Ocean Engineering 35 (2008) 1647–1659

Contents lists available at ScienceDirect

Ocean Engineering

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

A chattering-free sliding-mode controller for underwater vehicles with fault-tolerant infinity-norm thrust allocation

Serdar Soylu *, Bradley J. Buckham, Ron P. Podhorodeski

Department of Mechanical Engineering, University of Victoria, P.O. Box 3055, Victoria, BC, Canada V8W 3P6

ARTICLE INFO

Article history:
Received 30 November 2007
Accepted 23 July 2008
Available online 26 July 2008

Keywords:
Nonlinear control
Underwater vehicles
Thruster force allocation
Fault-tolerant systems

ABSTRACT

There are two objectives to this paper. First, a chattering-free sliding-mode controller is proposed for the trajectory control of remotely operated vehicles (ROVs). Second, a new approach for thrust allocation is proposed that is based on minimizing the largest individual component of the thrust manifold. With regards to the former, a new adaptive term is developed that eliminates the high-frequency control action inherent in a conventional sliding-mode controller. As opposed to the common adaptive approach, the new adaptive term does not require the linearity condition on the dynamic parameters and the creation of a regressor matrix. In addition, it removes the need for a *priori* knowledge of upper bounds on uncertainties in the dynamic parameters of the ROV. With regards to the latter, it is demonstrated that minimizing the l_∞ norm (infinity-norm) of the thrust manifold ensures low individual thruster forces. The new control and thrust allocation concepts are implemented in numerical simulations of a work class ROV, and the chattering-free nature of the controller is demonstrated during typical ROV manoeuvres. In the simulation studies, the l_∞ norm-based thrust allocation problem is cast as a linear programming problem that allows direct incorporation of the thruster saturation limits and a fault-tolerant property. To achieve real-time solution rates for the l_∞ norm-based thrust allocation problem, a recurrent neural network is designed. In the simulation studies, the l_∞ norm-based thrust allocation provides smaller maximum absolute value of the largest component of the thrust manifold than that of a conventional l_2 norm (2-norm) minimization, satisfies the saturation limits of each thruster, and accommodates faults that are introduced arbitrarily during the manoeuvre.

1. Introduction

Remotely operated vehicles (ROVs) play an important role in a number of shallow and deep-water missions for marine science, oil and gas extraction, exploration, and salvage. In these applications, the motions of the ROV are guided either by a human pilot on a surface support vessel through an umbilical cord providing power and telemetry, or by an automatic pilot. In the case of automatic control, ROV state feedback is provided by acoustic and inertial sensors, and this state information along with a controller strategy is used to drive several conventional thrusters arranged on the ROV chassis.

In the existing literature, several different automatic control approaches have been applied to control ROV motion such as the H_∞ approach (Conte and Serrani, 1998), adaptive control techniques (Antonelli et al., 2001, 2003), sliding-mode control (Yoerger and Slotine, 1985; Slotine and Coetsee, 1986; Healey and Lienard, 1993; Antonelli, 2003), fuzzy-logic control (Debitetto, 1994; Kato, 1995), and neural network methods (Ishii and Ura, 2000; Kodogiannis, 2003; Pepijn et al., 2005; Van de Ven et al., 2005). It has been shown that the model-based sliding-mode approach is an effective means of controlling an ROV, largely due to its ability to tolerate imprecision in the dynamics model (Yoerger and Slotine, 1985; Slotine and Coetsee, 1986). However, one major drawback of the sliding-mode approach is the high frequency of control action (chattering). This high-frequency control activity causes high heat losses in electrical power circuits and premature wear in actuators. In addition, the high control activity may excite unmodelled high-frequency dynamics, which in turn causes controller performance degradation. To eliminate or reduce chattering, various methods such as the boundary layer method (Yoerger and Slotine, 1985; Slotine and Shastry, 1983) and the disturbance compensation method (Elmali and Olgac, 1992; Zeinali and Notash, 2007) have been presented. The boundary layer approach makes the control activity continuous within the boundary layer and discontinuous outside the boundary layer. In this work, a disturbance compensation method is utilized.

* Corresponding author. Tel.: +1 250 472 4065; fax: +1 250 721 6035.
E-mail addresses: serdar@me.uvic.ca (S. Soylu), bbuckham@me.uvic.ca (B.J. Buckham), podhoro@me.uvic.ca (R.P. Podhorodeski).

0029-8018/$ - see front matter © 2008 Published by Elsevier Ltd.
doi:10.1016/j.oceaneng.2008.07.013

To eliminate chattering, the disturbance compensation approach replaces the discontinuous term of a conventional sliding-mode controller with an estimate of the uncertainties in an adaptive manner. In the current work, a disturbance compensation approach discussed by Zeinali and Notash (2007) for land-base manipulators is extended to mobile-base ROV systems.

To ensure manoeuvrability, the ROV thruster arrangement is redundant: there are more thrusters than there are active vehicle degrees of freedom. Due to the excess thrusters, there are an infinite number of ways to allocate the pilot's commanded generalized force over the existing thrusters. As such, an optimal thrust allocation must be selected by applying criteria to distinguish the various options. In this work, a new definition for optimal thrust distribution is proposed.

A prominent approach to the thrust allocation problem is the 2-norm (l_2 norm)-based solution in which the sum of the squares of the individual thruster forces is minimized. In Antonelli (2003), a pseudo-inverse solution, and in Fossen (1994), Sordalen (1997), and Omerdic and Roberts (2004), a weighted pseudo-inverse are used to generate an optimal distribution of a commanded generalized force. The pseudo-inverse method has the advantage of being relatively simple to compute. Pseudo-inverse solutions correspond to the minimization of either the l_2 norm or a weighted l_2 norm of the thrust manifold. However, the l_2 norm-based solution does not necessarily minimize the magnitudes of the individual thrusts, and can generate thrust demands that may exceed an individual thruster's saturation point. In addition, there could be an unequal distribution in the thrust manifold leading to a relatively high thrust demand for a particular thruster. In such cases, there exists a potential for a loss of manoeuvrability on subsequent control steps. The disadvantages of the l_2 norm-based optimization were reported for land-base manipulators by Arati and Walker (1997).

Furthermore, the pseudo-inverse method (l_2 norm minimization) does not afford easy implementation of thruster saturation limits (Omerdic and Roberts, 2004). It was reported by Durham (1993) that, even if thruster saturation is implemented, the pseudo-inverse solution is not guaranteed to satisfy the saturation constraints. The pseudo-inverse solution was also used for the thruster allocation problem by Sarkar et al. (2002). To generate reference thruster values that do not exceed the saturation limit of each thruster, Sarkar et al. (2002) employed a dynamic state feedback method.

In the current work, it is proposed that the complications associated with the l_2 norm-based solutions be avoided by using the infinity-norm (l_∞ norm) as the criterion in the thrust allocation. (The l_∞ norm is defined as the absolute value of the largest component of the thrust manifold). By using the l_∞ norm to gauge optimality of a thrust distribution, the largest single thrust in the distribution is minimized. The current work shows how the l_∞ norm thruster allocation can be cast as a constrained linear programming problem that allows direct implementation of thruster saturation limits and a fault-tolerant property. As pointed out by Sarkar et al. (2002), the allocation of thruster force problem in the presence of thruster faults and saturation limits for ROV systems has not been studied extensively. To obtain real-time computation rates for the linear programming problem, a recurrent neural network is proposed. An l_∞ norm-based thruster allocation has been preliminarily discussed by the authors of the current work in Soylu et al. (2007).

The efficacy of the proposed fault-accommodating thrust allocation scheme with the chattering-free sliding-mode control is demonstrated through numerical simulation studies on the ROPOS vehicle operated by the Canadian Scientific Submersible.

2. Dynamics and control

2.1. Dynamics of an ROV

The dynamic equations of motion of ROVs in the body-fixed frame can be represented as (Fossen, 1994)

$$\begin{aligned}&\mathbf{M}\dot{\boldsymbol{q}}+\mathbf{C}(\dot{\boldsymbol{q}})\boldsymbol{q}+\mathbf{D}(\dot{\boldsymbol{q}})\boldsymbol{q}+\mathbf{g}(\boldsymbol{\eta})=\boldsymbol{\tau}\\&\dot{\boldsymbol{\eta}}=\mathbf{J}(\boldsymbol{\eta})\boldsymbol{q}\end{aligned}\tag{1}$$

where $\boldsymbol{q}=[u\ \ v\ \ w\ \ p\ \ q\ \ r]^{\mathrm{T}}$ is the ROV spatial velocity state vector with respect to its body-fixed frame, and $\boldsymbol{\eta}=[x\ \ y\ \ z\ \ \phi\ \ \theta\ \ \psi]^{\mathrm{T}}$ is the position and orientation state vector with respect to the inertial frame. The coordinate systems considered are illustrated in Fig. 1.

In Eq. (1), the spatial transformation matrix between the inertial frame and the ROV's body-fixed frame can be defined through the Euler angle transformation (Fossen, 1994), denoted by $\mathbf{J}(\boldsymbol{\eta})\in\mathbb{R}^{6\times6}$. The term $\mathbf{M}(\boldsymbol{q})\in\mathbb{R}^{6\times6}$ is the inertia matrix including the added mass effects, $\mathbf{C}(\boldsymbol{q})\in\mathbb{R}^{6\times6}$ is the matrix of centrifugal and Coriolis terms, $\mathbf{D}(\boldsymbol{q})\in\mathbb{R}^{6\times6}$ is the drag matrix, $\mathbf{g}(\boldsymbol{\eta})\in\mathbb{R}^{6}$ is the vector of gravity and buoyancy forces and moments, and finally $\boldsymbol{\tau}\in\mathbb{R}^{6}$ is the control forces and moments acting on the ROV centre of mass.

Eq. (1) can be represented in the inertial reference frame as (Fossen, 1994)

$$\boldsymbol{f}=\mathbf{M}_\eta(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}}+\mathbf{C}_\eta(\boldsymbol{q},\boldsymbol{\eta})\dot{\boldsymbol{\eta}}+\mathbf{D}_\eta(\boldsymbol{q},\boldsymbol{\eta})\dot{\boldsymbol{\eta}}+\boldsymbol{g}_\eta(\boldsymbol{\eta})=\mathbf{J}^{-\mathrm{T}}\boldsymbol{\tau}\tag{2}$$

where $\mathbf{M}_\eta(\boldsymbol{\eta})=\mathbf{J}^{-\mathrm{T}}\mathbf{M}\mathbf{J}^{-1}$, $\mathbf{C}_\eta(\boldsymbol{q},\boldsymbol{\eta})=\mathbf{J}^{-\mathrm{T}}[\mathbf{C}-\mathbf{M}\mathbf{J}^{-1}\dot{\mathbf{J}}]\mathbf{J}^{-1}$, $\mathbf{D}_\eta(\boldsymbol{q},\boldsymbol{\eta})=\mathbf{J}^{-\mathrm{T}}\mathbf{D}\mathbf{J}^{-1}$ and $\boldsymbol{g}(\boldsymbol{\eta})=\mathbf{J}^{-\mathrm{T}}\boldsymbol{g}$. The dynamics of an ROV are assumed to have the following structural properties (Fossen, 1994):

Property 1. The inertia matrix $\mathbf{M}_\eta$ is symmetric and positive definite, i.e., $\mathbf{M}_\eta^{\mathrm{T}}=\mathbf{M}_\eta$;

Property 2. Matrix $\dot{\mathbf{M}}_\eta-2\mathbf{C}_\eta$ is skew symmetric, i.e., for any vector ζ, $\zeta^{\mathrm{T}}(\dot{\mathbf{M}}_\eta-2\mathbf{C}_\eta)\zeta=0$.

Eq. (2) can be written in a more compact form of

$$\boldsymbol{f}=\mathbf{M}_\eta(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}}+\boldsymbol{h}_\eta(\boldsymbol{q},\boldsymbol{\eta})\tag{3}$$

where $\boldsymbol{h}(\boldsymbol{q},\boldsymbol{\eta})=\mathbf{C}_\eta(\boldsymbol{q},\boldsymbol{\eta})\dot{\boldsymbol{\eta}}+\mathbf{D}_\eta(\boldsymbol{q},\boldsymbol{\eta})\dot{\boldsymbol{\eta}}+\boldsymbol{g}_\eta(\boldsymbol{\eta})$. As mentioned earlier, the ROV dynamics are dominated by hydrodynamic loads, and it is difficult to accurately measure or estimate the hydrodynamic coefficients that are valid for all ROV operating conditions and instrument and tether configurations. As such, the system dynamics are not exactly known. Therefore, the system dynamics given in Eq. (3) can be written as the sum of estimated dynamics $\hat{\boldsymbol{f}}$ and the unknown dynamics $\tilde{\boldsymbol{f}}$:

$$\boldsymbol{f}=\hat{\boldsymbol{f}}+\tilde{\boldsymbol{f}}\tag{4}$$

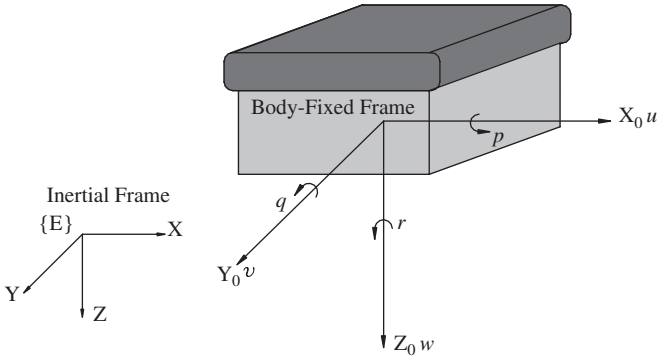

Fig. 1. Coordinate systems.

where the estimated dynamics vector is defined as

$$\hat{\boldsymbol{f}} = \hat{\mathbf{M}}_\eta(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}} + \hat{\boldsymbol{h}}(\boldsymbol{q},\boldsymbol{\eta}) \tag{5}$$

with $\hat{\boldsymbol{h}}_\eta(\boldsymbol{q},\boldsymbol{\eta}) = \hat{\mathbf{C}}_\eta(\boldsymbol{q},\boldsymbol{\eta})\dot{\boldsymbol{\eta}} + \hat{\mathbf{D}}_\eta(\boldsymbol{q},\boldsymbol{\eta})\dot{\boldsymbol{\eta}} + \hat{\boldsymbol{g}}_\eta(\boldsymbol{\eta})$ and the unknown dynamics vector are defined as

$$\tilde{\boldsymbol{f}} = \tilde{\mathbf{M}}_\eta(\boldsymbol{\eta})\ddot{\boldsymbol{\eta}} + \tilde{\boldsymbol{h}}_\eta(\boldsymbol{q},\boldsymbol{\eta}) + \boldsymbol{d} \tag{6}$$

with $\tilde{\boldsymbol{h}}_\eta(\dot{\boldsymbol{q}},\boldsymbol{\eta}) = \tilde{\mathbf{C}}_\eta(\dot{\boldsymbol{q}},\boldsymbol{\eta})\dot{\boldsymbol{\eta}} + \tilde{\mathbf{D}}_\eta(\dot{\boldsymbol{q}},\boldsymbol{\eta})\dot{\boldsymbol{\eta}} + \tilde{\boldsymbol{g}}_\eta(\boldsymbol{\eta})$, $\tilde{\mathbf{M}}_\eta = \mathbf{M}_\eta - \hat{\mathbf{M}}_\eta$, $\tilde{\mathbf{C}}_\eta = \mathbf{C}_\eta - \hat{\mathbf{C}}_\eta$, $\tilde{\mathbf{D}}_\eta = \mathbf{D}_\eta - \hat{\mathbf{D}}_\eta$, and $\tilde{\boldsymbol{g}}_\eta = \boldsymbol{g}_\eta - \hat{\boldsymbol{g}}_\eta$. Note that $\boldsymbol{d} \in \mathbb{R}^6$ is added as a disturbance force vector. The unknown dynamics vector is also called lumped uncertainty vector (Lin and Wai, 2002).

Assumption 1. Nonlinear lumped uncertainty vector $\tilde{\boldsymbol{f}}$ given in (6) and its time derivative are bounded.

2.2. Chattering-free adaptive sliding-mode controller design

2.2.1. Sliding-mode controller with a switching term

A controller design based on the sliding-mode methodology involves two main steps. The first step is to define the desired dynamics in the form of a vector of sliding manifolds $\boldsymbol{s} \in \mathbb{R}^6$. The second step is to find a control law $\boldsymbol{\tau} \in \mathbb{R}^6$ such that the system trajectories move toward the sliding manifold, and once they hit the manifold, remain on it in the presence of system uncertainties and disturbances (Slotine and Li, 1991).

For a second-order system such as an underwater vehicle system, reasonable desired dynamics would be a stable first-order system. This first-order dynamics can be defined as (Slotine and Li, 1991)

$$\boldsymbol{s} = \left(\frac{\mathrm{d}}{\mathrm{d}t} + \boldsymbol{\Lambda}\right)^2 \left(\int \boldsymbol{e}\,\mathrm{d}t\right) = \dot{\boldsymbol{e}} + 2\boldsymbol{\Lambda}\boldsymbol{e} + \boldsymbol{\Lambda}^2 \int \boldsymbol{e}\,\mathrm{d}t \tag{7}$$

where $\boldsymbol{\Lambda} \in \mathbb{R}^{6\times 6}$ is a constant, symmetric, positive definite and diagonal matrix that defines the break frequency of the desired error response. Each component of the sliding manifold represents a time-varying line in the state space that passes through the desired state variables. When $\boldsymbol{s} = 0$, the system states are on the surface meaning the system behaves consistently with the desired dynamics. This implies that the value of $\boldsymbol{s}$ indicates the extent of discrepancy between the desired state and the current state. Any sliding-mode-based controller works to keep the value of $\boldsymbol{s}$ at zero.

The tracking error between the measured state values and the desired state values is given by $\boldsymbol{e} = \boldsymbol{\eta} - \boldsymbol{\eta}_\mathrm{d}$ with the subscript d denoting the desired position and attitude of the ROV produced by a separate trajectory planner. The integral term ensures zero offset error. For notational simplicity, Eq. (7) can also be written as

$$\boldsymbol{s} = \dot{\boldsymbol{\eta}} - \dot{\boldsymbol{\eta}}_\mathrm{r} \tag{8}$$

where $\dot{\boldsymbol{\eta}}_\mathrm{r} = \dot{\boldsymbol{\eta}}_\mathrm{d} - 2\boldsymbol{\Lambda}\boldsymbol{e} - \boldsymbol{\Lambda}^2 \int \boldsymbol{e}\,\mathrm{d}t$, and the subscript r refers to virtual reference trajectory (Slotine and Li, 1991).

The standard sliding-mode control law is in the form

$$\boldsymbol{\tau} = \boldsymbol{\tau}_\mathrm{eq} + \boldsymbol{\tau}_\mathrm{sw} \tag{9}$$

where $\boldsymbol{\tau}$ corresponds to a generalized force acting at the centre of mass of the ROV, and $\boldsymbol{\tau}_\mathrm{eq}$ and $\boldsymbol{\tau}_\mathrm{sw}$ symbolize the equivalent control law and the switching control law, respectively.

The equivalent control law is continuous and model based. In the absence of uncertainties in the system dynamics, this equivalent control alone suffices to realize the desired dynamics. However, due to model uncertainties, an auxiliary switching term is needed that offsets the difference between the desired dynamics and real dynamics. This switching term is defined as $\boldsymbol{\tau}_\mathrm{sw} = -\mathbf{K}\,\mathrm{sgn}(\boldsymbol{s})$ in conventional sliding-mode control, where $\mathbf{K} \in \mathbb{R}^{6\times 6}$ is the positive definite diagonal gain matrix that is defined based on the upper bounds on the system parameter uncertainties, and $\mathrm{sgn}(\cdot)$ is the nonlinear signum function.

The switching term is a discontinuous feedback component that is in charge of compensating for deviations from the desired dynamics, and therefore is the source of the robustness of the sliding-mode control law. The switching term acts on the system in a bang–bang manner creating chatter in the actuators and causing the system state to oscillate intensely across the sliding manifold (Slotine and Li, 1991). The goal of this section is to replace the discontinuous switching term $\boldsymbol{\tau}_\mathrm{sw}$ with a continuous adaptive term $\boldsymbol{\tau}_\mathrm{ad}$ in an effort to eliminate the chattering problem.

2.2.2. Equivalent control law

The model-based equivalent control law component of the sliding-mode control signal can be derived by assuming that the motion is constrained to the sliding manifold. This implies that $\boldsymbol{s}$ is a constant vector, and thus $\dot{\boldsymbol{s}} = \mathbf{0}$. The time derivative of $\boldsymbol{s}$ can be defined based on Eq. (8), as

$$\dot{\boldsymbol{s}} = \ddot{\boldsymbol{\eta}} - \ddot{\boldsymbol{\eta}}_\mathrm{r} \tag{10}$$

where $\ddot{\boldsymbol{\eta}}_\mathrm{r} = \ddot{\boldsymbol{\eta}}_\mathrm{d} - 2\boldsymbol{\Lambda}\dot{\boldsymbol{e}} - \boldsymbol{\Lambda}^2\boldsymbol{e}$. Multiplying both sides of Eq. (10) by the inertia matrix $\hat{\mathbf{M}}_\eta$, and substituting $\mathbf{M}_\eta\ddot{\boldsymbol{\eta}} = \mathbf{J}^{-\mathrm{T}}\boldsymbol{\tau}_\mathrm{eq} - (\mathbf{C}_\eta\dot{\boldsymbol{\eta}} + \mathbf{D}_\eta\dot{\boldsymbol{\eta}} + \boldsymbol{g}_\eta)$ from Eq. (2) into the resulting equation yields

$$\hat{\mathbf{M}}_\eta\dot{\boldsymbol{s}} = \mathbf{J}^{-\mathrm{T}}\boldsymbol{\tau}_\mathrm{eq} - (\hat{\mathbf{M}}_\eta\ddot{\boldsymbol{\eta}}_\mathrm{r} + \hat{\mathbf{C}}_\eta\dot{\boldsymbol{\eta}} + \hat{\mathbf{D}}_\eta\dot{\boldsymbol{\eta}} + \hat{\boldsymbol{g}}_\eta). \tag{11}$$

Letting $\dot{\boldsymbol{s}} = 0$ and solving the resulting equation for $\boldsymbol{\tau}_\mathrm{eq}$ yield the equivalent control as

$$\boldsymbol{\tau}_\mathrm{eq} = \mathbf{J}^\mathrm{T}\hat{\boldsymbol{f}}_\mathrm{r}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}},\ddot{\boldsymbol{\eta}}_\mathrm{r}) \tag{12}$$

where

$$\hat{\boldsymbol{f}}_\mathrm{r}(\boldsymbol{\eta},\dot{\boldsymbol{\eta}},\ddot{\boldsymbol{\eta}}_\mathrm{r}) = \hat{\mathbf{M}}_\eta\ddot{\boldsymbol{\eta}}_\mathrm{r} + \hat{\mathbf{C}}_\eta\dot{\boldsymbol{\eta}} + \hat{\mathbf{D}}_\eta\dot{\boldsymbol{\eta}} + \hat{\boldsymbol{g}}_\eta \tag{13}$$

As mentioned before, in the absence of uncertainties in the system dynamics, this equivalent control alone can keep the state variables on the sliding surface.

2.2.3. Adaptive control law

With regard to the replacement of the discontinuous term, the following continuous adaptive control law is proposed in place of the switching term:

$$\boldsymbol{\tau}_\mathrm{ad} = \mathbf{J}^\mathrm{T}(\tilde{\boldsymbol{f}}_\mathrm{est} - (\mathbf{K} + \hat{\mathbf{C}}_\eta)\boldsymbol{s}) \tag{14}$$

where $\tilde{\boldsymbol{f}}_\mathrm{est}$ is an adaptive term that estimates the lumped uncertainty vector defined in Eq. (6), and $\mathbf{K} \in \mathbb{R}^{6\times 6}$ is a diagonal positive definite constant matrix that is related to the convergence rate of the controller. The estimation of the lumped uncertainty vector is proposed to follow:

$$\dot{\tilde{\boldsymbol{f}}}_\mathrm{est} = -\boldsymbol{\Gamma}\boldsymbol{s} \tag{15}$$

where $\boldsymbol{\Gamma} \in \mathbb{R}^{6\times 6}$ is a positive definite diagonal constant design matrix that determines the rate of adaptation. This adaptive term relates the error metric $\boldsymbol{s}$ function to the dynamic uncertainties, and acts on the controller in such a way that the estimated dynamics reflect the unknown dynamics more closely to the actual dynamics. The assumption of $\tilde{\boldsymbol{f}}$ being bounded ensures that Eq. (15) is bounded as well.

The total control input $\boldsymbol{\tau}$ is defined as the sum of the equivalent control signal and the adaptive control signal, and is given by

$$\begin{aligned}\boldsymbol{\tau} &= \boldsymbol{\tau}_\mathrm{eq} + \boldsymbol{\tau}_\mathrm{ad} \\ &= \mathbf{J}^\mathrm{T}(\hat{\boldsymbol{f}}_\mathrm{r} + \tilde{\boldsymbol{f}}_\mathrm{est} - (\mathbf{K} + \hat{\mathbf{C}}_\eta)\boldsymbol{s})\end{aligned} \tag{16}$$

Assumption 2. The following inequality is assumed to hold:

$$\boldsymbol{s}^\mathrm{T}\mathbf{K}\boldsymbol{s} \geqslant |\dot{\tilde{\boldsymbol{f}}}^\mathrm{T}\boldsymbol{\Gamma}^{-1}\boldsymbol{w}| \quad \text{only when} \quad \dot{\tilde{\boldsymbol{f}}}^\mathrm{T}\boldsymbol{\Gamma}^{-1}\boldsymbol{w} < 0 \tag{17}$$

where $\boldsymbol{w} = \tilde{\boldsymbol{f}}_{\text{est}} - \tilde{\boldsymbol{f}}$ is the difference vector between the estimated lumped uncertainty vector and the exact lumped uncertainty vector.

Remark. Under slow ROV motion assumptions, $\dot{\tilde{\boldsymbol{f}}}$ is relatively small. Also, the term $\boldsymbol{w}$ becomes smaller as the adaptive term acts on the system. Therefore, Assumption 2 is rather realistic. In the worst case scenario, Assumption 2 can be strengthened by increasing the $\mathbf{K}$ and $\boldsymbol{\Gamma}$ entries.

Theorem 1. *Consider the nonlinear dynamical system described by Eq. (2) with Assumptions 1 and 2. If the control law is expressed as Eq. (16), then stability of the closed-loop control system is guaranteed.*

Proof. See Appendix A.

The general scheme of the control law is illustrated in Fig. 2. The control law is given by Eq. (16) and the update law for the adaptive term is given by Eq. (15). The novelty of Eq. (16) is that it replaces the discontinuous term with a term, $\tilde{\boldsymbol{f}}_{\text{est}}$. This term continuously acts on the model-based equivalent control term and works to reduce the discrepancy between the known and the exact dynamics at each sampling time. As the discrepancy is reduced, the state variables stay on the sliding manifold once they are driven on to it.

When the boundary layer approach is used, the robustness property of the sliding-mode control is compromised for the sake of chattering elimination since sliding-mode controller acts similar to PD controllers within the boundary layers. The given adaptive term, however, preserves this robustness property since the unknown dynamics are estimated and added to the equivalent control signal as a corrective term. In addition, as opposed to conventional adaptive control theory, no parameterization of a regressor matrix and unknown parameter vector needs to be found. The regressor matrix is so complex that further assumptions are necessary in the system model leading to further degradation in the quality of the modelled system dynamics (Antonelli et al., 2001). The proposed method does not require this linearization since it estimates the complete accumulation of unknown dynamics. In this regard, the proposed adaptive term blends the advantages of the sliding and adaptive control while eliminating their disadvantages.

3. Infinity-norm thrust allocation

The fault-tolerant, l_∞ norm-based thrust allocation scheme will be formulated in this section.

3.1. Thruster force model

The generalized force vector $\boldsymbol{\tau}_i \in \mathbb{R}^{6\times 1}$ produced by the ith thruster can be expressed as (Omerdic and Roberts, 2004)

$$\boldsymbol{\tau}_i = \begin{bmatrix} \boldsymbol{f}_i \\ \boldsymbol{n}_i \end{bmatrix} = \begin{bmatrix} p_i \boldsymbol{e}_i \\ p_i(\boldsymbol{r}_i \times \boldsymbol{e}_i) \end{bmatrix} = \begin{bmatrix} e_{i,x} \\ e_{i,y} \\ e_{i,z} \\ (r_i \times e_i)_x \\ (r_i \times e_i)_y \\ (r_i \times e_i)_z \end{bmatrix} p_i \tag{18}$$

where p_i is the thrust exerted by the ith thruster, and $\boldsymbol{f}_i$ and $\boldsymbol{n}_i$ are the force and moment vectors associated with p_i. The vector $\boldsymbol{e}_i = [e_{i,x} \quad e_{i,y} \quad e_{i,z}]^{\mathrm{T}}$ in Eq. (18) indicates the positive direction of the thrust p_i and the position vector $\boldsymbol{r} = [r_{i,x} \quad r_{i,y} \quad r_{i,z}]^{\mathrm{T}}$ determines the position of each thruster with respect to the ROV centre of mass as illustrated in Fig. 4.

The control input $\boldsymbol{\tau}_i \in \mathbb{R}^{6\times 1}$ calculated in Eq. (16) is realized by superposing the individual generalized force vectors of each thruster:

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_X \\ \tau_Y \\ \tau_Z \\ \tau_K \\ \tau_M \\ \tau_N \end{bmatrix} = \sum_{i=1}^{n} \boldsymbol{\tau}_i = \sum_{i=1}^{n} \begin{bmatrix} e_i \\ (r_i \times e_i) \end{bmatrix} p_i = \mathbf{E}\boldsymbol{p} \tag{19}$$

where n is the number of thrusters, and $\boldsymbol{p} \in \mathbb{R}^n$ is the collection of individual thrusts, hereafter called the thrust manifold. The thruster configuration matrix $\mathbf{E} \in \mathbb{R}^{6\times n}$ captures the geometry of the thruster layout, and transforms the individual thruster forces into generalized forces experienced at the ROV mass centre.

3.2. Potential advantages of l_∞ norm over l_2 norm

In this paper, it is proposed to minimize the l_∞ norm of the thrust manifold rather than the conventional method of minimizing the l_2 norm. The l_∞ norm provides the exact representation of the feasible thrust solution space whereas the l_2 norm provides an approximation of the feasible solution space. This suggests the l_∞ norm always finds a feasible solution as long as a solution exists within the thrusters' saturation limits whereas the l_2 norm does not. To demonstrate this, consider an ROV with two onboard thrusters whose limits are normalized such that $p_{n,i} = \pm 1$ N, for $i = 1, 2$. In the l_2 norm case, the feasible thrust solutions are enclosed by a circle (an n-dimensional sphere in $\mathbb{R}^n$), i.e., $||\boldsymbol{p}||_2 = (p_1^2 + p_2^2)^{1/2} \leqslant 1$, and the corresponding task space points are enclosed by an ellipse (an n-dimensional ellipsoid in $\mathbb{R}^n$), which is obtained through the

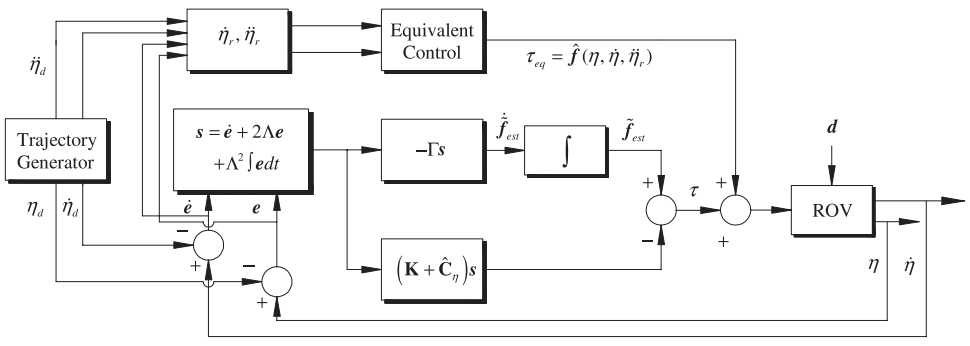

Fig. 2. Adaptive sliding-mode controller.

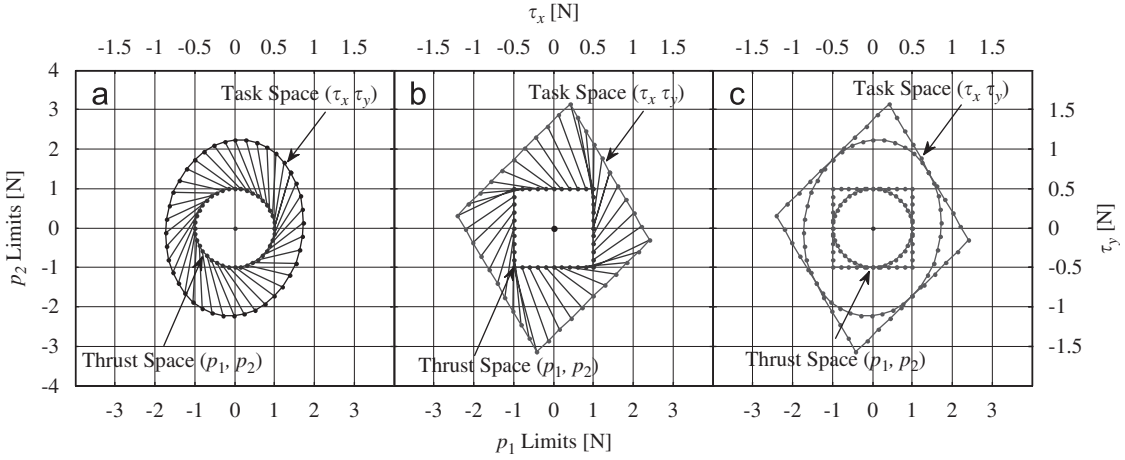

Fig. 3. The lines that connect inner and outer shapes represent the linear transformations.

linear transformation according to $\boldsymbol{\tau} = \mathbf{E}\boldsymbol{p}$ as illustrated in Fig. 3a. In the l_∞ norm case, however, the feasible thrust space points are enclosed by a square $||\boldsymbol{p}||_\infty = \max\{|p_1|, |p_2|\} \leqslant 1$ (an n-dimensional parallelepiped in $\mathbb{R}^n$), and the corresponding task space is enclosed by a polygon (a polytope in $\mathbb{R}^n$) obtained from the same linear transformation as illustrated in Fig. 3b. Now consider a feasible solution point $p_1 = p_2 = 1$ which corresponds to the upper right vertex of the square. While this feasible point is contained by the square, it is excluded by the circle rendering the otherwise feasible point infeasible. Similarly, the associated value of $p_1 = p_2 = 1$ in the task space (a vertex in the polygon) is not contained by the ellipse. The unfeasible regions of the 2-norm solution are illustrated in Fig. 3c where the circle and ellipse are inscribed within the square and polygon, respectively.

To extend this to an example, consider an ROV that is to be driven in horizontal motion (*X*-*Y* plane) with four thrusters. The thruster saturation limit for each thruster is ± 900 N. Assume that a generalized force vector of $\boldsymbol{\tau} = [1800 \quad 0 \quad 0; \quad 0 \quad 0 \quad 2742]^{\mathrm{T}}$ N; N m at the ROV centre of mass is commanded by the on-board controller. If the solution to $\boldsymbol{\tau} = \mathbf{E}\boldsymbol{p}$ is realized based on the minimization of the l_∞ norm, then the thrust vector of $\boldsymbol{p}_{l_\infty} = [900 \quad 900 \quad 900 \quad -900]^{\mathrm{T}}$ N is obtained. This corresponds to a vertex point on the polytope containing all the feasible points. However, if the distribution is realized based on the minimization of the l_2 norm, then the thrust vector of $\boldsymbol{p}_{l_2} = [1292 \quad 671 \quad 508 \quad -671]^{\mathrm{T}}$ N is obtained. This corresponds to a point outside the hypersphere formed by the approximation of the feasible points, and therefore, it is infeasible. This example shows that there exist solutions where minimizing l_∞ norm gives feasible solutions whereas minimizing l_2 norm does not. The practical outcome of this could be disastrous since the first component of $\boldsymbol{p}_{l_2}$ is truncated to the upper limit at the first thruster, the solution vector no longer satisfies $\boldsymbol{\tau} = \mathbf{E}\boldsymbol{p}$. As a consequence, the controller demand cannot be realized, and the controller fails. A similar discussion for the force capability of parallel manipulators is presented by Firmani et al. (2007). In general, since the l_∞ norm minimizes the maximum component of the thrust manifold, it allows the thrusters to run within a safer range than the l_2 norm providing more manoeuvrability for subsequent controller action.

3.3. Infinity-norm formulation

The l_∞ norm of the thruster force manifold $\boldsymbol{p} = [p_1 \quad p_2 \quad \cdots \quad p_n]^{\mathrm{T}}$ is defined as

$$||\boldsymbol{p}||_\infty = \max\{|p_1| \quad |p_2| \quad \cdots \quad |p_n|\} = \max_{1 \leqslant i \leqslant n} \left|\mathbf{I}_i^{\mathrm{T}} \boldsymbol{p}\right| \tag{20}$$

where $|\cdot|$ is the absolute value of each component of the thrust manifold, and $\mathbf{I}_i$ is the ith column vector of the identity matrix $\mathbf{I} \in \mathbb{R}^{n \times n}$. Taking into account the thruster saturation limits, the bounded l_∞ norm thrust allocation problem can be cast as the following constrained optimization problem:

$$\begin{aligned} &\text{minimize} \quad ||\boldsymbol{p}||_\infty \\ &\text{subject to} \quad \boldsymbol{\tau} = \boldsymbol{E}\boldsymbol{p}, \quad p_{i,l} \leqslant p_i \leqslant p_{i,u}, \quad i = 1, \ldots, n \end{aligned} \tag{21}$$

where $p_{i,\,l}$ and $p_{i,\,u}$ are the lower and upper thruster saturation limits, respectively. Redefining the objective function as

$$l = ||\boldsymbol{p}||_\infty = \max_{1 \leqslant i \leqslant n} \left|\mathbf{I}_i^{\mathrm{T}} \boldsymbol{p}\right| \tag{22}$$

leads to the following equivalent linear programming problem:

$$\begin{aligned} &\text{minimize} \quad l \\ &\text{subject to} \quad \max_{1 \leqslant i \leqslant n} \left|\mathbf{I}_i^{\mathrm{T}} \boldsymbol{p}\right| \leqslant l, \\ &\qquad\qquad \boldsymbol{\tau} = \mathbf{E}\boldsymbol{p}, \quad p_{i,l} \leqslant p_i \leqslant p_{i,u}, \quad i = 1, \ldots, n. \end{aligned} \tag{23}$$

The linear problem of Eq. (23) can be written in the following compact matrix form:

$$\begin{aligned} &\text{minimize} \quad \boldsymbol{c}^{\mathrm{T}} \boldsymbol{y} \\ &\text{subject to} \quad \mathbf{A}_1 \boldsymbol{y} \geqslant \boldsymbol{b}_1, \quad \mathbf{A}_2 \boldsymbol{y} = \boldsymbol{b}_2 \end{aligned} \tag{24}$$

where

$$\mathbf{A}_1 = \begin{bmatrix} -\mathbf{I} & \boldsymbol{I}_n \\ \mathbf{I} & \boldsymbol{I}_n \\ -\mathbf{I} & \boldsymbol{0}_n \\ \mathbf{I} & \boldsymbol{0}_n \end{bmatrix} \in \mathbb{R}^{4n \times (n+1)}, \quad \boldsymbol{b}_1 = \begin{bmatrix} \boldsymbol{0}_n \\ \boldsymbol{0}_n \\ -\boldsymbol{p}_u \\ \boldsymbol{p}_l \end{bmatrix} \in \mathbb{R}^{4n},$$

$$\mathbf{A}_2 = [\mathbf{E} \quad \boldsymbol{0}_m] \in \mathbb{R}^{m \times (n+1)}, \quad \boldsymbol{b}_2 = \boldsymbol{\tau} \in \mathbb{R}^m,$$

$$\boldsymbol{c} = [0 \quad 0 \quad \cdots \quad 0 \quad 1]^{\mathrm{T}} \in \mathbb{R}^{n+1}, \quad \boldsymbol{y} = [\boldsymbol{p} \quad l]^{\mathrm{T}} \in \mathbb{R}^{n+1}$$

with $\boldsymbol{I}_n = [1 \quad 1 \quad \cdots \quad 1]^{\mathrm{T}} \in \mathbb{R}^n$ and $\mathbf{I} \in \mathbb{R}^{n \times n}$ denoting the identity matrix. The term $\boldsymbol{p}_u = [p_{1,u} \quad p_{2,u} \quad \cdots \quad p_{n,u}]^{\mathrm{T}} \in \mathbb{R}^n$ is the upper saturation vector, and the term $\boldsymbol{p}_l = [p_{1,l} \quad p_{2,l} \cdots p_{n,l}]^{\mathrm{T}} \in \mathbb{R}^n$ is the lower saturation vector with $\boldsymbol{0}_n \in \mathbb{R}^n$ and $\boldsymbol{0}_m \in \mathbb{R}^m$ being the zero column vectors.

3.4. Fault-tolerant property

In the current work, joint failure refers to two incidents: first, when a thruster breaks down; second, when a thruster partly loses its driving capacity. A general fault-tolerant scheme detects faulty thrusters, and then applies a strategy in which the

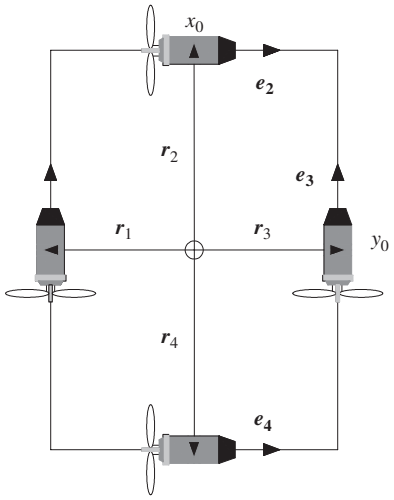

Fig. 4. The redundant thruster layout.

demanded controller output can be reallocated among the functioning thrusters.

In order to account for the faulty thrusters, the linear programming problem given in Eq. (24) must be modified. To this end, a diagonal weighting matrix can be used. The diagonal elements of the weighting matrix penalize the upper and lower saturation limits of faulty thrusters such that demanded thrust does not exceed the available capacity of the faulty thrusters. The diagonal weighting matrix can be defined as

$$\mathbf{W} = \operatorname{diag}(w_1 \quad w_2 \quad \cdots \quad w_n) \in \mathbb{R}^{n\times n}$$
$$w_i = \begin{cases} 0 & \text{if the } i\text{th thruster is in failure} \\ 0<w_i<1 & \text{if the } i\text{th thruster is in partial failure;} \quad i=1,\ldots,n. \\ 1 & \text{if the } i\text{th thruster is not in failure} \end{cases} \tag{25}$$

The fault-tolerant property is incorporated into the optimization formulation by redefining the vector $\boldsymbol{b}_1$ as

$$\bar{\boldsymbol{b}}_1 = \begin{bmatrix} \boldsymbol{0}_n & \boldsymbol{0}_n & -\bar{\boldsymbol{p}}_u & \bar{\boldsymbol{p}}_l \end{bmatrix}^{\mathrm{T}} \tag{26}$$

where

$$\bar{\boldsymbol{p}}_u = \mathbf{W}\boldsymbol{p}_u \quad \text{and} \quad \bar{\boldsymbol{p}}_l = \mathbf{W}\boldsymbol{p}_l.$$

Finally, the fault-tolerant, bounded l_∞ norm thrust allocation problem is formulated as

$$\begin{aligned} &\text{minimize} && \boldsymbol{c}^{\mathrm{T}}\boldsymbol{y} \\ &\text{subject to} && \mathbf{A}_1\boldsymbol{y} \geqslant \bar{\boldsymbol{b}}_1, \ \mathbf{A}_2\boldsymbol{y} = \boldsymbol{b}_2 \end{aligned} \tag{27}$$

3.5. Primal-dual problem

The linear problem of Eq. (27) can be transformed into an equivalent optimization problem in terms of its Lagrange multipliers. In this context, the transformed problem is called the *dual* problem whereas the original problem is referred to as the *primal* problem (Wolfe, 1961). In view of this, the linear programming problem of Eq. (27) will be called the primal problem hereafter.

Theorem 2. *Let $\boldsymbol{y}^* \in \mathbb{R}^n$ be the solution of the primal problem given in Eq. (27). Consider $\boldsymbol{\mu} \in \mathbb{R}^{4n}$ and $\boldsymbol{\lambda} \in \mathbb{R}^m$ to be the associated Lagrange multipliers (also called dual decision variables) related to the inequality and equality constraints of the primal problem. If $\boldsymbol{y}^*$ is a regular point of the constraints (i.e., it satisfies the equality constraints and A2 has full row rank), then $\boldsymbol{y}^*$, $\boldsymbol{\mu}^*$ and $\boldsymbol{\lambda}^*$ solve the dual problem:*

$$\begin{aligned} &\text{maximize} && L(\boldsymbol{y},\boldsymbol{\mu},\boldsymbol{\lambda}) \\ &\text{subject to} && \nabla_y L(\boldsymbol{y},\boldsymbol{\mu},\boldsymbol{\lambda}) = 0 \\ & && \boldsymbol{\mu} \geqslant 0 \end{aligned} \tag{28}$$

where $L(\boldsymbol{y}, \boldsymbol{\mu}, \boldsymbol{\lambda})$ is the Lagrangian function and $\nabla_y L(\boldsymbol{y}, \boldsymbol{\mu}, \boldsymbol{\lambda})$ is the gradient of the Lagrangian function. In addition, $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{y}^* = L(\boldsymbol{y}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$.

Proof. See Antoniou and Lu (2007).

According to Theorem 2, the objective functions of the primal and dual problems reach to the same value at their solution point. The difference is called duality gap, i.e., $\boldsymbol{c}^{\mathrm{T}}\boldsymbol{y}-L(\boldsymbol{y}, \boldsymbol{\mu}, \boldsymbol{\lambda})$, which will be used in Section 4.2 in the derivation of an energy function.

In an attempt to obtain the dual problem of the primal problem, the Lagrangian function of Eq. (27), $L(\boldsymbol{y}, \boldsymbol{\mu}, \boldsymbol{\lambda})$, is defined as

$$L(\boldsymbol{y},\boldsymbol{\mu},\boldsymbol{\lambda}) = \boldsymbol{y}^{\mathrm{T}}(\boldsymbol{c} - \mathbf{A}_1^{\mathrm{T}}\boldsymbol{\mu} - \mathbf{A}_2^{\mathrm{T}}\boldsymbol{\lambda}) + \bar{\boldsymbol{b}}_1^{\mathrm{T}}\boldsymbol{\mu} + \boldsymbol{b}_2^{\mathrm{T}}\boldsymbol{\lambda} \tag{29}$$

The gradient of Eq. (29) with respect to the primal variable $\boldsymbol{y}$ is

$$\nabla_y L(\boldsymbol{y},\boldsymbol{\mu},\boldsymbol{\lambda}) = \boldsymbol{c} - \mathbf{A}_1^{\mathrm{T}}\boldsymbol{\mu} - \mathbf{A}_2^{\mathrm{T}}\boldsymbol{\lambda} \tag{30}$$

Applying Theorem 2 to Eq. (27) with the definitions of Eqs. (29) and (30) yields the dual problem as

$$\begin{aligned} &\text{maximize} && \boldsymbol{y}^{\mathrm{T}}(\boldsymbol{c} - \mathbf{A}_1^{\mathrm{T}}\boldsymbol{\mu} - \mathbf{A}_2^{\mathrm{T}}\boldsymbol{\lambda}) + \bar{\boldsymbol{b}}_1^{\mathrm{T}}\boldsymbol{\mu} + \boldsymbol{b}_2^{\mathrm{T}}\boldsymbol{\lambda} \\ &\text{subject to} && \boldsymbol{c} - \mathbf{A}_1^{\mathrm{T}}\boldsymbol{\mu} - \mathbf{A}_2^{\mathrm{T}}\boldsymbol{\lambda} = 0 \\ & && \boldsymbol{\mu} \geqslant 0 \end{aligned} \tag{31}$$

Using the equality constraint directly within the objective function, Eq. (31) can be simplified to

$$\begin{aligned} &\text{maximize} && \bar{\boldsymbol{b}}_1^{\mathrm{T}}\boldsymbol{\mu} + \boldsymbol{b}_2^{\mathrm{T}}\boldsymbol{\lambda} \\ &\text{subject to} && \mathbf{A}_1^{\mathrm{T}}\boldsymbol{\mu} + \mathbf{A}_2^{\mathrm{T}}\boldsymbol{\lambda} - \boldsymbol{c} = 0 \\ & && \boldsymbol{\mu} \geqslant 0, \quad \boldsymbol{\lambda} \text{ unrestricted} \end{aligned} \tag{32}$$

Eq. (32) corresponds to the dual problem of the primal problem.

4. Solution with recurrent neural network

4.1. Neural network design preliminaries

The solution of the primal problem, Eq. (27), and the dual problem, Eq. (32), is the vector $\boldsymbol{y}$ containing the thrust manifold $\boldsymbol{p}$. Since real-time updates to the thrust manifold are required, the linear programme in the previous section must be solved in real time. However, the existing sequential algorithms such as simplex methods (Gill et al., 1991) and interior point methods (Wright, 1992) are often not capable of delivering solutions at this rate. One promising approach to obtaining real-time solutions is a recurrent neural network on the basis of an analogue circuit. A neural network can satisfy real-time requirements due to its fast parallel processing nature. In addition, as long as there is a solution to the primal problem, a properly designed recurrent neural network will converge to the optimal solution without the requirement of a feasible initial point.

In what follows, a recurrent neural network will be designed to solve the primal problem following the gradient model derivation outlined by Xia and Wang (1998). The premise of the gradient model derivation is to design a dynamic system whose state variables naturally converge to the solution of the optimization problems of Eqs. (27) and (32).

4.2. Energy function derivation

In an attempt to design a recurrent neural network, an energy function can be defined using the synergy between the primal problem and its dual problem. In view of this, the following continuously differentiable, convex, positive definite, energy-like function $E(\boldsymbol{y}, \boldsymbol{\mu}, \boldsymbol{\lambda})$ is proposed for the neural network design:

$$E = \tfrac{1}{2}(\boldsymbol{c}^{\mathrm{T}}\boldsymbol{y} - \bar{\boldsymbol{b}}_1^{\mathrm{T}}\boldsymbol{\mu} - \boldsymbol{b}_2^{\mathrm{T}}\boldsymbol{\lambda})^2 + \tfrac{1}{2}\left\|\mathbf{A}_2\boldsymbol{y} - \boldsymbol{b}_2\right\|_2^2 + \tfrac{1}{2}\left\|\mathbf{A}_1^{\mathrm{T}}\boldsymbol{\mu} + \mathbf{A}_2^{\mathrm{T}}\boldsymbol{\lambda} - \boldsymbol{c}\right\|_2^2 + \tfrac{1}{4}(\mathbf{A}_1\boldsymbol{y} - \bar{\boldsymbol{b}}_1)^{\mathrm{T}}(\mathbf{A}_1\boldsymbol{y} - \bar{\boldsymbol{b}}_1 - |\mathbf{A}_1\boldsymbol{y} - \bar{\boldsymbol{b}}_1|) + \tfrac{1}{4}\boldsymbol{\mu}^{\mathrm{T}}(\boldsymbol{\mu} - |\boldsymbol{\mu}|) \quad (33)$$

The first term in Eq. (33) is the squared difference (duality gap) between the objective functions of the primal and dual problems. The second and the third terms are the equality constraints associated with the primal and dual problems. Finally, the last two terms are associated with the inequality constraints of the same primal and dual optimization problems.

4.3. Gradient model derivation

Since (33) is a convex function whose minimum is at zero, its solution corresponds to the solution of the following optimization problem (Xia, 1996; Xia and Wang, 1998):

$$\begin{aligned} &\text{minimize} \quad E(\boldsymbol{v}) \\ &\text{subject to} \quad \boldsymbol{v} \in \mathbb{R}^k \end{aligned} \quad (34)$$

where $\boldsymbol{v} = [\boldsymbol{y} \quad \boldsymbol{\mu} \quad \boldsymbol{\lambda}]^{\mathrm{T}}$.

To transform the minimization problem (34) into an associated system of ordinary differential equations that govern the recurrent neural network, the gradient of the energy-like function can be used:

$$\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}t} = -\kappa \nabla E(\boldsymbol{v}) \quad (35)$$

and $\kappa \in \mathbb{R}$ is a strictly positive number defining the convergence rate of the optimization problem of (34) (i.e., greater values provide faster convergence rates), and $\nabla E(\boldsymbol{v})$ is the gradient of the energy-like function. The equations given in Eq. (35) can be written more explicitly as

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{y}}{\mathrm{d}t} &= -\kappa \nabla_y E(\boldsymbol{v}) = -\kappa[\boldsymbol{c}(\boldsymbol{c}^{\mathrm{T}}\boldsymbol{y} - \bar{\boldsymbol{b}}_1^{\mathrm{T}}\boldsymbol{\mu} - \boldsymbol{b}_2^{\mathrm{T}}\boldsymbol{\lambda}) \\ &\quad + \mathbf{A}_2^{\mathrm{T}}(\mathbf{A}_2\boldsymbol{y} - \boldsymbol{b}_2) + \mathbf{A}_1^{\mathrm{T}}(\mathbf{A}_1\boldsymbol{y} - \bar{\boldsymbol{b}}_1)^-] \\ \frac{\mathrm{d}\boldsymbol{\mu}}{\mathrm{d}t} &= -\kappa \nabla_\mu E(\boldsymbol{v}) = -\kappa[-\bar{\boldsymbol{b}}_1(\boldsymbol{c}^{\mathrm{T}}\boldsymbol{y} - \bar{\boldsymbol{b}}_1^{\mathrm{T}}\boldsymbol{\mu} - \boldsymbol{b}_2^{\mathrm{T}}\boldsymbol{\lambda}) \\ &\quad + \mathbf{A}_1(\mathbf{A}_1^{\mathrm{T}}\boldsymbol{\mu} + \mathbf{A}_2^{\mathrm{T}}\boldsymbol{\lambda} - \boldsymbol{c}) + \boldsymbol{\mu}^-] \\ \frac{\mathrm{d}\boldsymbol{\lambda}}{\mathrm{d}t} &= -\kappa \nabla_\lambda E(\boldsymbol{v}) = -\kappa[-\boldsymbol{b}_2(\boldsymbol{c}^{\mathrm{T}}\boldsymbol{y} - \bar{\boldsymbol{b}}_1^{\mathrm{T}}\boldsymbol{\mu} - \boldsymbol{b}_2^{\mathrm{T}}\boldsymbol{\lambda}) \\ &\quad + \mathbf{A}_2(\mathbf{A}_1^{\mathrm{T}}\boldsymbol{\mu} + \mathbf{A}_2^{\mathrm{T}}\boldsymbol{\lambda} - \boldsymbol{c})] \end{aligned} \quad (36)$$

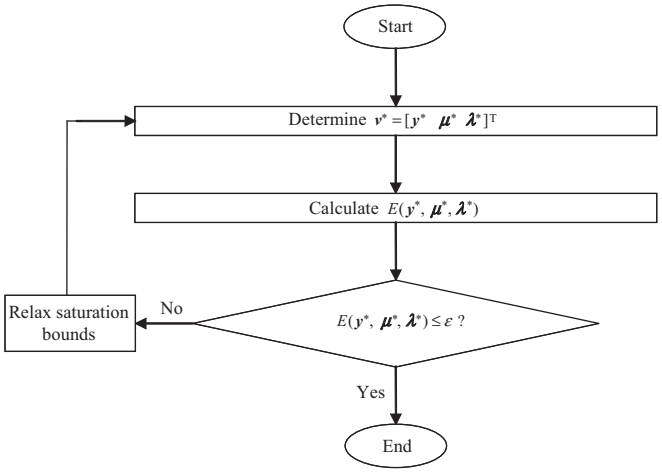

Fig. 6. Flow chart of the solution determination.

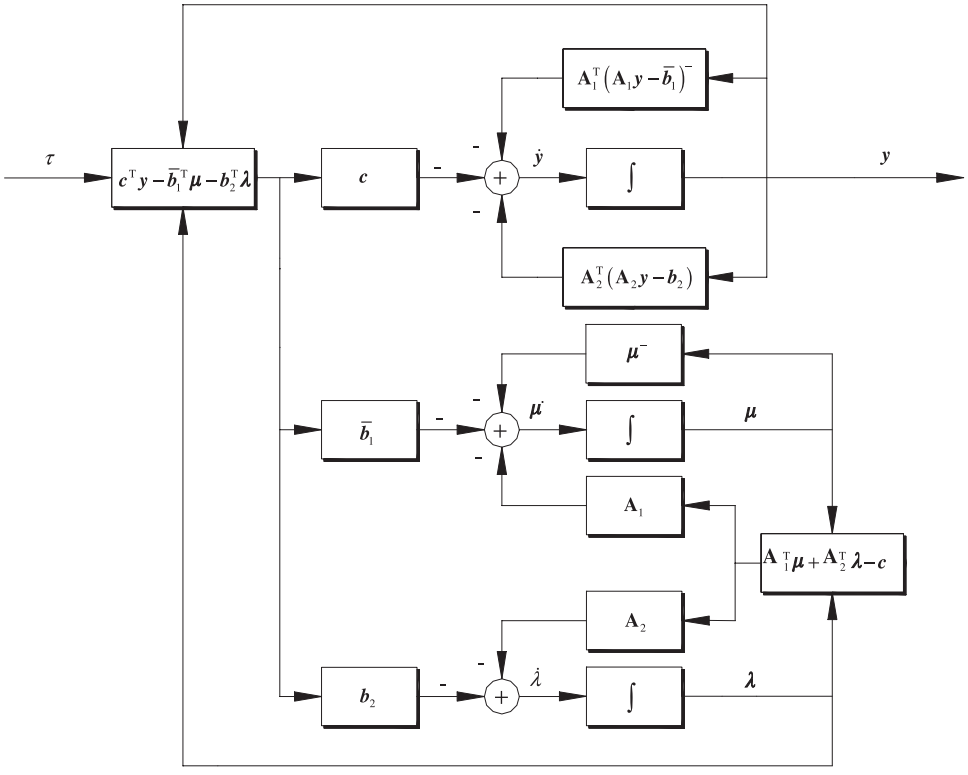

Fig. 5. Block diagram for the architecture of the neural network for the l_∞ norm thrust allocation.

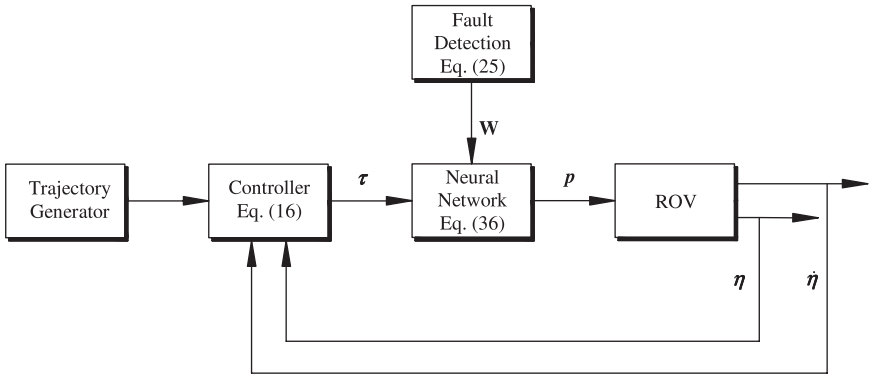

Fig. 7. Block diagram for the controller-neural network scheme.

Table 1
ROPOS hydrodynamic parameters

Coefficients	Surge	Lateral	Heave	Roll	Pitch	Yaw
Linear drag	725 N s/m	1240 N s/m	825 N s/m	3000 N s/m	3000 N s/m	1804 N s/m
Quad. drag	1000 N s^2/m^2	525 N s^2/m^2	400 N s^2/m^2	100 N s^2	100 N s^2	72 N s^2
Added mass	−4380 kg	−9518 kg	−4268 kg	−5000 kg m^2	−5000 kg m^2	−5000 kg m^2

where for any column vector $\boldsymbol{\zeta}$, the symbol $\boldsymbol{\zeta} - |\boldsymbol{\zeta}| = 2(\boldsymbol{\zeta})^-$ with $(\boldsymbol{\zeta})^- = \begin{bmatrix} \zeta_1^- & \zeta_2^- & \cdots & \zeta_n^- \end{bmatrix}^T$ and $\zeta_i^- = \min\{0 \quad \zeta_i\}$. Fig. 5 illustrates the block diagram for the architecture of the recurrent neural network defined in Eq. (36).

Theorem 3. *The neural network given in Eq. (36) is globally stable and will converge to the optimal solutions of* $\boldsymbol{v}^* = \begin{bmatrix} \boldsymbol{y}^{*T} & \boldsymbol{\mu}^{*T} & \boldsymbol{\lambda}^{*T} \end{bmatrix}^T$ *that corresponds to the solution of the primal problem, Eq. (27), and its dual problem, Eq. (32), respectively.*

Proof. See Appendix B.

Note that neural network design challenges such as the generation of training data, the validation of the trained neural network, the determination of type of activation functions, the number of neurons, etc. are eliminated through the hardware implementation of the proposed recurrent neural network, and are not addressed in the current work. For the hardware implementation, neural network dynamic equations are derived by the current research. A detailed discussion regarding the creation of an integrated circuit from neural network dynamic equations is given by Wang (1993).

4.4. Nonexistence of solution

In practice, there are two cases where there is no solution. First case is when there are too many faulty thrusters and the remaining thruster forces can no longer span the task space. Second case is when a solution does not exist within the saturation limits. With regards to the second case, the energy function given in Eq. (33) can be used to determine whether or not the neural network converged to the true solution; if the energy function is not within an allowable tolerance, i.e., $E(\boldsymbol{y}^*, \boldsymbol{\mu}^*, \boldsymbol{\lambda}^*) > \varepsilon$, where ε is the allowable tolerance and is set to nearly zero, then it can be concluded that the inequality constraints of the primal problem could not be satisfied. In this case, a heuristic strategy is applied to generate the best approximate solution to the thrust allocation: the upper and lower saturation limits are extended which translates into relaxing the solution range on $\boldsymbol{p}$. The relaxed problem is solved using the same neural network. Finally, the thrusts p_i within the relaxed solution that do exceed saturation limits are truncated to their actual upper or lower saturation limits, and the resulting thrust manifold $\boldsymbol{p}$ is applied to the ROV. This process is outlined in Fig. 6. A similar truncation approach was applied by Bordignon (1996).

Table 2
ROPOS inertia parameters

Mass (kg)	I_x (N m s^2)	I_y (N m s^2)	I_z (N m s^2)	Ixy (N m s^2)	Iyz (N m s^2)	Ixz (N m s^2)
2268	1937	2883	2457	0	0	0

Table 3
Controller parameters

$\boldsymbol{\Lambda}$ = diag(5 5 0 0 0 5)
$\mathbf{K}$ = diag(500 500 0 0 0 500)
$\mathbf{G}$ = diag(500 500 0 0 0 1500)

Fig. 7 depicts the block diagram for the controller-neural network scheme. The output command vector $\boldsymbol{\tau}$ of the adaptive-sliding-mode control in combination with the output of the fault detection algorithm $\mathbf{W}$ are fed into the neural network. The neural network then generates an output signal $\boldsymbol{y}$ containing the solution vector $\boldsymbol{p}$ of the thrust allocation problem. Note that it is assumed that the thruster failure can be detected when it occurs. A detailed discussion on fault-detection methodologies is given by Omerdic and Roberts (2004).

5. Simulation and discussion

5.1. ROV system overview

Simulation studies were performed in order to demonstrate the effectiveness of the proposed control and thruster allocation scheme. The ROV system considered in the simulations is the Canadian Scientific Submersible Facility ROPOS vehicle. The

ROPOS vehicle is rectangular in shape with length, width, and height being 1.75, 2.6, and 1.45 m, respectively. The system has 3 pairs of electro-hydraulic thrusters: one pair of aft thrusters drives the surge and yaw motion; a second pair activates the heave direction through openings in the foam pack; and a third pair, directs lateral motions. The thruster saturation limit for each thruster is ± 900 N. The thruster layout on x–y axis of the vehicle is illustrated in Fig. 4. Dynamic parameters of the ROPOS vehicle are taken from Steinke (2006) and tabulated in Tables 1 and 2.

In the current simulation study, surge sway and yaw position and velocity state feedback is available through the onboard ROPOS navigation system. The ocean current of 0.5 knots along the positive X and Y axis of the inertial frame is considered in the simulation study. The system is assumed to be neutrally buoyant. In addition, the thruster dynamic response is assumed to be much faster than that of the ROV itself, and thus was neglected.

5.2. First simulation study

In this study, ROPOS obstacle-avoidance manoeuvres were simulated in the vehicle's local level plane. ROPOS was driven through a series of waypoints: $A\,(0, 0; 0)$, $B\,(2, 2; \pi/4)$, $C\,(4, 4; \pi/2)$, $D\ (2,\ 6;\ 3\pi/4)$, $E\ (0,\ 8;\ \pi)$ m; rad where the first two digits represent the absolute position values, and the last digit represents the yaw orientation value. Each segment of the path was traversed in 50 s. A corresponding continuous set of trajectory path values was generated using a fifth-order polynomial function.

The new adaptive method was implemented and used to drive the ROPOS vehicle through the manoeuvre. To reflect uncertainties that are present in the underwater vehicle paradigm, 40% modelling inaccuracies were incorporated into the controller's dynamic model. In addition, a constant disturbance moment of

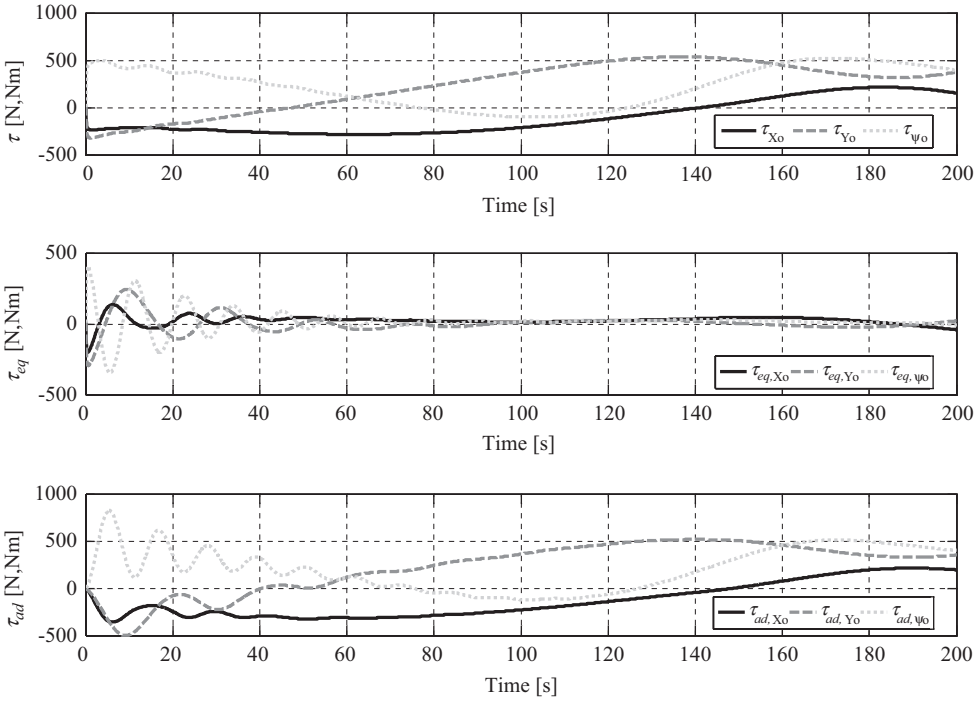

Fig. 8. Chattering-free adaptive sliding-mode controller output.

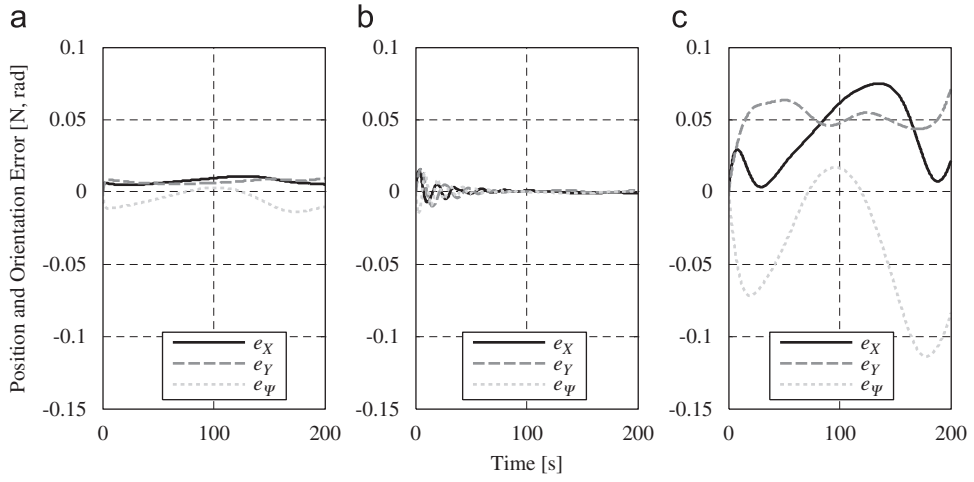

Fig. 9. Position and orientation error with respect to the inertial frame: (a) sliding-mode with boundary layer; (b) adaptive law is ON; (c) adaptive law is OFF.

150 N m is applied to the yaw axis of the vehicle to demonstrate the disturbance rejection capability of the controller. The knowledge of constant disturbance moment and the ocean current are not included into the controller's dynamic model to introduce further error. Controller parameters are given in Table 3. The controller output is given in Fig. 8.

The control law $\boldsymbol{\tau}$ and its equivalent and adaptive components, $\boldsymbol{\tau}_{eq}$ and $\boldsymbol{\tau}_{ad}$, are shown in Fig. 8. As can be seen, the proposed adaptive sliding-mode controller's on-line adaptation law yields a smooth control output without high-frequency controller demand, and therefore is chattering-free. In addition, the proposed controller provides excellent performance in the trajectory—following task is shown in Fig. 9b. This high performance is achieved despite the presence of high uncertainties that are illustrated in Fig. 10a.

As the vehicle moves, the controller performance significantly increases as shown in Fig. 9b. This can be attributed to the adaptation term: it learns as the vehicle progresses over its path, and as a consequence, begins to make better estimation of the lumped uncertainty vector as demonstrated in Fig. 10b and c. This high accuracy in the adaptive estimation justifies the choice of adaptive function given in Eq. (15). The estimation of the uncertainty is incorporated into the control law according to Eq. (16) leading to the improvement in the dynamic model inside the controller. The controller performance without the adaptation term is relatively poor as demonstrated in Fig. 9c. This is due to the fact that when the adaptation term is turned off, only the equivalent control law acts on the system. In this case, the performance relies heavily on the quality of the dynamic model inside the controller, which is poor due to the high uncertainty. The validity of Assumption 2 can be given by the plot of Eq. (43) which is given in Fig. 11. The initial positive value is due to the initial conditions of the simulation and is very quickly eliminated by the controller as the adaptive term begins to close the gap between the exact and estimated unknown dynamics.

A comparison study was performed to reveal the performance difference between two chattering-free sliding-mode approaches: one uses a conventional boundary layer approach (Slotine and Li, 1991) and the other is the proposed adaptive approach. The boundary layer width was set to 0.15 to ensure chattering-free response. The adaptive approach outperforms the boundary layer approach as shown in Fig. 9a, and the superior performance can be credited to the adaptive capability of the new method.

Fig. 12a demonstrates the l_∞ thrust output obtained from the designed neural network. In the simulations, the state dynamic equations of Eq. (36) were solved using the MATLAB *ode15s* command with $\kappa = 1 \times 10^6$. To have a comparison study, the thruster allocation was also completed using the l_2 norm minimization, i.e., the pseudo-inverse, and the results are demonstrated in Fig. 12b. Fig. 13 depicts the history of the largest single commanded thrust produced by the l_2 norm and l_∞ norm methods during the ROPOS manoeuvre. The result shows that the largest thrust value obtained from the l_2 norm minimization and l_∞ norm minimization are 342.6 and 268.4 N, respectively. Therefore, l_∞ norm minimization reduces the largest thrust by 21.6% in comparison to the one obtained from the l_2 norm minimization. The average improvement over the whole manoeuvre is found to be 11.9%. The results are given in Table 4. The results illustrate that when the lowest possible thrust magnitudes are desired to ensure future ROV manoeuvrability for subsequent control iterations, the l_∞ norm thrust allocation is an attractive alternative to the conventional pseudo-inverse solution. Note that since both l_∞ and l_2 generate the same control input based on $\boldsymbol{\tau} = \boldsymbol{Ep}$, there is no difference in the controller performance.

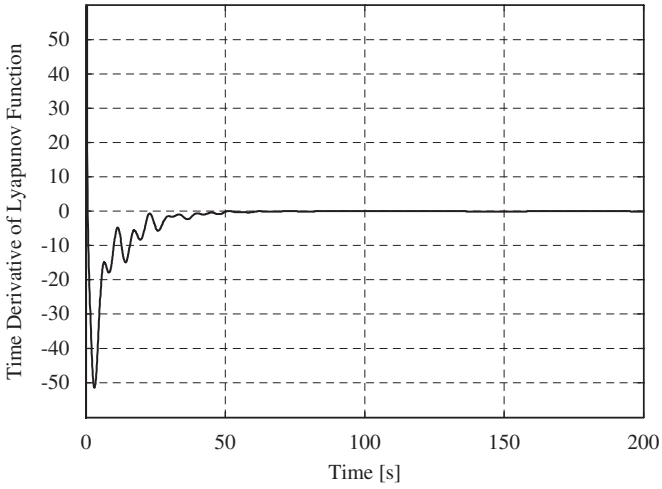

Fig. 11. Time derivative of Lyapunov function.

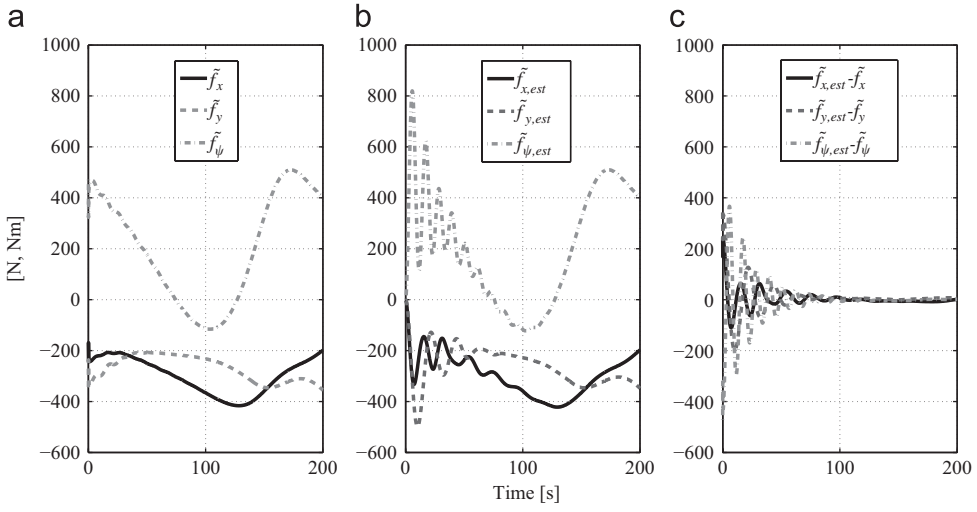

Fig. 10. Lumped uncertainty estimations: (a) real lumped uncertainty; (b) estimation of lumped uncertainty; (c) difference between real and lumped estimation.

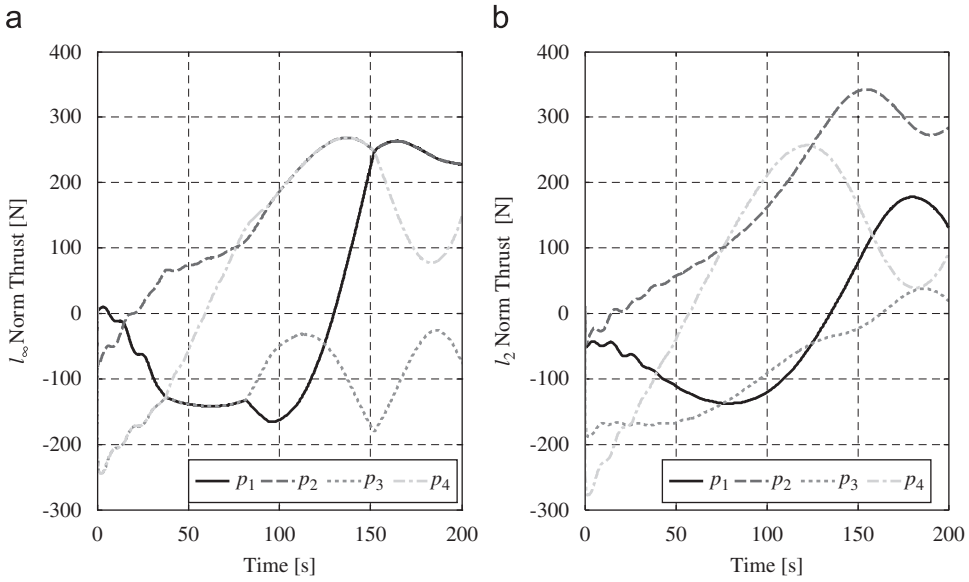

Fig. 12. Demanded thrusts: (a) l_∞ approach; (b) l_2 approach.

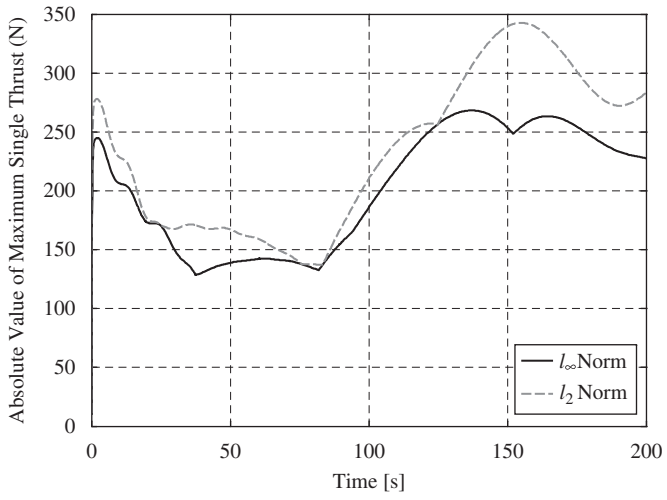

Fig. 13. Maximum single output of the l_∞ norm solution $\boldsymbol{p}_{l_\infty}$, and l_2 norm solution $\boldsymbol{p}_{l_2}$.

Table 4
Comparison of the norms

Max l_2 norm thrust (N)	Max l_∞ norm thrust (N)	Reduction (%)	Average reduction (%)
342.6	268.4	21.6	11.9

5.3. Second simulation study

In the second simulation study, the first case study was extended to include a thruster failure to demonstrate fault-tolerant property of the thrust allocation scheme.

It is assumed that the second and third thrusters partially fail between 40 and 150 s due to a temporary fouling of the propeller. The fouling produces a drop of 78% and 20% in the driving capacity of the second and third thrusters, respectively. This incident is accounted for in the thruster allocation scheme by setting $w_2 = 2/9$ and $w_3 = 4/5$. A complete failure of the second thruster occurs at 150 s, and for $t > 150$ s, the horizontal ROPOS manoeuvre is accomplished with its three functioning thrusters. Note that since the number of the remaining thrusters is the same as the number of the task space dimensions, the thruster allocation problem of $\boldsymbol{\tau} = \boldsymbol{Ep}$ becomes determined. In this case, the neural network converges to the unique solution of $\boldsymbol{\tau} = \boldsymbol{Ep}$ assuming a solution exists within the specified saturation range.

As Fig. 14a reveals, the faulty second thruster was driven to its upper saturation limit of 200 N at the time between 104 and 150 s. After the complete failure that occurs at 150 s, the faulty first thruster is not commanded and the remaining three thrusters are used by the fault-tolerant scheme to meet the controller demand. The third faulty thruster functions within its limits throughout the manoeuvre. Note that the instantaneous changes in the demanded thrust are caused by the sudden thruster failures. As Fig. 14b reveals, the controller performance is not affected by the faulty thrusters, and that the vehicle keeps the normal working process in the presence of faults and saturation limits. This means that the thrust allocation scheme is capable of finding a feasible solution in the smaller solution space due to the thrust failures. This result demonstrates the ability of the thruster allocation scheme to account for the thruster failure and saturation limits.

6. Conclusion

The contribution of the current work is twofold. First, a chattering-free sliding-mode controller has been proposed for ROV systems. The new controller uses an adaptive term in place of the conventional, discontinuous switching term. This adaptive term continuously compensates for the unknown system dynamics caused by poorly approximated nonlinear hydrodynamics or sudden environmental loads. As opposed to conventional sliding-mode controllers, the proposed controller does not need a *priori* knowledge of the upper bounds on the dynamics parameters of ROVs. In addition, the adaptive term does not require the parameterization of a regressor matrix and unknown parameter vector. Second, the l_∞ norm-based thrust allocation of ROVs has been proposed. It has been shown that a thrust allocation based on minimizing the l_∞ norm of the thrust manifold can be formulated as a linear programming problem that allows direct incorporation of faulty thrusters and thruster saturation limits. A recurrent neural network has been proposed

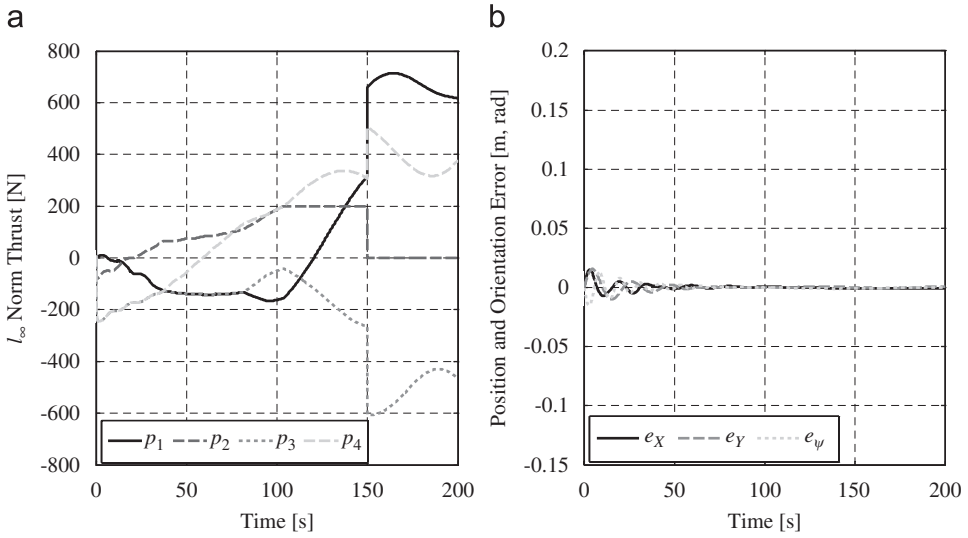

Fig. 14. (a) Demanded thrust from the l_∞ norm approach; (b) position and orientation error.

to solve the resulting linear programming problem. To this end, a general neural network design technique has been extended to include bounds on the thrust variables. The proposed neural network has the ability to achieve the real-time solution rates necessary in the ROV application. The resulting neural network has been applied to minimize the thrust allocation instantaneously with thruster saturation points and thruster failure considered. Future work will involve experimentally validating the proposed controller-thrust allocation scheme on an existing ROV system. This process will involve hardware realization of the recurrent neural network.

Acknowledgment

The authors would like to thank the Natural Sciences and Engineering Research Council (NSERC) of Canada for providing financial support for this work. The authors also wish to thank Dr. F. Firmani for contributing through several fruitful discussions.

Appendix A. Proof of Theorem 1

Define a Lyapunov function as

$$V = \tfrac{1}{2}(\boldsymbol{s}^{\mathrm{T}}\mathbf{M}_\eta\boldsymbol{s} + \boldsymbol{w}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\boldsymbol{w}) \tag{37}$$

where $\boldsymbol{w} = \tilde{\boldsymbol{f}}_{\mathrm{est}} - \tilde{\boldsymbol{f}}$ is the difference vector between the estimated lumped uncertainty vector and the exact lumped uncertainty vector. Differentiating V with respect to time yields

$$\dot{V} = \tfrac{1}{2}(\boldsymbol{s}^{\mathrm{T}}\dot{\mathbf{M}}_\eta\boldsymbol{s} + \dot{\boldsymbol{s}}^{\mathrm{T}}\mathbf{M}_\eta\boldsymbol{s} + \boldsymbol{s}^{\mathrm{T}}\mathbf{M}_\eta\dot{\boldsymbol{s}}) + \dot{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\boldsymbol{w} \tag{38}$$

In light of the fact that $\dot{\boldsymbol{s}}^{\mathrm{T}}\mathbf{M}_\eta\boldsymbol{s} = \boldsymbol{s}^{\mathrm{T}}\mathbf{M}_\eta\dot{\boldsymbol{s}}$, and using (10), $\mathbf{M}_\eta\ddot{\boldsymbol{\eta}} = \mathbf{J}^{-\mathrm{T}}\boldsymbol{\tau}_{\mathrm{eq}} - (\mathbf{C}_\eta\dot{\boldsymbol{\eta}} + \mathbf{D}_\eta\dot{\boldsymbol{\eta}} + \boldsymbol{g}_\eta)$ from (2), and $\dot{\boldsymbol{\eta}} = \boldsymbol{s} + \dot{\boldsymbol{\eta}}_r$ from Eq. (8)

$$\dot{V} = \tfrac{1}{2}\boldsymbol{s}^{\mathrm{T}}(\dot{\mathbf{M}}_\eta - 2\mathbf{C}_\eta)\boldsymbol{s} + \boldsymbol{s}^{\mathrm{T}}[\mathbf{J}^{-\mathrm{T}}\boldsymbol{\tau} - (\mathbf{M}_\eta\ddot{\boldsymbol{\eta}}_r + \mathbf{C}_\eta\dot{\boldsymbol{\eta}}_r + \mathbf{D}_\eta\dot{\boldsymbol{\eta}} + \boldsymbol{g}_\eta)] + \dot{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\boldsymbol{w} \tag{39}$$

Since $\dot{\mathbf{M}}_\eta - 2\mathbf{C}_\eta$ is a skew symmetric matrix, then $\boldsymbol{s}^{\mathrm{T}}(\dot{\mathbf{M}}_\eta - 2\mathbf{C}_\eta)\boldsymbol{s} = \mathbf{0}$, and Eq. (39) can be simplified to

$$\dot{V} = \boldsymbol{s}^{\mathrm{T}}(\mathbf{J}^{-\mathrm{T}}\boldsymbol{\tau} - \boldsymbol{f}_{\underline{r}}) + \dot{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\boldsymbol{w} \tag{40}$$

where $\boldsymbol{f}_{\underline{r}} = \mathbf{M}_\eta\ddot{\boldsymbol{\eta}}_r + \mathbf{C}_\eta\dot{\boldsymbol{\eta}}_r + \mathbf{D}_\eta\dot{\boldsymbol{\eta}} + \boldsymbol{g}_\eta$. Substituting Eq. (16) into Eq. (40), and using $\dot{\boldsymbol{\eta}}_r = \dot{\boldsymbol{\eta}} - \boldsymbol{s}$ and $\boldsymbol{w} = \tilde{\boldsymbol{f}}_{\mathrm{est}} - \tilde{\boldsymbol{f}}$ yields

$$\dot{V} = \boldsymbol{s}^{\mathrm{T}}(\boldsymbol{w} - \mathbf{K}\boldsymbol{s}) + \dot{\boldsymbol{w}}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\boldsymbol{w} \tag{41}$$

Substituting the adaptation law of Eq. (15) into Eq. (41) and using $\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma} = \mathbf{I}$, $\boldsymbol{\Gamma}^{\mathrm{T}} = \boldsymbol{\Gamma}$ gives

$$\dot{V} = -\boldsymbol{s}^{\mathrm{T}}\mathbf{K}\boldsymbol{s} - \dot{\tilde{\boldsymbol{f}}}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\boldsymbol{w} \tag{42}$$

In view of the positive definiteness of the gain matrix $\mathbf{K}$ and *Assumptions 1 and 2*, it can be stated that

$$\dot{V} \leqslant -\boldsymbol{s}^{\mathrm{T}}\mathbf{K}\boldsymbol{s} - \dot{\tilde{\boldsymbol{f}}}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\boldsymbol{w} \leqslant 0 \tag{43}$$

The inequality given in Eq. (43) implies that the system trajectories will converge to the sliding manifold $\boldsymbol{s} = \mathbf{0}$ from any nonzero initial error. However, Eq. (43) alone does not imply that $\dot{V} \to 0$ as $t \to \infty$, i.e., the system trajectories may not converge to the desired values in finite time. This problem can be solved using Barbalat's lemma (Popov, 1973; Slotine and Li, 1991), i.e., since V is lower bounded ($V \geqslant 0$), $\dot{V}$ is negative semi-definite ($\dot{V} \leqslant 0$), and $\ddot{V}$ is bounded (i.e., since $\boldsymbol{w}$ and $\dot{\tilde{\boldsymbol{f}}}$ are bounded, then s is bounded), then $\dot{V} \to 0$ as $t \to \infty$. In turn, $\boldsymbol{e} \to \mathbf{0}$ and $\dot{\boldsymbol{e}} \to \mathbf{0}$ as $t \to \infty$. □

Appendix B. Proof of Theorem 3

The stability proof of the neural network given in Eq. (35) involves the following two lemmas:

Lemma 1. *If $f : C \subset \mathbb{R}^n \to \mathbb{R}$ is differentiable over a convex set $C_c \subset C$, then f is convex over C_c if and only if*

$$(\boldsymbol{x}_1 - \boldsymbol{x})^{\mathrm{T}}\nabla f(\boldsymbol{x}) \leqslant f(\boldsymbol{x}_1) - f(\boldsymbol{x}) \quad \forall \boldsymbol{x}_1,\ \boldsymbol{x} \in C_c \tag{44}$$

where $\nabla f(x)$ is the gradient vector of $f(x)$.

Proof of Lemma 1. See Ortega and Rheinboldt (1970).

Lemma 2. *$\boldsymbol{y}^*$ and $(\boldsymbol{\mu}^*, \boldsymbol{\lambda}^*)$ are the optimal solutions to the primal problem, Eq. (27), and its dual problem, Eq. (32), respectively, if and only if $E(\boldsymbol{v}^*) = 0$ and*

$$(\boldsymbol{v}^* - \boldsymbol{v})^{\mathrm{T}}\nabla E(\boldsymbol{v}) \leqslant -E(\boldsymbol{v}) \tag{45}$$

where $\boldsymbol{v}^* = [\boldsymbol{y}^{*\mathrm{T}} \quad \boldsymbol{\mu}^{*\mathrm{T}} \quad \boldsymbol{\lambda}^{*\mathrm{T}}]^{\mathrm{T}}$ and $\boldsymbol{v} = \begin{bmatrix}\boldsymbol{y}^{\mathrm{T}} & \boldsymbol{\mu}^{\mathrm{T}} & \boldsymbol{\lambda}^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$.

Proof of Lemma 2. According to the definition of the energy function, it is evident that $E(\boldsymbol{v}^*) = 0$. By applying Lemma 1 to the energy function with $E(\boldsymbol{v}^*) = 0$, Lemma 2 can be verified. □

Without loss of generality, let $\kappa = 1$ and consider the following positive definite Lyapunov function candidate for the neural

network of Eq. (35) (Tang and Wang, 2001):

$$V(\boldsymbol{v}) = \tfrac{1}{2}(\boldsymbol{v}^* - \boldsymbol{v})^{\mathrm{T}}(\boldsymbol{v}^* - \boldsymbol{v}) \tag{46}$$

Taking the time derivative of Eq. (46) and using Eq. (35) yields

$$\frac{\mathrm{d}V(\boldsymbol{v})}{\mathrm{d}t} = \frac{\mathrm{d}V(\boldsymbol{v})}{\mathrm{d}\boldsymbol{v}}\left(\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}t}\right) = (\boldsymbol{v}^* - \boldsymbol{v})^{\mathrm{T}}\left(\frac{\mathrm{d}\boldsymbol{v}}{\mathrm{d}t}\right) = (\boldsymbol{v}^* - \boldsymbol{v})^{\mathrm{T}}\nabla E(\boldsymbol{v}) \tag{47}$$

By applying Lemma 2 and using the fact that the energy function is positive definite $E(\boldsymbol{v}) \geqslant 0$,

$$(\boldsymbol{v}^* - \boldsymbol{v})^{\mathrm{T}}\nabla E(\boldsymbol{v}) \leqslant -E(\boldsymbol{v}) \leqslant 0 \tag{48}$$

Therefore, the neural network of (35) is Lyapunov stable. According to LaSalle's invariance rule (Slotine and Li, 1991), all trajectories $\boldsymbol{v}(t)$ converge to the largest invariant set in the set $\Omega = \{\boldsymbol{v} \in \mathbb{R}^{5n+m} | \dot{V} = 0\}$.

In view of Eq. (47), $\dot{V} = 0$ implies $\dot{\boldsymbol{v}} = 0$, and this further implies that the set Ω also contains the equilibrium points of (35) i.e., $\Omega = \{\boldsymbol{v} \in \mathbb{R}^{5n+m} | \dot{V} = 0\} = \{\boldsymbol{v} \in \mathbb{R}^{5n+m} | \dot{\boldsymbol{v}} = 0\}$. Thus, Eq. (35) converges to its equilibrium points. According to Lemma 2, $E(\boldsymbol{v}^*) = 0$ if and only if $\nabla E(\boldsymbol{v}^*) = 0$ implying that $\boldsymbol{v}^*$ makes $\dot{\boldsymbol{v}} = 0$. Therefore, the equilibrium points of Eq. (35) correspond to $\boldsymbol{v}^*$.

Since the energy function Eq. (33) is differentiable and convex for all $\boldsymbol{v}$, the local minimum of the energy function corresponds to the global minimum. The neural network given in Eq. (35) is globally stable and will converge to the optimal solutions of the primal problem, Eq. (27), and its dual problem, Eq. (32). This completes the proof. □

References

Antonelli, G., 2003. Underwater Robots: Motion and Force Control of Vehicle–Manipulator Systems. Springer, Berlin.

Antonelli, G., Chiaverini, S., Sarkar, N., West, M., 2001. Adaptive control of and autonomous underwater vehicle: experimental results on ODIN. IEEE Transactions on Control Systems and Technology 9 (5), 756–765.

Antonelli, G., Caccavale, F., Chiaverini, S., Fusco, G., 2003. A novel adaptive control law for underwater vehicles. IEEE Transactions on Control Systems and Technology 11 (2), 109–120.

Antoniou, A., Lu, W.S., 2007. Practical Optimization: Algorithms and Engineering Applications. Springer, Berlin.

Arati, S.D., Walker, I.D., 1997. Minimum effort inverse kinematics for redundant manipulators. IEEE Transactions on Robotics and Automation 13 (5), 767–775.

Bordignon, K.A., 1996. Constrained control allocation for systems with redundant control effectors. Ph.D. Thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Conte, G., Serrani, A., 1998. Robust control of a remotely operated underwater vehicle. Automatica 34 (2), 193–198.

Debitetto, P., 1994. Fuzzy logic for depth control of unmanned undersea vehicles. In: Proceedings of the AUV Symposium, Cambridge, MA, USA, pp. 233–241.

Durham, W., 1993. Constraint control allocation. Journal of Guidance, Control and Dynamics 16 (4), 717–725.

Elmali, H., Olgac, N., 1992. Theory and implementation of sliding mode control with perturbation estimation. In: Proceedings of the IEEE International Conference on Robotics and Automation, vol. 3, Nice, France, pp. 2114–2119.

Firmani, F., Zibil, A., Nokleby, S.B., Podhorodeski, R.P., 2007. Wrench capabilities of planar parallel manipulators—part I: wrench polytopes and performance indices, Robotica, in preparation.

Fossen, T., 1994. Guidance and Control of Ocean Vehicles. Wiley, New York.

Gill, P., Murray, W., Wright, M.H., 1991. Numerical Linear Algebra and Optimization 1. Addison-Wesley, Reading, MA.

Healey, A., Lienard, D., 1993. Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. IEEE Journal of Oceanic Engineering 18 (3), 327–339.

Ishii, K., Ura, T., 2000. An adaptive neural-net controller system for an underwater vehicle. Control Engineering Practice 8, 177–184.

Kato, N., 1995. Applications of fuzzy algorithm to guidance of and control of underwater vehicle. In: Yuh, J. (Ed.), Underwater Robotic Vehicles: Design and Control. TSI Press, Albuquerque.

Kodogiannis, V., 2003. Direct adaptive control of underwater vehicles using neural networks. Journal of Vibration and Control 9, 605–619.

Lin, F.J., Wai, R.J., 2002. Robust control using neural network uncertainty observer for linear induction motor servo drive. IEEE Transactions on Power Electronics 17 (2), 241–254.

Omerdic, E., Roberts, G., 2004. Thruster fault diagnosis and accommodation for open-frame underwater vehicles. Control Engineering Practice 12, 1575–1598.

Ortega, J.M., Rheinboldt, W.G., 1970. Iterative Solution of Nonlinear Equations in Several Variables. Academic Press, New York.

Pepijn, W.J., Colin van de Ven, F., Daniel, T., 2005. Neural network control of underwater vehicles. Engineering Applications of Artificial Intelligence 18, 533–547.

Popov, V., 1973. Hyperstability of Control Systems. Springer, Berlin.

Sarkar, N., Podder, T.K., Antonelli, G., 2002. Fault-accommodating thruster force allocation of an AUV considering thruster redundancy and saturation. IEEE Transactions on Robotics and Automation 18 (2), 223–231.

Slotine, J.J., Coetsee, J., 1986. Adaptive sliding controller synthesis for non-linear systems. International Journal of Control 43, 1631–1651.

Slotine, J.J., Li, W., 1991. Applied Nonlinear Control. Prentice-Hall, Englewood Cliffs, NJ.

Slotine, J.J., Shastry, S., 1983. Tracking control of nonlinear system using sliding surfaces. International Journal of Control 38 (2), 465–492.

Sordalen, O., 1997. Optimal thrust allocation method for marine vessels. Control Engineering Practice 5 (9), 93–106.

Soylu, S., Buckham, B., Podhorodeski, R., 2007. Robust control of underwater vehicles with fault-tolerant infinity-norm thruster force allocation. In: Proceedings of the OCEANS 2007 MTS/IEEE Conference and Exhibition, Vancouver, BC, Canada.

Steinke, D., 2006. Design and simulation of a Kalman filter for ROV navigation. Master's Thesis, Victoria, BC, Canada.

Tang, W., Wang, J., 2001. A recurrent neural network for minimum infinity-norm kinematic control of redundant manipulators with an improved problem formulation and reduced architecture complexity. IEEE Transactions on Systems, Man, and Cybernetics 31 (1), 98–105.

Van de Ven, P., Flanagan, C., Toal, D., 2005. Neural network control of underwater vehicles. Engineering Applications of Artificial Intelligence 18, 533–547.

Wang, J., 1993. Analysis and design of a recurrent neural network for linear programming. IEEE Transactions on Circuits and Systems I 40 (9), 613–618.

Wolfe, P., 1961. A duality theorem for nonlinear programming. Quarterly of Applied Mathematics 19, 239–244.

Wright, M.H., 1992. Interior methods for constrained optimization. Acta Numerica 1, 341–407.

Xia, Y., 1996. A new neural network for solving linear programming problems and its applications. IEEE Transactions on Neural Networks 7, 525–529.

Xia, Y., Wang, J., 1998. A general methodology for designing globally convergent optimization neural networks. IEEE Transactions on Neural Networks 9 (6), 1331–1343.

Yoerger, D., Slotine, J.J., 1985. Robust trajectory control of underwater vehicles. IEEE Journal of Oceanic Engineering OE-10, 462–470.

Zeinali, M., Notash, L., 2007. Fuzzy logic-based adaptive robust control for parallel manipulators. In: Proceedings of the 12th World Congress in Mechanism and Machine Science, Besançon, France.