom. 18 (2), 223–233.

Scibilia, F., Skjetne, R., 2012. Constrained control allocation for vessels with azimuth thrusters. IFAC Proc. Vol. 45 (27), 7–12.

Shen, Q., Wang, D., Zhu, S., Poh, E.K., 2017. Robust control allocation for spacecraft attitude tracking under actuator faults. IEEE Trans. Control Syst. Technol. 25 (3), 1068–1075.

Shi, J.P., Zhang, W.G., Li, G.W., Liu, X.X., 2010. Research on allocation efficiency of the redistributed pseudo inverse algorithm. Sci. China Inf. Sci. 53 (2), 271–277.

Sørdalen, O., 1997. Optimal thrust allocation for marine vessels. Control Eng. Pract. 5 (9), 1223–1231.

Sørensen, A.J., 2011. A survey of dynamic positioning control systems. Annu. Rev. Control 35 (1), 123–136.

Soylu, S., Buckham, B.J., Podhorodeski, R.P., 2008. A chattering-free sliding-mode controller for underwater vehicles with fault-tolerant infinity-norm thrust allocation. Ocean Eng. 35 (16), 1647–1659.

Virnig, J., Bodden, D., 1994. Multivariable control allocation and control law conditioning when control effectors limit. In: Guidance, Navigation, and Control Conference. p. 3609.

Wang, Y., Gu, J., Zou, C., 2013. Thrust allocation in dynamic positioning system based on particle swarm optimization algorithm. In: 2013 OCEANS - San Diego. pp. 1–6.

Wang, X., Yang, G.-H., 2016. Distributed fault-tolerant control for a class of cooperative uncertain systems with actuator failures and switching topologies. Inform. Sci. 370–371, 650–666.

Wu, D.F., Ren, F.K., Zhang, W.D., 2016. An energy optimal thrust allocation method for the marine dynamic positioning system based on adaptive hybrid artificial bee colony algorithm. Ocean Eng. 118, 216–226.

Yadav, P., Kumar, R., Panda, S.K., Chang, C.S., 2014. Optimal thrust allocation for semisubmersible oil rig platforms using improved harmony search algorithm. IEEE J. Ocean. Eng. 39 (3), 526–539.

Zheng, T., Luo, W., 2019. An improved squirrel search algorithm for optimization. Complexity 2019, 6291968.