

Hierarchical Control of Marine Vehicles for Autonomous Maneuvering in Offshore Operations

Guoyuan Li, Wei Li, Karl Henning Halse, Hans Petter Hildre, and Houxiang Zhang^{*†}

February 23, 2015

Abstract

This paper presents a hierarchical control scheme that can achieve autonomous maneuvering for marine vehicles in offshore operations. Different from dynamic positioning problems, fine manoeuvring not only requests low speed of vessels and high control accuracy, but also has very strict and limited working spaces during offshore operations, such as oil platform maintenance, seabed survey, and pipeline installations. To address fine maneuvering, a two level control architecture is proposed with an emphasis on developing autonomous maneuvering from a specific application to final thrust allocation. On the high level, a ship controller is designed for regulation and dynamic positioning of ships. Thanks to the function approximation capabilities of neural networks, the controller is able to force the ship to desired targets with unknown dynamics of the ship. A thrust allocator is further developed on the low level that distributes control forces to individual actuators while satisfying optimizing constraints. By linearizing the constraints and employing the quadratic programming approach, an optimized thrust allocation is obtained. Both the ship controller and the thrust allocator are verified by simulation with the Cybership II model. The results confirm the effectiveness of the proposed hierarchical control scheme.

Keywords: Hierarchical control, ship maneuvering, neural network controller, thrust allocation

^{*}This work is supported by a grant from “MS GUNNERUS” project.

[†]G. Y. Li, W. Li, K. H. Halse, H. P. Hildre, and H. X. Zhang are with the Faculty of Maritime Technology and Operations, Aalesund University College, Postboks 1517, N-6025, Aalesund, Norway (corresponding author: G. Y. Li, email: gulj@hials.no).

1 Introduction

With the growth of emerging demands from civilian and military applications, the control of marine vehicles has become an intensive research topic in recent years. More and more offshore applications, such as the operations around the offshore oil drilling platform, as shown in Fig. 1, demand fine maneuvering of marine vehicles nearby the target and collision avoidance in the designated area. Fine maneuvering does not need fast approaching but require high control accuracy of positioning. It is tough for accomplishing by manual control of marine vehicles, since the operators have to deal with multiple thrusters while keeping the safety of operations. To release the burden of the operators, new knowledge and technology for fine manoeuvring are urgently demanded.

According to the types of applications, marine vehicles are classified as autonomous surface vehicles (ASVs) and autonomous underwater vehicles (AUVs). Control of the two types of marine vehicles not only requires an accurate dynamic model and a prior knowledge of environmental disturbances, but also takes into account constraints such as the power consumption and the saturation of thrusters. However, the dynamic model is nonlinear, time-varying and coupled in environment, and the perturbations from current, waves and wind are unpredictable. In addition, how to distribute the generalized forces to thrust devices while satisfying the constraints is also complex. Thus, developing a high-performance control system from a specific application to final thrust allocation is still challenging [1].

In the literature, control of marine vehicles is divided into two independent subproblems. On the one hand, the ship maneuvering problem such as the tracking problem in dynamic positioning system has attracted a lot of attention from control communities [2]. Conventional PID control is one of the

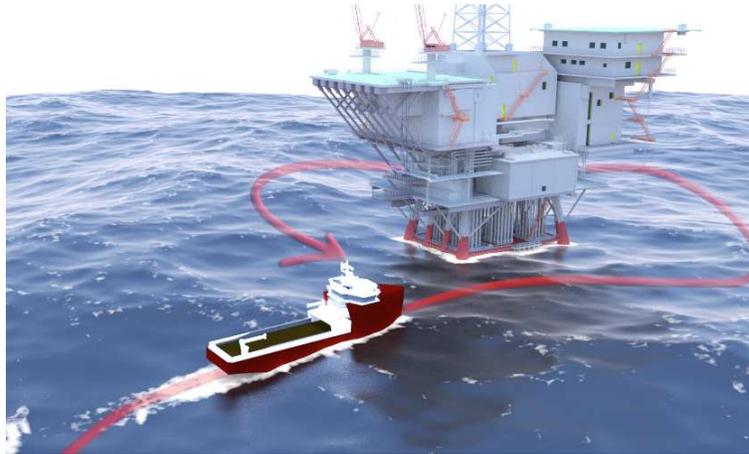


Figure 1: Concept of fine maneuvering in offshore operations.

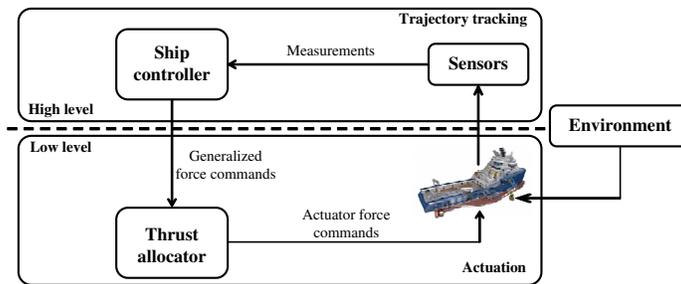


Figure 2: Hierarchical control scheme.

most common control approaches [3]. Although PID control is simple and easy to implement, it has poor ability to cope with perturbations and thus may cause poor control performance [4]. Compared to PID control approach, nonlinear adaptive method and sliding mode control method taking advantages of Lyapunov theory and backstepping technique, have better control performance [5] [6]. However, both of them need an accurate dynamic model of the vehicle which can only be partially depicted using current modeling techniques. To overcome the uncertainty of the model, intelligent control method including fuzzy control and neural network control has been proposed to approximate the dynamic model [7–9]. Because the fuzzy control mainly obtains rules by trial and error from experiences, it is nontrivial and time consuming in practice. In contrast, neural network controllers have a key ability of deterministic learning, which are able to adapt to unknown system dynamics through online adjustment of control parameters [10–13].

On the other hand, control allocation problems for marine vehicles has also received attention [14]. Control allocation can be formulated as a dynamic nonlinear optimization problem. Most control allocation approaches rely on linear model that describes the constraints in a linear manner. Solutions can be found in both an explicit form and numerical optimization. Explicit solutions are implemented by combining simple matrix computation such as generalized inverses, and other algorithm used for solving singularity and nonlinear constraints such as the singular value decomposition and filtering techniques [15] [16]. The benefit of the solution is its efficient real-time implementation. An alternative method is to use numerical optimization techniques, such as quadratic programming (QP), to iteratively handle nonlinearities and constraints [17–19]. Although the iterative way increases computational complexity, a warm start can guarantee the real-time optimization. More importantly, more solvers using efficient interior point algorithms emerge for QP problems [20].

Current research considers the control allocator is a modular design since the ship maneuvering problems can be solved without any knowledge of the ship’s thrusters [1]. This is true in theory.

However, in practice, the separation design may lead to mission failure if the controller for ship maneuvering ignores thruster features and capabilities and generates unattainable forces. Unfortunately, the necessity of hierarchical control is less concerned and seldom papers involve in this topic. Our project focuses on the hierarchical control of marine vehicles. Hierarchical control offers the operators intuitive guidance from a custom application to feasible thrust allocation. Furthermore, hierarchical control provides a possible way of verification of thruster configuration for ship design. Performance tests can be easily carried out in simulation with different thrust configurations. In this paper, we address the hierarchical control of ASVs to achieve trajectory tracking in offshore operations. Fig. 2 shows the two level control architecture. On the high level, sensor information is gathered and transmitted to a ship controller. The ship controller plays a role in motion planning, generating surge and sway forces and yaw moment to force the ship to follow the desired trajectory. On the low level, the generalized forces and moment are decomposed to each actuator by a thrust allocator using QP optimizing solver. As a result, the control system can be realized for robust tracking control of ASVs. In addition to the hierarchical control, this paper also emphasizes the design of adaptive neural network controller for autonomous maneuvering. Considering that traditional control techniques require prior knowledge of ship model and time consuming for solving ship maneuvering problems, a neural network method together with backstepping technique is proposed to approximate the nonlinear dynamics of the ship model. The main contributions of this paper include:

- An adaptive neural network control algorithm together with a stability proof is proposed without prior knowledge of dynamics of vehicles.
- A complete simulation of hierarchical control for trajectory tracking is carried out, using the Cybership II data [21].

The rest of the paper is organized as follows. Section 2 presents the details of the adaptive neural network controller at the high level. In Section 3, how to formulate thrust region and linearize constraints are introduced, which is followed by simulations in Section 4. Conclusions are given finally.

2 Ship Controller Design

In this section, an adaptive neural network controller is designed for trajectory tracking of ASVs. A radial basis function neural network is used for modeling the dynamics of an ASV. A rule for online

weight update is derived via Lyapunov theory. The stability and the convergence of the tracking error are both guaranteed.

2.1 Radial Basis Function Neural Network

As one type of function approximators, the radial basis function neural network (RBFNN) is usually used to represent nonlinear functions [22]. An RBFNN typically has three layers: an input layer, a hidden layer and an output layer. Assume the number of nodes in the three layers are m , n and p , respectively. The RBFNN output $F(X)$ can be described as:

$$F(X) = W^T R(X) + \epsilon(X), \quad (1)$$

where $X \in R^m$ represents the input vector; $W \in R^{n \times p}$ is the weight matrix; $R \in R^n$ is the radial basis functions in the hidden layer; and ϵ is the approximation error. Here, the Gaussian function is chosen as the basis function for the hidden vector $R(X) = [r_1(X), r_2(X), \dots, r_i(X), \dots, r_n(X)]^T$, which has the form:

$$r_i(X) = \exp(-\|X - \mu_i\|^2/2\sigma^2), \quad (2)$$

where $\mu_i = [\mu_{i1}, \mu_{i2}, \dots, \mu_{im}]^T$ is the center of the i -th Gaussian function and σ is the width of the Gaussian function, and $\|\cdot\|$ donates the Euclidean norm of vectors.

[22] has shown that if the node number in the hidden layer n is sufficiently large, there exists an ideal constant weight matrix W^* that the RBFNN output $F(X)$ can smoothly approximate any continuous functions to any degree of accuracy, i.e., for any $\epsilon^* > 0$ and $F(X) = W^{*T} R(X) + \epsilon(X)$, it always satisfies $|\epsilon(X)| < \epsilon^*$.

2.2 Ship Dynamics

Consider the dynamics of a three degree-of-freedom (3-DOF) ASV [1]:

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + \Delta = \tau, \quad (3)$$

$$\dot{\eta} = J(\eta)\nu, \quad (4)$$

where M , $C(\nu)$ and $D(\nu)$ are the ship inertia matrix, the total Coriolis and centripetal acceleration matrix and the damping matrix, respectively; $\eta = [x, y, \psi]^T$ represents the positions (x, y) and the

heading ψ in the earth-fixed frame; $\nu = [u, v, r]^T$ represents velocities in surge, sway and yaw in the body-fixed frame, respectively; $\Delta \in R^3$ is the environmental disturbance vector; $\tau \in R^3$ is the input signals. The rotation matrix $J(\eta)$ is given by:

$$J(\eta) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (5)$$

The dynamics of (3) can be rewritten in terms of η :

$$M_\eta \ddot{\eta} + C_\eta \dot{\eta} + D_\eta \dot{\eta} + \Delta_\eta = \tau_\eta, \quad (6)$$

where

$$\begin{aligned} M_\eta &= JMJ^{-1}, \\ C_\eta &= J[C(\nu) - MJ^{-1}\dot{J}]J^{-1}, \\ D_\eta &= JD(\nu)J^{-1}, \\ \Delta_\eta &= J\Delta, \\ \tau_\eta &= J\tau. \end{aligned}$$

with the following properties [1]:

1. The matrix M_η is symmetric positive definite;
2. The matrix D_η is positive definite;
3. And the matrix $\dot{M}_\eta - 2C_\eta$ is skew-symmetric.

2.3 Design of Adaptive Neural Network Controller

In this paper, we have two assumptions [4]:

1. Both η and ν are available without measurement error;
2. Δ_η is time-variant and bounded, i.e., there exists a constant vector $\bar{\Delta} = [\bar{\Delta}_1, \bar{\Delta}_2, \bar{\Delta}_3]^T$ such that

$$|\Delta_\eta| \leq \bar{\Delta}. \quad (7)$$

The control objective is to generate bounded forces and moment τ_η such that the ASV follows a desired trajectory η_d with the tracking error converging to a small neighborhood of the origin.

Define $\tilde{\eta} = \eta - \eta_d$ as the tracking error. Consider a Lyapunov function candidate:

$$V_1 = \frac{1}{2} \tilde{\eta}^T \tilde{\eta}. \quad (8)$$

Its derivative is:

$$\dot{V}_1 = \tilde{\eta}^T \dot{\tilde{\eta}} = \tilde{\eta}^T (\dot{\eta} - \dot{\eta}_d). \quad (9)$$

If the velocity vector $\dot{\eta}$ follows a command signal $\dot{\eta}_c$, i.e.,

$$\dot{\eta}_c = \dot{\eta}_d - K_\eta \tilde{\eta}, \quad (10)$$

where $K_\eta \in R^{3 \times 3}$ is a diagonal positive definite design parameter matrix, then substituting (10) into (9) yields

$$\dot{V}_1 = -\tilde{\eta}^T K_\eta \tilde{\eta} \leq 0. \quad (11)$$

Note $\dot{V}_1 = 0$ only if $\tilde{\eta} = 0$. Thus, the tracking error $\tilde{\eta}$ under the control law (10) is asymptotically stable.

Due to the uncertainty of M_η , C_η , D_η and Δ_η in (6), a vectorial backstepping with RBFNN is employed to approximate them. Let $\dot{\eta}$ be the virtual control input, and $\dot{\eta}_c$ be the virtual control law as in (10). Define the velocity error as:

$$e_c = \dot{\eta}_c - \dot{\eta}. \quad (12)$$

Consider the Lyapunov function candidate V_2 :

$$V_2 = V_1 + \frac{1}{2} e_c^T M_\eta e_c. \quad (13)$$

Taking the time derivative of (13) and using (6) and its property 1 yield:

$$\begin{aligned} \dot{V}_2 &= \dot{V}_1 + e_c^T M_\eta \dot{e}_c + \frac{1}{2} e_c^T \dot{M}_\eta e_c \\ &= \dot{V}_1 + e_c^T [M_\eta \ddot{\eta}_c + C_\eta \dot{\eta}_c + D_\eta \dot{\eta}_c + \Delta_\eta - \tau_\eta] \\ &\quad + \frac{1}{2} e_c^T (\dot{M}_\eta - 2C_\eta) e_c - e_c^T D_\eta e_c. \end{aligned} \quad (14)$$

Define an RBFNN f to approximate the dynamics of the ASV:

$$f = \omega^T r(X) + \epsilon(X) = M_\eta \ddot{\eta}_c + C_\eta \dot{\eta}_c + D_\eta \dot{\eta}_c + \Delta_\eta, \quad (15)$$

where $X = [\ddot{\eta}_c^T, \dot{\eta}_c^T]^T$ is the input vector of the RBFNN; $r(X) \in R^n$ is the basis function vector, and n is the node number in the hidden layer; $\epsilon(X)$ is the approximation error. The feedback control law is chosen as:

$$\tau_\eta = \hat{\omega}^T r + K_c e_c, \quad (16)$$

where $\hat{\omega} \in R^{n \times 3}$ is the weight estimate of the RBFNN; $K_c \in R^{3 \times 3}$ is a positive definite design matrix. Combining the property 3 of (6) and substituting (15) and (16) into (14), we have

$$\dot{V}_2 = \dot{V}_1 - e_c^T (D_\eta + K_c) e_c + e_c^T \epsilon - e_c^T \tilde{\omega}^T r. \quad (17)$$

where $\tilde{\omega}^T = \hat{\omega}^T - \omega^T$ is the corresponding weight estimate error of the RBFNN. As mentioned in 2.1, the optimal weights of the RBFNN can be approached by applying enough node numbers in the hidden layer. Here, suppose the hidden nodes n is sufficient. The Lyapunov function candidate V_3 is chosen as

$$V_3 = V_2 + tr\left(\frac{1}{2} \tilde{\omega}^T \Gamma^{-1} \tilde{\omega}\right), \quad (18)$$

where $\Gamma \in R^{n \times n}$ is a positive definite design matrix. Differentiating V_3 with respect to time yields

$$\begin{aligned} \dot{V}_3 &= \dot{V}_2 + tr(\tilde{\omega}^T \Gamma^{-1} \dot{\tilde{\omega}}) \\ &= \dot{V}_1 - e_c^T (D_\eta + K_c) e_c + e_c^T \epsilon - e_c^T \tilde{\omega}^T r + tr(\tilde{\omega}^T \Gamma^{-1} \dot{\tilde{\omega}}). \end{aligned} \quad (19)$$

Note that $e_c^T \tilde{\omega}^T r$ is a scalar and the fact that the trace of product of two matrices satisfies swap property, i.e., $tr(AB) = tr(BA)$. Furthermore, if choosing the update law of the weight of the RBFNN as

$$\dot{\tilde{\omega}} = \dot{\hat{\omega}} = \Gamma(r e_c^T - K_\omega \hat{\omega}), \quad (20)$$

then (19) becomes

$$\begin{aligned}\dot{V}_3 &= \dot{V}_1 - e_c^T(D_\eta + K_c)e_c + e_c^T\epsilon - tr(\tilde{\omega}^T r e_c^T) + tr(\tilde{\omega}^T \Gamma^{-1} \Gamma(r e_c^T - K_\omega \hat{\omega})) \\ &= -\tilde{\eta}^T K_\eta \tilde{\eta} - e_c^T(D_\eta + K_c)e_c + e_c^T\epsilon - tr(\tilde{\omega}^T K_\omega \tilde{\omega}) - tr(\tilde{\omega}^T K_\omega \omega),\end{aligned}\quad (21)$$

where $K_\omega \in R^{n \times n}$ is a diagonal positive definite design parameter matrix. Considering the following complete square inequalities:

$$e_c^T\epsilon \leq \frac{e_c^T e_c}{2} + \frac{1}{2}\|\epsilon^*\|^2, \quad (22)$$

$$-tr(\tilde{\omega}^T K_\omega \omega) \leq \frac{tr(\tilde{\omega}^T K_\omega \tilde{\omega})}{2} + \frac{tr(\omega^T K_\omega \omega)}{2} \leq \frac{tr(\tilde{\omega}^T K_\omega \tilde{\omega})}{2} + \frac{tr(\omega^{*T} K_\omega \omega^*)}{2}, \quad (23)$$

where ϵ^* is the bound of the approximation error, and ω^* is the ideal weight matrix, \dot{V}_3 satisfies

$$\begin{aligned}\dot{V}_3 &\leq -\lambda_{min}(K_\eta)\tilde{\eta}^T \tilde{\eta} - \lambda_{min}\left\{(D_\eta + K_c - \frac{I_{3 \times 3}}{2})M_\eta^{-1}\right\}e_c^T M_\eta e_c \\ &\quad - \lambda_{min}\left\{\frac{K_\omega}{2}\Gamma^{-1}\right\}tr(\tilde{\omega}^T \Gamma^{-1} \tilde{\omega}) + \frac{\|\epsilon^*\|^2}{2} + \frac{tr(\omega^{*T} K_\omega \omega^*)}{2} \\ &\leq -\alpha_0 V_3 + \beta_0,\end{aligned}\quad (24)$$

with

$$\alpha_0 = \min(2\lambda_{min}(K_\eta), 2\lambda_{min}\left\{(D_\eta + K_c - \frac{I_{3 \times 3}}{2})M_\eta^{-1}\right\}, 2\lambda_{min}\left\{\frac{K_\omega}{2}\Gamma^{-1}\right\}), \quad (25)$$

$$\beta_0 = \frac{\|\epsilon^*\|^2}{2} + \frac{tr(\omega^{*T} K_\omega \omega^*)}{2}, \quad (26)$$

where $\lambda_{min}(\cdot)$ represents the minimum eigenvalue of a matrix; $\|\cdot\|^2$ denotes the Euclidean norm of a vector. If K_c satisfies $K_c \geq \frac{I_{3 \times 3}}{2}$, the above inequality (24) can be solved as

$$0 \leq V_3(t) \leq (V_3(0) - \frac{\beta_0}{\alpha_0})e^{-\alpha_0 t} + \frac{\beta_0}{\alpha_0}. \quad (27)$$

From (27), it is observed that $V_3(t)$ is globally uniformly ultimately bounded. Hence, the signals $\tilde{\eta}$, e_c and $\tilde{\omega}$ are globally uniformly ultimately bounded. Since η_d and $\dot{\eta}_d$ are known to be bounded, η and $\dot{\eta}_c$ are bounded, and in turn $\dot{\eta}$ is bounded. As the ideal weight matrix ω^* is constant, $\hat{\omega}$ is also bounded. As a result, all signals are bounded, and the tracking error $\tilde{\eta}$ converges to a compact set $\Omega = \{\tilde{\eta} \in R^3 \mid \|\tilde{\eta}\| \leq \sqrt{2\beta_0/\alpha_0}\}$ as t tends to infinity. This means that by appropriately choosing the design matrices K_η , K_c and K_ω , the compact set Ω can be made within an arbitrarily small

neighborhood of zero. Thus, the ASV can converge to the target position and orientation with the desired accuracy.

3 3-DOF Thrust Allocation

Given the commanded force vector $\tau = J^{-1}\tau_\eta$ computed by the ship controller at the high level, the thrust allocator at the low level deals with the mapping from the force and moment to a pair of angle and thrust for each actuator, while satisfying the power, magnitude and other constraints.

3.1 Configuration Matrix

Consider an ASV equipped with h actuators. The generalized force vector τ is a linear combination of the forces and the moment for all actuators, which can be described as a function of rotation angle α between the actuator axis and the ASV, and the magnitude of the force of actuators:

$$\tau = B_0(\alpha)U_0, \quad (28)$$

where $B_0(\alpha) \in R^{3 \times h}$ denotes the configuration matrix, and $U_0 \in R^h$ represents a vector of magnitude of the forces produced by individual actuator. The i -th column of $B_0(\alpha)$ is given by

$$B_{0i}(\alpha_i) = \begin{pmatrix} \cos(\alpha_i) \\ \sin(\alpha_i) \\ -L_{yi}\cos(\alpha_i) + L_{xi}\sin(\alpha_i) \end{pmatrix}, \quad (29)$$

where (L_{xi}, L_{yi}) is the location of the i -th actuator with respect to the center of gravity of the ASV. Note B_0 is variable with respect to α .

In order to avoid complicated nonlinear relationships, extended thrust force is utilized here [15]. Assume each actuator contains two states: $u = [u_x, u_y]^T$, representing the force acting on the surge and sway direction of the ASV, respectively. Then, the force and moment generated by all actuators

are formed:

$$\begin{aligned} \tau &= BU \\ &= \begin{bmatrix} 1 & 0 & \cdots & 1 & 0 \\ 0 & 1 & \cdots & 0 & 1 \\ -L_{y1} & L_{x1} & \cdots & -L_{yh} & L_{xh} \end{bmatrix} \begin{bmatrix} u_{x1} \\ u_{y1} \\ \vdots \\ u_{xh} \\ u_{yh} \end{bmatrix}, \end{aligned} \quad (30)$$

where U is a vector of surge and sway forces generated by all actuators, and $B \in R^{3 \times 2h}$ is the static configuration matrix which only depends on the position of actuators.

3.2 Thrust Region Formation

According to the working areas, actuators are divided into non-rotatable and rotatable models [14]. For different types of actuators, a set of inequalities is created to linearize to the thrust region of actuators [20].

3.2.1 Fixed Thruster

A fixed thrust model is a line segment. It keeps a constant angle $\bar{\alpha}$ within its thrust limits $[T_{min}, T_{max}]$. Combining the form of actuator states, the thrust region constraint for fixed thrusters can be described as

$$A_f u \leq b_f, \quad (31)$$

with an equivalent form of

$$\begin{bmatrix} \sin(\bar{\alpha}) & -\cos(\bar{\alpha}) \\ -\sin(\bar{\alpha}) & \cos(\bar{\alpha}) \\ \cos(\bar{\alpha}) & \sin(\bar{\alpha}) \\ -\cos(\bar{\alpha}) & -\sin(\bar{\alpha}) \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ T_{max} \\ -T_{min} \end{bmatrix}. \quad (32)$$

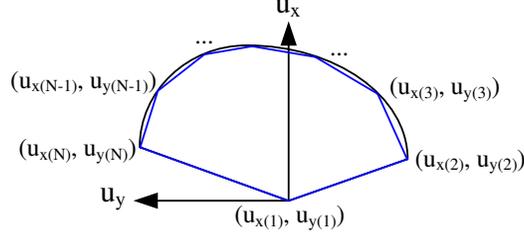


Figure 3: Thrust region linear approximation.

3.2.2 Rotatable Thruster

For a rotatable thruster, its thrust region is more complex. On the one hand, the magnitude of force u is no more than the thruster's extreme T_{max} :

$$|u| = \sqrt{u_x^2 + u_y^2} \leq T_{max}. \quad (33)$$

On the other hand, there is forbidden zone that the thrust device should or can not approach. Thus, the resultant thrust region is a circular-like sector.

Note (33) is a nonlinear constraint that needs to be linearized before QP optimization. Here we only discuss the convex thrust region. Non-convex thrust region can be split into several convex thrust regions. Fig. 3 shows the linear approximation of a convex thrust region. It is sampled with an acceptable error such that a polygon with N vertices is formed. For every two adjacent vertices $(u_{x(i)}, u_{y(i)})$ and $(u_{x(i+1)}, u_{y(i+1)})$, where $i = 1, \dots, N$, the linearized inequality constraint is

$$u_y - u_{y(i)} \leq \frac{u_{y(i+1)} - u_{y(i)}}{u_{x(i+1)} - u_{x(i)}} (u_x - u_{x(i)}), \quad (34)$$

namely,

$$\begin{bmatrix} a_{1,i} & a_{2,i} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \leq b_{r(i)}, \quad (35)$$

where

$$a_{1,i} = u_{y(i+1)} - u_{y(i)},$$

$$a_{2,i} = u_{x(i)} - u_{x(i+1)},$$

$$b_{r(i)} = u_{x(i)}u_{y(i+1)} - u_{x(i+1)}u_{y(i)}.$$

Because the polygon is a closed shape, we have $(u_{x(1)}, u_{y(1)}) = (u_{x(N+1)}, u_{y(N+1)})$. Taking all pairs of adjacent vertices in a counterclockwise order, the thrust region is formed as

$$A_r u \leq b_r, \quad (36)$$

corresponding to

$$\begin{bmatrix} a_{1,1} & a_{2,1} \\ \vdots & \vdots \\ a_{1,N+1} & a_{2,N+1} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \leq \begin{bmatrix} b_{r(i)} \\ \vdots \\ b_{r(N+1)} \end{bmatrix}. \quad (37)$$

3.3 QP Optimization

Based on the configuration matrix and the formation of thrust regions, the QP problem for thrust allocation can be solved in QP solvers. Considering minimizing the power consumption as well as the largest thrust force, the QP problem is described as

$$\begin{aligned} J &= \min_{U, S, \bar{u}} \{U^T P U + S^T Q S + \beta \bar{u}\}, \\ s.t. \quad & B U - S = \tau, \\ & A U \leq b, -\bar{u} \leq U \leq \bar{u}, \end{aligned} \quad (38)$$

with

$$\begin{aligned} b &= [b_{f1}, \dots, b_{fh}, b_{r1}, \dots, b_{rh}]^T, \\ A &= \text{diag}(A_{f1}, \dots, A_{fh}, A_{r1}, \dots, A_{rh}), \end{aligned}$$

where S is the slack variable; $\bar{u} = \max U$ is the largest force among the force vector U ; β is a positive constant; P and Q are diagonal positive definite matrix with appropriate dimension, where $\det(Q) \gg \det(P) > 0$.

In order to make QP solvers, e.g., Matlab, understand the description of QP problem, Eq. (38) needs to be rewritten in a standard QP form. Let $Z = [U, S, \bar{u}]^T$, then the standard form is given as

follow:

$$J = \min_Z Z^T \begin{bmatrix} P & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & 0 \end{bmatrix} Z + \begin{bmatrix} 0 & 0 & \beta \end{bmatrix} Z,$$

$$s.t. \quad \begin{bmatrix} B & -I & 0 \end{bmatrix} Z = \tau, \quad \begin{bmatrix} A & 0 & 0 \\ -I & 0 & -1 \\ I & 0 & -1 \end{bmatrix} Z \leq \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix}.$$

The QP solver returns an optimized vector of U , containing the surge and sway force pair of each thruster. All these pairs are then further transformed into a polar form. Thus, the thrust allocation completes.

4 Simulation

To validate the effectiveness of the proposed hierarchical control design, numerical simulations based on the Cybership II model are carried out. The Cybership II is a 1:70 scale model of a supply vessel, which has a mass of 23.8kg and a length of 1.255m. For more details of the model, please refer to [21].

4.1 Trajectory Tracking Experiment

For this experiment, the dynamic parameters of the model represented by (3) are given as follows

$$M = \begin{bmatrix} 25.8 & 0 & 0 \\ 0 & 24.661 & 1.095 \\ 0 & 1.095 & 2.76 \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & 0 & -24.661v - 1.095r \\ 0 & 0 & 25.8u \\ 24.661v + 1.095r & -25.8u & 0 \end{bmatrix},$$

$$D = \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & d_{23} \\ 0 & d_{32} & d_{33} \end{bmatrix},$$

Table 1: RBFNN Parameters

Symbols	Values	Description
m	6	Input nodes
n	1000	Hidden nodes
p	3	Output nodes
σ	5	Width of Gaussian function
μ	[-1,1]	Center of Gaussian function
Γ	I_{1000}	Weight update rate

with

$$\begin{aligned}
d_{11} &= 0.7225 + 1.3274|u| + 5.8664u^2, \\
d_{22} &= 0.8612 + 36.2823|v| + 8.05|r|, \\
d_{23} &= -0.1079 + 0.845|v| + 3.45|r|, \\
d_{32} &= -0.1052 - 5.0437|v| - 0.13|r|, \\
d_{33} &= 1.9 - 0.08|v| + 0.75|r|.
\end{aligned}$$

The disturbance vector Δ is defined as a time-varying function

$$\Delta = \begin{bmatrix} 0.2 + 0.1\sin(0.2t) + 0.3\sin(0.4t) + 0.3\cos(0.2t) \\ 0.2 + 0.1\sin(0.2t) + 0.2\sin(0.1t) - 0.1\cos(0.4t) \\ 0.2 + 0.1\sin(0.2t) - 0.3\sin(0.4t) - 0.5\cos(0.1t) \end{bmatrix}.$$

The above matrices are unknown to the control system. Assume the initial position of the vessel is $[0, 0, \pi/3]$. The reference trajectory is chosen as an ellipse with heading along the tangent direction

$$\begin{aligned}
x_d &= 3\sin(0.1t), \\
y_d &= 2\cos(0.1t), \\
\psi_d &= \tan^{-1}\left(\frac{\dot{y}_d}{\dot{x}_d}\right).
\end{aligned}$$

In the simulation, a RBFNN was constructed. Table 1 lists the parameters of the RBFNN. Note that the inputs of the RBFNN are normalized before function approximation. Therefore, the centers of the radial basis function μ are randomly spaced in $[-1, 1]$. The initial values of the weight estimation

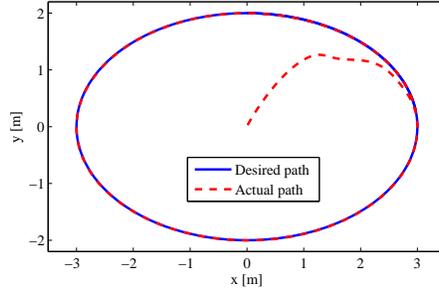


Figure 4: Tracking result of an elliptic trajectory.

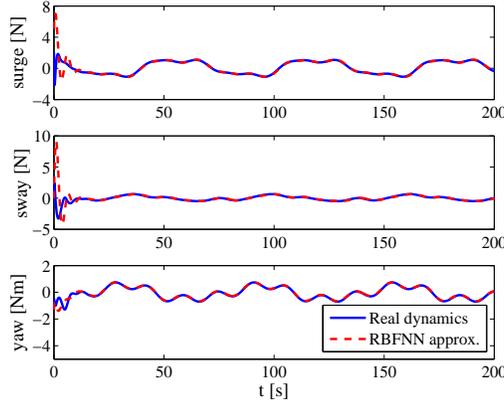


Figure 5: Function approximation.

$\hat{\omega}$ are set to zero. The design parameter matrices of the ship controller are taken as $K_\eta = 0.25I_3$, $K_c = 15I_3$ and $K_\omega = 0.02I_{1000}$.

The tracking performance is depicted in Fig. 4. The result shows that despite the existence of disturbance, the ASV converges to the desired path and tracks it with a high accuracy. Fig. 5 illustrates RBFNN based approximation of the dynamics of the ASV. The RBFNN output \hat{f} smoothly approximates the dynamic f in (15) in a short period of time. The fast and precise approximation ensures the convergence and the boundedness of tracking errors. The variation of surge velocity u , sway velocity v and yaw rate r with respect to time are shown Fig. 6. Fig. 7 shows the corresponding control forces and moment. Here we emphasize that the ship controller has taken the restrictions from the low level such as the maximum attainable forces into account. Thus, from a practical point of view, the control forces and moment as shown in Fig. 7 are reasonable and realistic.

The simulation results demonstrate that the proposed adaptive neural network controller is effective for motion control of ASVs.

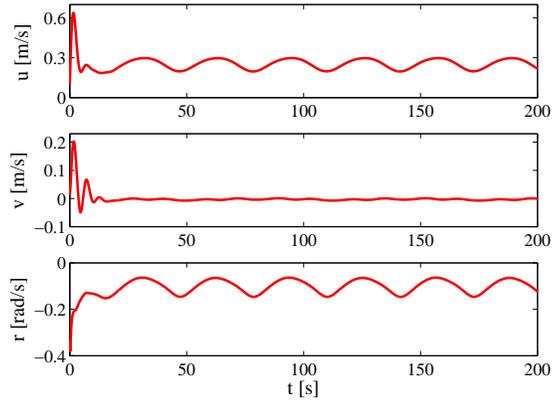


Figure 6: Tracking velocities.

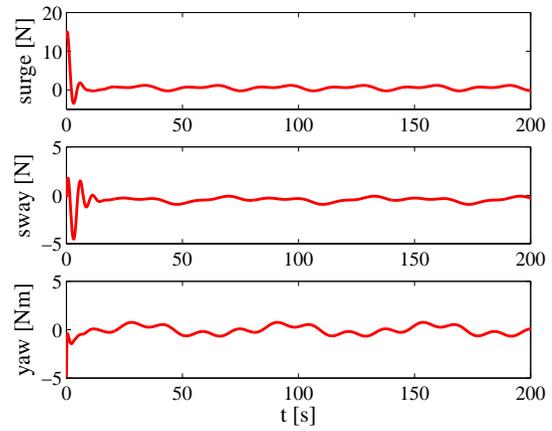


Figure 7: Control forces and moment.

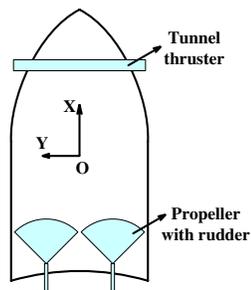


Figure 8: Thruster configuration of Cybership II.

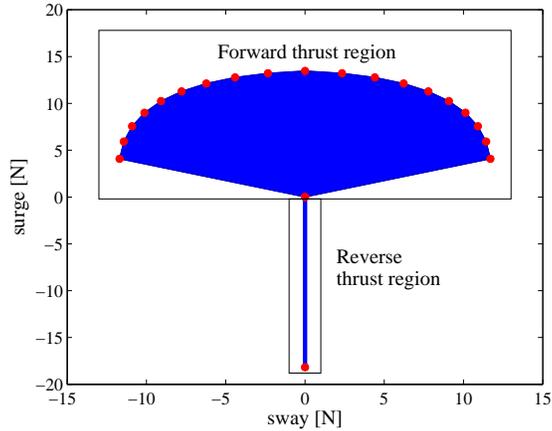


Figure 9: Thrust region approximation for a propeller with rudder.

Table 2: Thrust Devices Distribution

Thruster No.	Thruster type	Position	
		$L_x[m]$	$L_y[m]$
1	Tunnel thruster	1.14	0.0
2	Propeller with rudder	-0.54	0.075
3	Propeller with rudder	-0.54	-0.075

4.2 Thrust Allocation Experiment

The Cybership II is equipped with one tunnel thruster in the bow and two main propellers with rudders at the stern, as shown in Fig. 8. Table 2 lists the types of the three thrust devices, as well as their distribution. Assume the origin direction is towards the stern. The rotating angle is defined ranging from 0° to 360° , which increases with a counterclockwise rotation.

From the tunnel thruster model and parameters in [20], the tunnel thruster is fixed at 90° and produces a sway force within $[-6.62, 6.76]$.

The two main propellers with rudders are also fixed that point to the stern of the ship. Both of them have a nominal thrust along the surge direction. The propeller can operate in forward mode or reverse mode.

In forward mode, besides the nominal thrust, the rudder is able to deliver additional lift and drag forces to the ship. While in reverse mode, rotating the rudder has no effect on the ship. Fig. 9 shows the linear approximation of the thrust region for one of the propeller with rudder.

Due to the two disjunct thrust regions for each propeller with rudder, four possible thrust region combinations were taken into consideration in the simulation, as shown in Table 3. Each combination

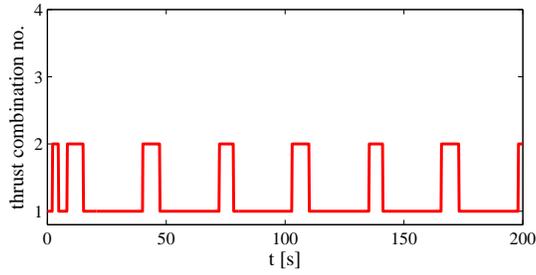


Figure 10: Thrust combination sequence.

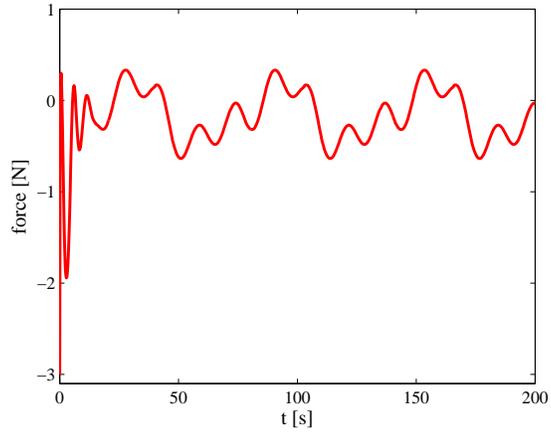


Figure 11: Thrust allocation for thruster No.1 (tunnel thruster).

has its own QP problem and has been formulated before simulation. A QP solver based on sequential quadratic programming method was used here to find the optimized thrust allocation for the three thrusters. The design parameters are chosen as: $P = I_6$, $Q = \text{diag}(100, 100, 1000)$ and $\beta = 1$.

In the simulation, the resultant forces and moment as shown in Fig. 7 was imported. Fig. 10 shows the thrust combination sequence with respect to time. The result shows that only the 1st and the 2^{ed} thrust combinations were utilized during the simulation. The thrust allocation for thruster No.1 is illustrated in Fig. 11. Because thruster No.1, i.e. the tunnel thruster, is fixed, only the magnitude

Table 3: Thrust Region Combinations

Combination No.	Thruster		
	No. 1	No. 2	No. 3
1	Line segment	Circular sector	Circular sector
2	Line segment	Circular sector	Line segment
3	Line segment	Line segment	Circular sector
4	Line segment	Line segment	Line segment

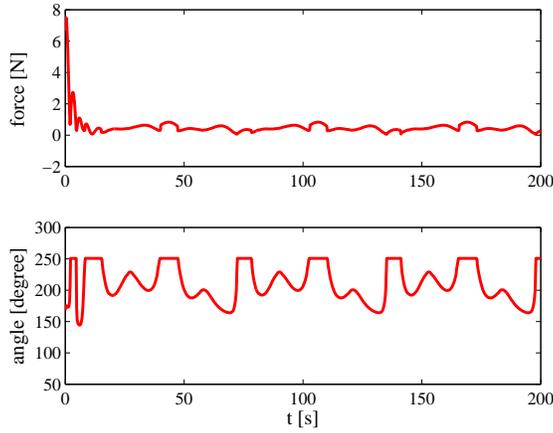


Figure 12: Thrust allocation for thruster No.2 (propeller with rudder).

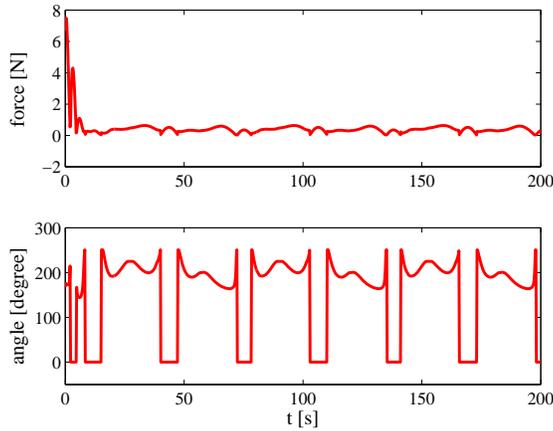


Figure 13: Thrust allocation for thruster No.3 (propeller with rudder).

of force in sway is depicted. Fig. 12 and Fig. 13 show the thrust allocation for thruster No.2 and No.3, respectively. From Fig. 12, the propeller with rudder was always operated in forward mode with the magnitude of force in $(0N, 8N)$ and the rotating angle in $(144^\circ, 251^\circ)$. Combining Fig. 10 with Table 3, thruster No.3 was operated in forward and reverse mode back and forth. In Fig. 13, the dramatic change from a rotating angle to zero indicates the change of operation from forward mode to reverse mode, and vice versa. This is reasonable because 1) the change of operation only reverses the propeller shaft without changing of the rudder and 2) the rudder does not deliver any forces when the propeller is in reverse mode. Therefore, the change of rudder in the simulation is actually smooth and attainable. Except for the reverse mode in Fig. 13, thruster No.3 has almost the same force range and the rotating angle range as thruster No.2 has. From the simulation results, the QP method with linearized constraints is validated useful for thrust allocation.

5 Conclusion

In this paper, we address the design of a hierarchical control scheme for autonomous maneuvering marine vehicles in offshore operations. A robust neural network based controller in conjunction with a quadratic programming based thrust allocator comprises the hierarchical control system. Through Lyapunov stability analysis and linearizing approximation, the control system is proved capable of regulating the position and heading of the ship to the desired targets without explicit prior knowledge of the ship's dynamics. Simulations using the Cybership II model show the feasibility of the proposed approach in realizing autonomous maneuvering of marine vehicles.

The research of hierarchical control for autonomous maneuvering is still undergoing. We are now making efforts on the influence of the limitation of actuators on the ship controller. More constraints related to the thruster features and capabilities such as the thruster's rotation speed and response time are taken into consideration. Based on current results, future work will focus on: (1) Developing training and evaluation system for nautical certification; (2) Implementing real-time guidance system to provide the pilot with optimized operations.

References

- [1] **Fossen, T. I.** (2002). *Marine control systems: guidance, navigation and control of ships, rigs and underwater vehicles*. Trondheim: Marine Cybernetics.
- [2] **Sørensen, A. J.** (2011). *A survey of dynamic positioning control systems*. *Annual reviews in control*, 35(1): 123-136.
- [3] **Loria, A. and Fossen, T. I.** (2000). *A separation principle for dynamic positioning of ships: theoretical and experimental results*. *IEEE Trans. Control Syst. Technol*, 8(2): 332-343.
- [4] **Du, J., Yang, Y., Wang, D. and Guo, C.** (2013). *A robust adaptive neural networks controller for maritime dynamic positioning system*. *Neurocomputing*, 110: 128-136.
- [5] **Aguiar, A. P. and Hespanha, J. P.** (2007). *Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty*. *IEEE Trans. Automatic Control*, 52(8): 1362-1379.
- [6] **Ashrafiuon, H., Muske, K. R., McNinch, L. C. and Soltan, R. A.** (2008). *Sliding-mode tracking control of surface vessels*. *IEEE Trans. Industrial Electronics*, 55(11): 4004-4012.

- [7] **Velagic, J., Vukic, Z. and Omerdic, E.** (2003). *Adaptive fuzzy ship autopilot for track-keeping. Control engineering practice*, 11(4): 433-443.
- [8] **Rigatos, G. and Tzafestas, S.** (2006). *Adaptive fuzzy control for the ship steering problem. Mechatronics*, 16(8): 479-489.
- [9] **Dai, S. L., Wang, C. and Luo, F.** (2012). *Identification and learning control of ocean surface ship using neural networks. IEEE Trsns. Industrial Informatics*, 8(4): 801-810.
- [10] **Kim, M. H.** (2000). *Nonlinear control and robust observer design for marine vehicles. Ph.D. dissertation, Virginia Polytechnic Institute and State University.*
- [11] **Leonessa, A., VanZwieten, T. and Morel, Y.** (2006). *Neural network model reference adaptive control of marine vehicles. Current Trends in Nonlinear Systems and Control. Birkhäuser Boston*, 421-440.
- [12] **Tee, K. P. and Ge, S. S.,** (2006). *Control of fully actuated ocean surface vessels using a class of feedforward approximators. IEEE Trans. Contr. Syst. Technol.* 14(4): 750-756.
- [13] **Pan, G. Z., Lai, X. Z., Yang, S. X. and Wu, M.** (2013). *An efficient neural network approach to tracking control of an autonomous surface vehicle with unknown dynamics, Expert Systems with Applications*, 40(5): 1629-1635.
- [14] **Fossen, T. I. and Johansen, T. A.** (2006). *A survey of control allocation methods for ships and underwater vehicles. In the 14th Mediterranean Conf. Control and Automation*, 1-6.
- [15] **Sørdalen, O. J.** (1997). *Optimal thrust allocation for marine vessels. Control Engineering Practice*, 5(9): 1223-1231.
- [16] **Lindgaard, K. P. and Fossen, T. I.** (2003). *Fuel-efficient rudder and propeller control allocation for marine craft: experiments with a model ship. IEEE Trans. Contr. Syst. Technol.*, 11(6): 850-862.
- [17] **Johansen, T. A., Fossen, T. I. and Berge, S. P.** (2004). *Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming. IEEE Trans. Contr. Syst. Technol.*, 12(1): 211-216.

- [18] **Johansen, T. A., Fuglseth, T. P., Tøndel, P. and Fossen, T. I.** (2008). *Optimal constrained control allocation in marine surface vessels with rudders. Control Engineering Practice, 16(4): 457-464.*
- [19] **Wei, Y., Fu, M., Ning, J. and Sun, X.** (2013). *Quadratic programming thrust allocation and management for dynamic positioning ships. Journal of Electrical Engineering, 11(3): 1632-1638.*
- [20] **Wit, C. D.** (2009). *Optimal thrust allocation methods for dynamic positioning of ships. M.S. thesis, Delft University of Technology.*
- [21] **Lindegaard, K. P.** (2003). *Acceleration feedback in dynamic positioning. Ph.D. dissertation, Norwegian University of Science and Technology, Trondheim, Norway.*
- [22] **Park, J. and Sandberg, I. W.** (1991). *Universal approximation using radial-basis-function networks. Neural computation, 3(2): 246-257.*