

A Ship Heading and Speed Control Concept Inherently Satisfying Actuator Constraints

Mikkel Eske Nørsgaard Sørensen, Morten Breivik and Bjørn-Olav H. Eriksen

Abstract—Satisfying actuator constraints is often not considered in the academic literature on the design of ship heading and speed controllers. This paper considers the use of a simplified dynamic window algorithm as a way to ensure that actuator constraints are satisfied. To accomplish this, we use the simplified dynamic window algorithm as a dynamic window-based controller (DWC) to guarantee that the velocities remain within a set of feasible boundaries, while simultaneously respecting the actuator constraints. We also develop a modified nonlinear ship model on which to test the proposed concept. The DWC is compared with a more traditional ship heading and speed controller, using performance metrics which consider both control accuracy and energy use.

I. INTRODUCTION

When a ship sails the sea, its autopilot system usually leads the ship along the desired heading. Numerous motion controllers and autopilots have been proposed over the years. However, many control algorithms found in the literature do not consider saturation constraints for the actuators. Examples of traditional control designs for ship autopilot systems are given in [1]. Not considering actuator constraints may lead to unsatisfying performance or stability issues. In [2], a gain-scheduled control law is developed and tested for handling actuator constraints for a rudder-roll model of a ship.

In [3], the dynamic window (DW) algorithm is suggested as a method to perform collision avoidance and deal with constraints imposed by limited velocities and accelerations for mobile robots. This algorithm first generates a set of possible trajectories. Based on these trajectories, a search space of possible velocities can be approximated. The acceleration constraints are considered by limiting the search space to reachable velocities within a next time interval. To reduce the search space even further, all non-admissible velocities are removed to make the vehicle stop safely before it reaches the closest obstacle on the corresponding trajectory.

The DW algorithm is modified for AUVs in [4] and shows promising results for handling magnitude and rate constraints for the actuators. In this paper, we consider a simplification of the DW algorithm in [4], by removing the collision avoidance part of the algorithm. In particular, this DW-based controller (DWC) will be combined with a heading controller based on the design in [5].

The contribution of this paper is the proposal of the DWC, which inherently satisfies actuator constraints. Furthermore,

M. E. N. Sørensen, M. Breivik and B.-O. H. Eriksen are with the Centre for Autonomous Marine Operations and Systems, Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), NO-7491 Trondheim, Norway. Email: {Mikkel.E.N.Sorensen, morten.breivik, bjorn-olav.h.eriksen}@ieee.org

necessary modifications to a 3 degrees-of-freedom nonlinear ship model based on [6] are done, in order to achieve a more physically realistic behavior. The DWC is compared with a traditional controller (TC) from [5], and the performance of the controllers are compared through simulations, where the comparison is made using performance metrics which consider both control accuracy and energy use.

The structure of the paper is as follows: A mathematical ship model and assumptions are presented in Section II; Section III describes the assumptions and control objective; Section IV presents the design of a traditional control inspired by backstepping and constant-bearing guidance; Section V presents the proposed DWC concept; Section VI presents simulation results; while Section VII concludes the paper.

II. SHIP MODEL

The motion of a ship can be represented by the pose vector $\boldsymbol{\eta} = [x, y, \psi]^\top \in \mathbb{R}^2 \times \mathbb{S}$ and the velocity vector $\boldsymbol{\nu} = [u, v, r]^\top \in \mathbb{R}^3$. Here, (x, y) represents the Cartesian position in the local earth-fixed reference frame, ψ is the yaw angle, (u, v) represents the body-fixed linear velocities and r is the yaw rate. The 3 degrees-of-freedom dynamics of a ship can then be stated as [1]:

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\boldsymbol{\nu} \quad (1)$$

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{D}(\boldsymbol{\nu})\boldsymbol{\nu} = \boldsymbol{\tau}, \quad (2)$$

where $\mathbf{M} \in \mathbb{R}^{3 \times 3}$, $\mathbf{C}(\boldsymbol{\nu}) \in \mathbb{R}^{3 \times 3}$, $\mathbf{D}(\boldsymbol{\nu}) \in \mathbb{R}^{3 \times 3}$ and $\boldsymbol{\tau} = [\tau_1, \tau_2, \tau_3]^\top$ represent the inertia matrix, Coriolis and centripetal matrix, damping matrix and control input vector, respectively. The rotation matrix $\mathbf{R}(\psi) \in SO(3)$ is given by

$$\mathbf{R}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

The system matrices are assumed to satisfy the properties $\mathbf{M} = \mathbf{M}^\top > 0$, $\mathbf{C}(\boldsymbol{\nu}) = -\mathbf{C}(\boldsymbol{\nu})^\top$ and $\mathbf{D}(\boldsymbol{\nu}) > 0$.

A. Nominal Model

The model and parameters of the model-scale ship CyberShip II [6] will be used for control design and evaluation through numerical simulations in this paper. CyberShip II is a 1:70 scale replica of a supply ship, with a length of $L = 1.255$ m. The inertia matrix is given as

$$\mathbf{M} = \mathbf{M}_{RB} + \mathbf{M}_A, \quad (4)$$

where

$$\mathbf{M}_{RB} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & mx_g \\ 0 & mx_g & I_z \end{bmatrix} \quad (5)$$

$$\mathbf{M}_A = \begin{bmatrix} -X_{\dot{u}} & 0 & 0 \\ 0 & -Y_{\dot{v}} & -Y_{\dot{r}} \\ 0 & -N_{\dot{v}} & -N_{\dot{r}} \end{bmatrix}. \quad (6)$$

The mass of CyberShip II is $m = 23.8$ kg, while $x_g = 0.046$ m is the distance along the x -axis in the body frame from the centre of gravity, and $I_z = 1.760$ kg m² is the moment of inertia about the z -axis in the body frame. Other parameter values are listed in Table I. The Coriolis and centripetal matrix is

$$\mathbf{C}(\boldsymbol{\nu}) = \mathbf{C}_{RB}(\boldsymbol{\nu}) + \mathbf{C}_A(\boldsymbol{\nu}), \quad (7)$$

with

$$\mathbf{C}_{RB}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & -m(x_g r + v) \\ 0 & 0 & mu \\ m(x_g r + v) & -mu & 0 \end{bmatrix} \quad (8)$$

$$\mathbf{C}_A(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & -c_{A,13}(\boldsymbol{\nu}) \\ 0 & 0 & c_{A,23}(\boldsymbol{\nu}) \\ c_{A,13}(\boldsymbol{\nu}) & -c_{A,23}(\boldsymbol{\nu}) & 0 \end{bmatrix}, \quad (9)$$

where

$$c_{A,13}(\boldsymbol{\nu}) = -Y_{\dot{v}}v - \frac{1}{2}(N_{\dot{v}} + Y_{\dot{r}})r \quad (10)$$

$$c_{A,23}(\boldsymbol{\nu}) = -X_{\dot{u}}u. \quad (11)$$

Finally, the damping matrix $\mathbf{D}(\boldsymbol{\nu})$ is given as

$$\mathbf{D}(\boldsymbol{\nu}) = \mathbf{D}_L + \mathbf{D}_{NL}(\boldsymbol{\nu}), \quad (12)$$

where

$$\mathbf{D}_L = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & -Y_r \\ 0 & -N_v & -N_r \end{bmatrix} \quad (13)$$

$$\mathbf{D}_{NL}(\boldsymbol{\nu}) = \begin{bmatrix} d_{NL,11}(\boldsymbol{\nu}) & 0 & 0 \\ 0 & d_{NL,22}(\boldsymbol{\nu}) & d_{NL,23}(\boldsymbol{\nu}) \\ 0 & d_{NL,32}(\boldsymbol{\nu}) & d_{NL,33}(\boldsymbol{\nu}) \end{bmatrix}, \quad (14)$$

and

$$d_{NL,11}(\boldsymbol{\nu}) = -X_{|u|u}|u| - X_{uuu}u^2 \quad (15)$$

$$d_{NL,22}(\boldsymbol{\nu}) = -Y_{|v|v}|v| - Y_{|r|v}|r| \quad (16)$$

$$d_{NL,23}(\boldsymbol{\nu}) = -Y_{|v|r}|v| - Y_{|r|r}|r| \quad (17)$$

$$d_{NL,32}(\boldsymbol{\nu}) = -N_{|v|v}|v| - N_{|r|v}|r| \quad (18)$$

$$d_{NL,33}(\boldsymbol{\nu}) = -N_{|v|r}|v| - N_{|r|r}|r|. \quad (19)$$

The considered model describes a fully actuated ship. However, heading and speed controllers are typically used at higher speeds, where the ship is underactuated. We have therefore excluded the bow thruster from the actuator model since it loses its effectiveness at high speeds. Inspired by [6], the modified ship actuator forces and moments can be modelled using two thrusters $\mathbf{n} = [n_1, n_2]^\top \in \mathbb{R}^2$

TABLE I: Parameters for CyberShip II [6]

Parameter	Value	Parameter	Value
$X_{\dot{u}}$	-2	N_v	0.03130
$Y_{\dot{v}}$	-10	$N_{ v v}$	3.95645
$Y_{\dot{r}}$	0	$Y_{ r v}$	-0.805
$N_{\dot{v}}$	0	Y_r	-7.250
$N_{\dot{r}}$	-1	$Y_{ v r}$	-0.845
X_u	-0.72253	$Y_{ r r}$	-3.450
$X_{ u u}$	-1.32742	$Y_{ v r}$	0.080
X_{uuu}	-5.86643	$N_{ r v}$	0.130
Y_v	-0.88965	N_r	-1.900
$Y_{ v v}$	-36.47287	$N_{ r r}$	-0.750

with revolutions per second (rps) and two rudder angles $\boldsymbol{\delta} = [\delta_1, \delta_2]^\top \in \mathbb{S}^2$. These are related to the input vector $\boldsymbol{\tau}$ through the actuator model

$$\boldsymbol{\tau}(\boldsymbol{\nu}, \mathbf{n}, \boldsymbol{\delta}) = \mathbf{B}\boldsymbol{\tau}_{act}(\boldsymbol{\nu}, \mathbf{n}, \boldsymbol{\delta}), \quad (20)$$

where $\mathbf{B} \in \mathbb{R}^{3 \times 4}$ is an actuator configuration matrix. The function $\boldsymbol{\tau}_{act} : \mathbb{R}^3 \times \mathbb{R}^2 \times \mathbb{S}^2 \rightarrow \mathbb{R}^4$ relates the actuator variables \mathbf{n} and $\boldsymbol{\delta}$ to the input vector $\boldsymbol{\tau}$ for a given velocity $\boldsymbol{\nu}$. The actuator configuration matrix and actuator force vector is

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ |l_{yT_1}| & -|l_{yT_2}| & -|l_{xR_1}| & -|l_{xR_2}| \end{bmatrix}. \quad (21)$$

Moreover, $\boldsymbol{\tau}_{act}$ is given by

$$\boldsymbol{\tau}_{act} = [T_1, T_2, L_1, L_2]^\top, \quad (22)$$

with

$$T_i \triangleq T_{|n|n}|n_i|n_i - T_{|n|u}|n_i|u, \quad \text{for } i = 1, 2 \quad (23)$$

$$L_i \triangleq (L_\delta \delta_i - L_{|\delta|\delta}|\delta_i|\delta_i)|u|u, \quad \text{for } i = 1, 2, \quad (24)$$

where T_i is the thrust force from the preceding propeller and L_i is the lift force from the preceding rudder. The constants $l_{yT_1}, l_{yT_2}, l_{xR_1}, l_{xR_2}$ represent physical placements of the actuators, and the parameters $T_{|n|n}, T_{|n|u}, L_\delta, L_{|\delta|\delta}$ are positive coefficients. It should be noted that (24) is dependent on the surge speed, which leads to a saturation constraint of the yaw moment having a nonlinear behavior, which is 0 if the surge speed is 0. In [6], the actuator variable limitations are stated as $n_i \in [0, 33.33]$ rps and $\delta_i \in [-35, 35]$ deg. Based on (20)-(24), infinitely many combinations of the actuator variables \mathbf{n} and $\boldsymbol{\delta}$ can generate the input vector $\boldsymbol{\tau}$. Here, we assume that $n_1 = n_2$ and $\delta_1 = \delta_2$. We will only consider the control of the surge and yaw motion, since our model is underactuated.

Using this ship model, we can map the steady-state solution of (2) associated with a given control input. In particular, the blue asterisks in Fig. 1 represent the steady-state solutions for a set of uniformly distributed control inputs. Analysing the model from [6], it is concluded that the modelled Munk moment, which is a destabilizing factor, give rise to physically impossible motion. In this model, the equilibrium point at $r = 0$ is unstable, which is not consistent with the actual behavior of CyberShip II.

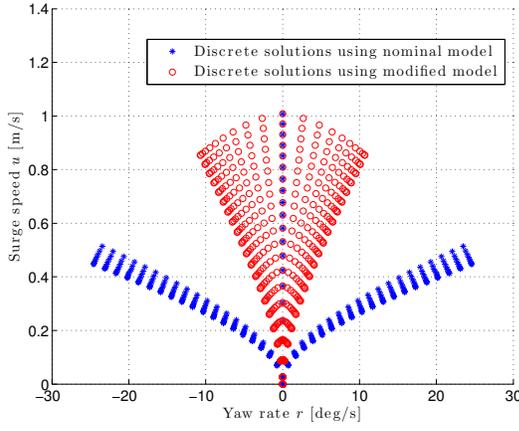


Fig. 1: Possible combinations of surge speed and yaw rate using the same control inputs for the nominal model [6] (blue) and the modified model based on [7] (red).

B. Modified Model

In [7], an analysis on how to accommodate for the Munk moment on an AUV is made. It is suggested to add damping terms to the damping matrix that are linearly increasing with the forward speed. Based on the observation in [7], we change (17)-(19) to

$$d_{NL,23}(\boldsymbol{\nu}) = -Y_{|v|r}|v| - Y_{|r|r}|r| - Y_{ur}u \quad (25)$$

$$d_{NL,32}(\boldsymbol{\nu}) = -N_{|v|v}|v| - N_{|r|v}|r| - N_{uv}u \quad (26)$$

$$d_{NL,33}(\boldsymbol{\nu}) = -N_{|v|r}|v| - N_{|r|r}|r| - N_{ur}u, \quad (27)$$

where

$$Y_{ur} = X_{\dot{u}} \quad (28)$$

$$N_{uv} = -(Y_{\dot{u}} - X_{\dot{u}}) \quad (29)$$

$$N_{ur} = Y_{\dot{r}}, \quad (30)$$

to get a more physically realistic model behavior. The red circles in Fig. 1 show the steady-state solutions for the combined [6] and [7] model using the same set of control inputs as previously. The steady-state response of this modified model qualitatively corresponds to the experimentally derived response of a high-speed boat in [8].

III. ASSUMPTIONS AND CONTROL OBJECTIVE

It is assumed that both the pose vector $\boldsymbol{\eta}(t)$ and velocity vector $\boldsymbol{\nu}(t)$ can be measured, and that no disturbances and uncertainties affect the system.

The control objective is to make $\tilde{\psi}(t) \triangleq \psi(t) - \psi_t(t) \rightarrow 0$ as $t \rightarrow \infty$ and $\tilde{u}(t) \triangleq u(t) - u_t(t) \rightarrow 0$ as $t \rightarrow \infty$, where $\psi_t(t) \in \mathbb{S}$ represents the heading associated with a target ship and u_t is the target surge speed. Furthermore, $\psi_t(t)$ is \mathcal{C}^2 and bounded. The motion of the target ship is typically defined by a human or generated by a guidance system.

For notational simplicity, the time t is omitted in most of this paper.

IV. TRADITIONAL CONTROL DESIGN

Using a combination of a cascaded feedback controller [5] and a dynamic feedback controller where the dynamics of the uncontrolled sway mode enters the yaw control law [9], the control input can be chosen as

$$\boldsymbol{\tau} = \boldsymbol{M}\dot{\boldsymbol{\alpha}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\alpha} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\alpha} - \boldsymbol{K}_2(\cdot)\boldsymbol{z}_2. \quad (31)$$

The error variables z_1 and $\boldsymbol{z}_2 = [z_{2,u}, z_{2,v}, z_{2,r}]^\top$ are defined as

$$z_1 \triangleq \psi - \psi_t \quad (32)$$

$$\boldsymbol{z}_2 \triangleq \boldsymbol{\nu} - \boldsymbol{\alpha}, \quad (33)$$

where $\boldsymbol{\alpha} = [\alpha_u, \alpha_v, \alpha_r] \in \mathbb{R}^3$ is a vector of stabilising functions, which can be interpreted as a desired velocity

$$\alpha_u = u_t \quad (34)$$

$$\alpha_r = \dot{\psi}_t - K_1(\cdot)z_1, \quad (35)$$

where

$$K_1(\cdot) = \Gamma_1 \frac{1}{\sqrt{z_1^2 + \Delta_{\dot{\psi}}^2}}, \quad (36)$$

represents a nonlinear control gain with $\Gamma_1 > 0$ and $\Delta_{\dot{\psi}} > 0$. The nonlinear feedback term in (31) is given as

$$\boldsymbol{K}_2(\cdot) = \boldsymbol{\Gamma}_2 \begin{bmatrix} \frac{1}{\sqrt{z_{2,\bar{v}}^2 z_{2,\bar{v}} + \Delta_{\bar{v}}^2}} \boldsymbol{I}_{2 \times 2} & \mathbf{0}_{2 \times 1} \\ \mathbf{0}_{1 \times 2} & \frac{1}{\sqrt{z_{2,\bar{r}}^2 + \Delta_{\bar{r}}^2}} \end{bmatrix}, \quad (37)$$

with the control gain $\boldsymbol{\Gamma}_2 > 0$, where $z_{2,\bar{v}}$ is defined as $z_{2,\bar{v}} \triangleq [z_{2,u}, z_{2,v}]^\top$, $\Delta_{\bar{v}} > 0$ and $\Delta_{\bar{r}} > 0$. The time derivative of the vector of stabilising functions then becomes

$$\dot{\boldsymbol{\alpha}} = [\dot{u}_t, \dot{\alpha}_v, \dot{\alpha}_r]^\top, \quad (38)$$

where \dot{u}_t is the target surge acceleration and

$$\dot{\alpha}_r = \ddot{\psi}_t - \dot{K}_1(z_1, \Delta_i)z_1 - K_1(z_1, \Delta_i)\dot{z}_1, \quad (39)$$

with

$$\dot{z}_1 = -K_1(\cdot)z_1 + z_{2,\bar{r}}, \quad (40)$$

and

$$\dot{K}_1(\cdot) = \Gamma_1 \frac{z_1 \dot{z}_1}{(z_1^2 + \Delta_{\dot{\psi}}^2)^{\frac{3}{2}}}. \quad (41)$$

Based on design of the dynamics of the uncontrolled sway mode in [9], the variable α_v is a dynamic state of the controller, and is given by

$$m_{22}\dot{\alpha}_v = -d_{22}(\boldsymbol{\nu})\alpha_v + \gamma(\alpha_r, \dot{\alpha}_r, \boldsymbol{z}_2), \quad (42)$$

where

$$\gamma(\alpha_r, \dot{\alpha}_r, \boldsymbol{z}_2) = K_{2,22}(\cdot)z_{2,v} - m_{23}\dot{\alpha}_r - d_{23}(\boldsymbol{\nu})\alpha_r - c_{23}(\boldsymbol{\nu})\alpha_r, \quad (43)$$

and m_{ij} , $c_{ij}(\boldsymbol{\nu})$, $d_{ij}(\boldsymbol{\nu})$ and $K_{2,ij}(\cdot)$ are components at the i th row and j th column of the matrices \boldsymbol{M} , $\boldsymbol{C}(\boldsymbol{\nu})$, $\boldsymbol{D}(\boldsymbol{\nu})$ and $\boldsymbol{K}_2(\cdot)$, while

$$\alpha_v = \int_0^t \dot{\alpha}_v(\sigma) d\sigma, \quad \alpha_v(0) = v(0). \quad (44)$$

V. DYNAMIC WINDOW-BASED CONTROL DESIGN

A. Simplified Dynamic Window Algorithm

Here, we describe a step-by-step design procedure for a simplified version of the dynamic window (DW) algorithm presented in [4] by removing the collision avoidance part of the algorithm.

Based on the modified ship model and its actuator magnitude constraints, a set of possible velocities can be found. This set contains all velocities the ship can achieve, with respect to the actuator constraints. The possible velocities can be found by computing the steady-state solution of the kinetics (2) for all possible control inputs:

$$\tau(\boldsymbol{\nu}_{ss}, \mathbf{n}, \boldsymbol{\delta}) = \mathbf{C}(\boldsymbol{\nu}_{ss})\boldsymbol{\nu}_{ss} + \mathbf{D}(\boldsymbol{\nu}_{ss})\boldsymbol{\nu}_{ss}, \quad (45)$$

within the actuator magnitude constraints

$$n_i \in [0, 33.33] \text{ rps} \quad (46)$$

$$\delta_i \in [-30, 30] \text{ deg.} \quad (47)$$

The steady-state solutions of (45)-(47) for a uniformly distributed set of the control inputs is shown in Fig. 2. By designing an approximation of the boundaries, the set of possible velocities can be defined as:

$$V_p = \{(u, r) \in \mathbb{R} \times \mathbb{R} | g(u, r) \geq 0\}, \quad (48)$$

where $g(u, r)$ is greater than or equal to zero for valid solutions of (45)-(47), and negative otherwise. Given m approximated boundaries, defined by the functions $h_a(u, r) = 0$, $a \in \{1, 2, \dots, m\}$ where $\nabla h_a(u, r)$ is required to be pointing inwards to the valid solutions, the approximated $g(u, r)$ is given as:

$$g(u, r) = \min(h_1(u, r), h_2(u, r), \dots, h_m(u, r)). \quad (49)$$

In Fig. 3, a plot of the function $g(u, r)$ is shown.

Next, the space of reachable points within a time step T needs to be defined. This is done by finding acceleration

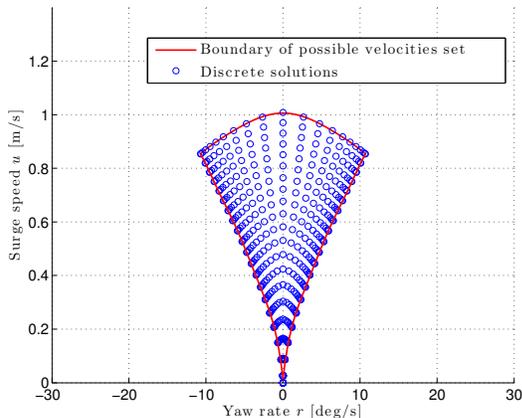


Fig. 2: Possible combinations of surge speed and yaw rate, with respect to actuator magnitude limits. The approximated boundary of V_p is shown as the red line.

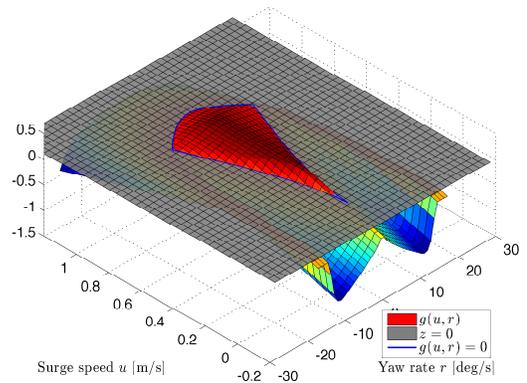


Fig. 3: Function to find possible velocities.

limits, and based on these, the set of reachable velocities can be computed. The possible ship accelerations can be found by evaluating

$$\dot{\boldsymbol{\nu}} = \mathbf{M}^{-1}(\tau(\boldsymbol{\nu}^*, \mathbf{n}, \boldsymbol{\delta}) - \mathbf{C}(\boldsymbol{\nu}^*)\boldsymbol{\nu}^* - \mathbf{D}(\boldsymbol{\nu}^*)\boldsymbol{\nu}^*), \quad (50)$$

for the current velocity $\boldsymbol{\nu}^* = \boldsymbol{\nu}(t)$ and boundaries of the control input vector. The acceleration limits at the current time step can be computed as:

$$\dot{\boldsymbol{\nu}}_{min} = \mathbf{M}^{-1}(\tau(\boldsymbol{\nu}^*, \mathbf{n}_{min}, \boldsymbol{\delta}_{max}) - \mathbf{C}(\boldsymbol{\nu}^*)\boldsymbol{\nu}^* - \mathbf{D}(\boldsymbol{\nu}^*)\boldsymbol{\nu}^*) \quad (51)$$

$$\dot{\boldsymbol{\nu}}_{max} = \mathbf{M}^{-1}(\tau(\boldsymbol{\nu}^*, \mathbf{n}_{max}, \boldsymbol{\delta}_{min}) - \mathbf{C}(\boldsymbol{\nu}^*)\boldsymbol{\nu}^* - \mathbf{D}(\boldsymbol{\nu}^*)\boldsymbol{\nu}^*). \quad (52)$$

It should be noted that this method does not consider actuator rate saturations. However, by introducing dynamics to the control input vector τ , the algorithm can also be further developed to handle rate constraints. Additionally, it is worth noticing that a positive rudder deflection results in a negative yaw moment.

Using T as the time allowed for acceleration during the next time step, the dynamic velocity window is then defined using the acceleration limits from (51) and (52) as

$$V_w = \{(u, r) \in \mathbb{R} \times \mathbb{R} | u \in [u^* + \dot{u}_{min}T, u^* + \dot{u}_{max}T] \wedge r \in [r^* + \dot{r}_{min}T, r^* + \dot{r}_{max}T]\}. \quad (53)$$

The set of dynamically feasible velocities is defined as

$$V_f \triangleq V_p \cap V_w. \quad (54)$$

Next, the set of dynamically feasible velocities V_f is discretized uniformly to obtain a discrete set of dynamically feasible velocity pairs.

The desired velocity is defined as

$$\boldsymbol{\nu}_{1d} \triangleq [u_d, r_d]^\top, \quad (55)$$

since the focus is on controlling the surge and yaw rate. Given $\boldsymbol{\nu}_{1t}$, the optimal velocity pair $\boldsymbol{\nu}_{1f} = [u_f, r_f]^\top$ can be

selected as

$$\boldsymbol{\nu}_{1f} = \arg \max_{(u,r) \in V_f} G(u, u_d, r, r_d), \quad (56)$$

where $G(u, u_d, r, r_d)$ is an objective function, which is defined as

$$G(u, u_d, r, r_d) \triangleq \text{surge}(u, u_d) + \text{yawrate}(r, r_d), \quad (57)$$

with

$$\text{surge}(u, u_d) = 1 - \frac{|u_d - u|}{\max_{u' \in V_f} (|u_d - u'|)} \in [0, 1] \quad (58)$$

$$\text{yawrate}(r, r_d) = 1 - \frac{|r_d - r|}{\max_{r' \in V_f} (|r_d - r'|)} \in [0, 1]. \quad (59)$$

Notice that by using this objective function, we minimise the scaled 1-norm of the entire discrete set of dynamically feasible velocity pairs. In [3], [4] and [10], a distance function and tuning parameters are used to achieve collision avoidance, but this function is removed here since we only focus on handling actuator constraints. As a result, the tuning parameters also become redundant since the remaining two functions are orthogonal to each other. Fig. 4 illustrates V_p , V_w , V_f and $\boldsymbol{\nu}_{1d} = [0.79 \text{ m/s } 4.0107 \text{ deg/s}]$ given a current velocity pair of $[0.76 \text{ m/s } 5.6723 \text{ deg/s}]$.

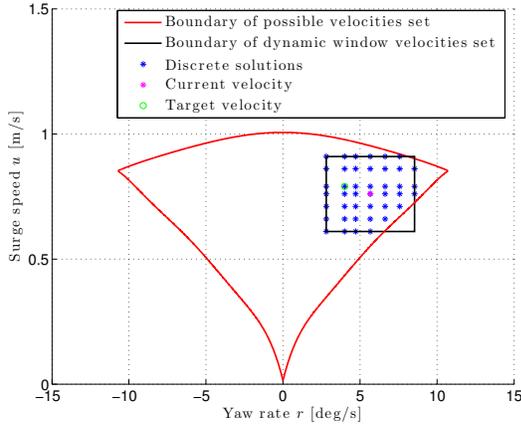


Fig. 4: The dynamically feasible velocity set, surrounded by the boundaries of the dynamic velocity window and the possible velocity set.

B. Dynamic Window-based Controller

We now combine the traditional control design with the simplified DW algorithm in order to develop a dynamic window-based controller (DWC).

In this setup, the simplified DW algorithm will use $\boldsymbol{\alpha}_1 = [\alpha_u, \alpha_r]^\top$ as an input such that $\boldsymbol{\nu}_{1d} = \boldsymbol{\alpha}_1$. In the case where $\boldsymbol{\alpha}_1$ is an infeasible velocity, the simplified DW algorithm will modify $\boldsymbol{\alpha}_1$ to a feasible velocity $\boldsymbol{\alpha}_{1f} = [\alpha_{f,u}, \alpha_{f,r}]^\top$, otherwise $\boldsymbol{\alpha}_{1f} = \boldsymbol{\alpha}_1$. A pseudocode of the simplified DW algorithm is shown in Algorithm 1.

Algorithm 1 Pseudocode of the simplified DW algorithm

- 1: V_w is calculated using (51) to (53) and discretized uniformly
- 2: **if** the desired velocity vector $\boldsymbol{\alpha}_1 \in V_f$ **then**
- 3: The closest reachable velocity row and column to $\boldsymbol{\alpha}_1$ is shifted such that $\boldsymbol{\alpha}_1$ is one of the reachable velocity pairs in V_w
- 4: **end if**
- 5: Remove all the reachable velocity pairs in V_w which are outside of the $g(u, r)$ boundaries to describe the set of dynamically feasible velocities V_f .
- 6: Select the optimal velocity pair $\boldsymbol{\alpha}_{1f}$ through maximizing the objective function (57) over the discrete feasible search space $V_f = V_p \cap V_w$

After the optimal velocity pair $\boldsymbol{\alpha}_{1f}$ is found, the vector of stabilising functions is given as

$$\boldsymbol{\alpha}_f = [\alpha_{f,u}, \alpha_{f,v}, \alpha_{f,r}]^\top, \quad (60)$$

where $\alpha_{f,v}$ is given as

$$\alpha_{f,v} = \int_0^t \dot{\alpha}_{f,v}(\sigma) d\sigma, \quad \alpha_{f,v}(0) = v(0), \quad (61)$$

where

$$m_{22} \dot{\alpha}_{f,v} = -d_{22}(\boldsymbol{\nu}) \alpha_{f,v} + \gamma(\alpha_r, \dot{\alpha}_r) \quad (62)$$

$$\gamma(\alpha_r, \dot{\alpha}_r) = -m_{23} \dot{\alpha}_r - d_{23}(\boldsymbol{\nu}) \alpha_r - c_{23}(\boldsymbol{\nu}) \alpha_r. \quad (63)$$

We want the ship to reach $\boldsymbol{\alpha}_f$ after time T , hence the desired acceleration is chosen to be

$$\dot{\boldsymbol{\alpha}}_{DWC} = \frac{\boldsymbol{\alpha}_f - \boldsymbol{\nu}}{T}, \quad (64)$$

which means that

$$\boldsymbol{\alpha}_{DWC} = \int_0^t \dot{\boldsymbol{\alpha}}_{DWC} d\sigma. \quad (65)$$

Both $\boldsymbol{\alpha}_{DWC}$ and $\dot{\boldsymbol{\alpha}}_{DWC}$ are used in the kinetic controller which is modified to

$$\boldsymbol{\tau} = \mathbf{M} \dot{\boldsymbol{\alpha}}_{DWC} + \mathbf{C}(\boldsymbol{\nu}) \boldsymbol{\alpha}_{DWC} + \mathbf{D}(\boldsymbol{\nu}) \boldsymbol{\alpha}_{DWC}. \quad (66)$$

The DWC uses the heading controller given in (35) together with the target speed u_t as inputs to the simplified DW algorithm, which is described in Algorithm 1, in order to determine the vector of stabilizing functions given in (60)-(63). Based on (60), the desired acceleration and velocity vectors are found using (64)-(65), which are used to construct the control input (66).

A block diagram of the new dynamic window-based controller is shown in Fig. 5.

C. Discussion

When comparing the control law in (66) against (31), it can be seen that the feedback term $-\mathbf{K}_2(\cdot) \mathbf{z}_2$ in (31) is not included in (66) since the DWC makes the optimal velocity pair track the target velocity by using (64)-(65). However, (66) can only fulfil the control objective when the

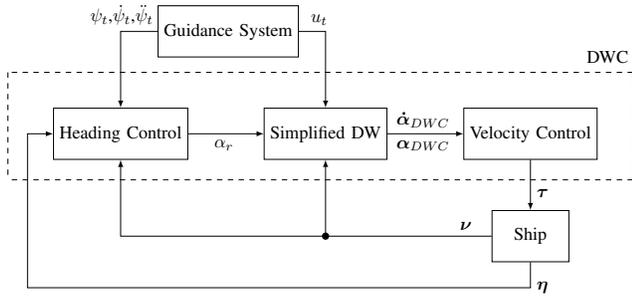


Fig. 5: Schematic structure of the dynamic window-based controller.

model is correct. In practice, when the model is not perfectly known, the control law (66) should also include a feedback term. This feedback term will also accommodate for internal uncertainties and external disturbances, instead of just controlling the surge speed and yaw rate. When the system is affected by internal uncertainties and external disturbances, the DWC will attempt to compensate for them since the DWC tries to find the optimal velocity pair. However, the performance in terms of robustness for the DWC is limited by the actuator constraints, which give a maximum bound on the uncertainties and disturbances which the controller can compensate for. This is similar to many robust controllers.

VI. SIMULATION AND EVALUATION

In this section, we present numerical simulation results of the considered heading and speed controllers using the model and actuator constraints of CyberShip II presented in Section II. In addition, performance metrics are used to evaluate the controller behavior.

The control target is defined as a constant heading $\psi_t = 30$ deg and a constant surge speed $u_t = 0.9$ m/s. Furthermore, the initial ship states are chosen to be $\boldsymbol{\eta}(0) = \mathbf{0}$ and $\boldsymbol{\nu}(0) = \mathbf{0}$. The chosen control gains are listed in Table II.

TABLE II: Control gains

	TC	DWC
Γ_1	0.0873	0.0873
Γ_2	$\text{diag}([4, 4, 0.1745])\mathbf{M}$	—
$\Delta_{\tilde{\psi}}$	0.3	0.3
$\Delta_{\tilde{v}}$	10	—
$\Delta_{\tilde{r}}$	4	—

A. Performance Metrics

To evaluate and compare the performance of the two controllers, performance metrics are used. We define

$$e_1(t) \triangleq \sqrt{\tilde{\psi}^2} \quad (67)$$

$$e_2(t) \triangleq \sqrt{\tilde{u}_n^2 + \tilde{r}_n^2}, \quad (68)$$

as the error inputs for the performance metrics, with $\tilde{u}_n \triangleq u_n - u_{t,n}$ and $\tilde{r}_n \triangleq r_n - r_{t,n}$. Here, since the surge speed

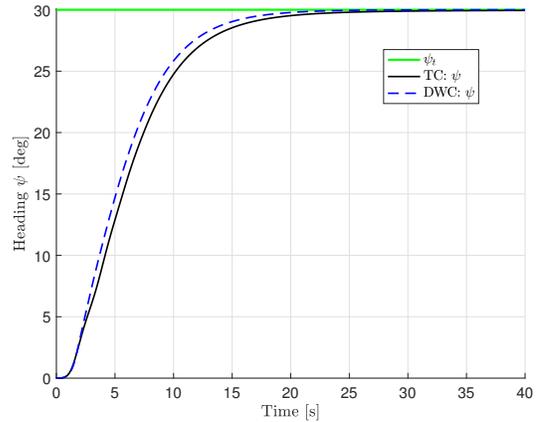


Fig. 6: Tracking the target heading.

and yaw rate have different units, we define the normalized signals u_n , $u_{t,n}$, r_n and $r_{t,n}$ in the intervals $[0, 1]$ and $[-0.5, 0.5]$ in the expected operational space of the ship [8]. In addition, these signals represent the instantaneous control errors, while we would like to consider the accumulated errors over time. Hence, we use the performance metric IAE (integral of the absolute error)

$$IAE(e_i, t) \triangleq \int_0^t |e_i(\sigma)| d\sigma, \quad (69)$$

which integrates the temporal evolution of the absolute error. We also consider the integral of the absolute error multiplied by the energy consumption (IAEW), which was proposed earlier in [11] as

$$IAEW(e_i, t) \triangleq \int_0^t |e_i(\sigma)| d\sigma \int_0^t P(\sigma) d\sigma, \quad (70)$$

where

$$P(t) = |\boldsymbol{\nu}(t)^\top \boldsymbol{\tau}(t)| \quad (71)$$

represents the mechanical power. IAEW thus indicates which controller has the best combined control accuracy and energy use in one single metric.

B. Simulation Results

In Fig. 6, the ship and target heading is plotted to show the transient convergence behavior. It can be seen that both control laws manage to converge the target heading in about 25 seconds, but the DWC gives a slightly faster convergence.

Fig. 7 shows the surge speed and yaw rate of the ship together with the target surge speed and yaw rate. It can be seen that both control laws are able to track the target surge speed and yaw rate even though the DWC does not have a traditional velocity feedback term. Additionally, it can be seen that there is a difference in how fast the controllers are able to make the surge speed and yaw rate converge to the target since the DWC makes the surge speed converge in less than 5 seconds but it takes about 10 seconds for the surge speed to converge using the TC. Notice that the trajectory

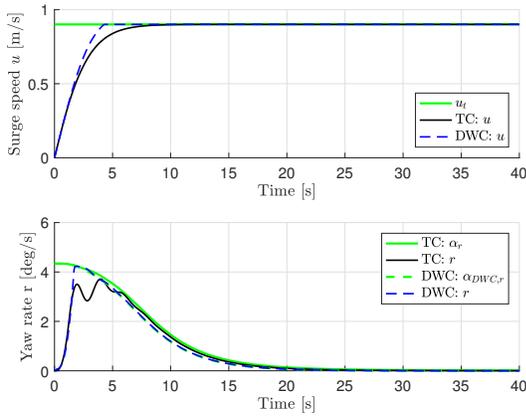


Fig. 7: Tracking of the surge speed (top) and yaw rate (bottom).

of α_r is different for the two controllers since it depends on the tracking performance of the heading.

Fig. 8 shows that the DWC commands the control inputs to stay at the maximum magnitude constraints of the actuators for a longer time than the control inputs from TC. These constraints (red lines) are calculated by using (20)-(24) and the limits of n_i and δ_i given in (46)-(47). The DWC keeps the control inputs at the maximum magnitude constraints of the actuators as long as possible, since the DWC tracks the optimal velocity pair α_{1f} which is on the boundaries for the window unless the target velocity pair α_1 is inside the velocity window, while the control inputs from TC have a more conservative behavior. The oscillations in the yaw moment control input of the TC is a side-effect the TC is because it tries to compensate for the nonlinear magnitude constraint in the yaw moment.

Fig. 9 illustrates how the surge speed and yaw rate moves in the velocity space in order to track the target heading and surge speed. Note that the considered controllers move along two different trajectories inside V_p in order to solve the same control problem.

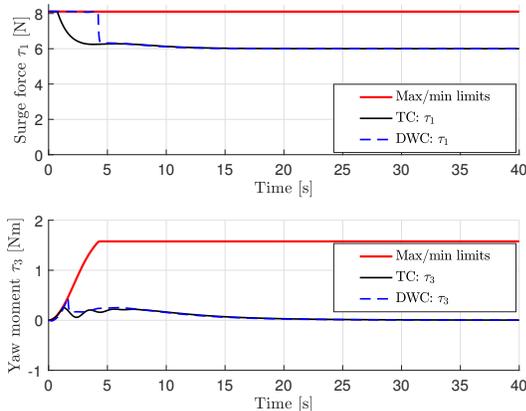


Fig. 8: The commanded surge force and yaw moment with magnitude limits.

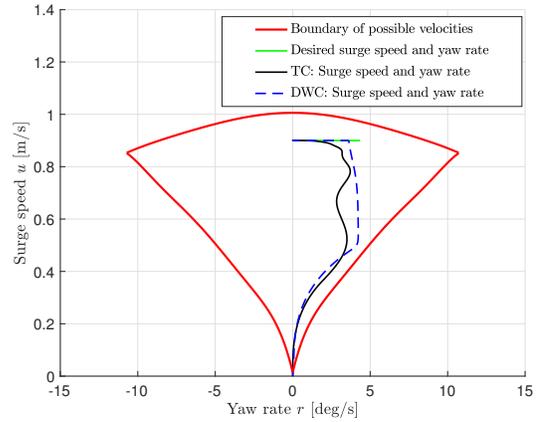


Fig. 9: Velocity trajectories in the set of possible velocities V_p , where the target heading and target speed are $\psi_t = 30$ deg and $u_t = 0.9$ m/s.

In Fig. 10, the performance metrics IAE and IAEW with e_1 as the error input are shown. In particular, the IAE trajectory in the top of Fig. 10 confirms that the DWC has a faster transient response since it converges faster to a stationary value. The IAEW trajectory in the bottom of Fig. 10 shows that the DWC uses a larger amount of energy to fulfil the control objective. However, the DWC has a faster transient response, which makes the DWC have a better overall performance.

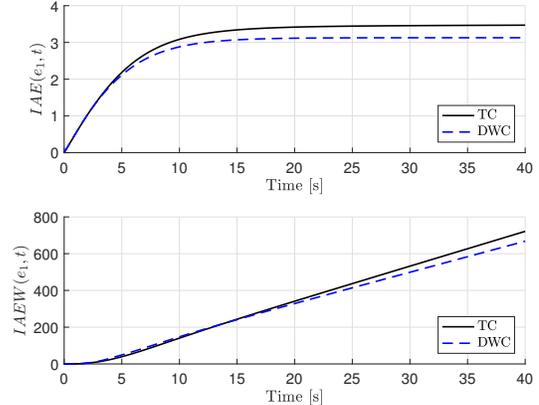


Fig. 10: IAE and IAEW performance metrics with $e_1 = \sqrt{\hat{\psi}^2}$ as the error input.

Fig. 11 displays the performance metrics IAE and IAEW with e_2 as the error input, where a similar result as in Fig. 10 can be seen. Based on Fig. 10 and Fig. 11, it can be concluded that DWC has the better overall control performance for this scenario.

The presented results show that both the TC and the DWC stay within the boundaries in Fig. 9 and the tracking performance of the DWC is slightly faster, even though the DWC has fewer tuning parameters than the TC. It should be stated that the TC has been extensively tuned to get the optimal performance for this scenario and remain inside

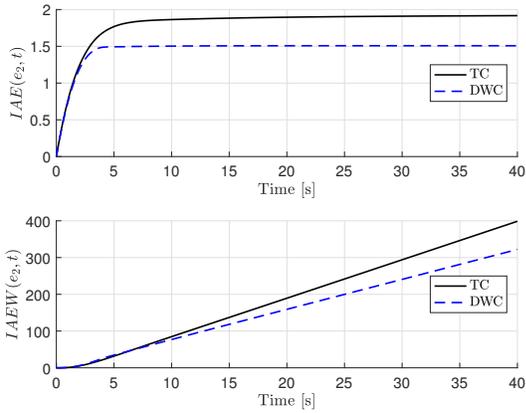


Fig. 11: IAE and IAEW performance metrics with $e_2 = \sqrt{\hat{u}_n^2 + \hat{r}_n^2}$ as the error input.

velocity boundaries, but it still falls short. For experimental purposes, it is suggested to add a feedback term to (66) in order to accommodate for model uncertainties and external disturbances.

One of the advantages of using DWC can be shown by changing the target heading and target speed to $\psi_t = 90$ deg and $u_t = 0.6$ m/s, but keeping the control gains unchanged. Fig. 12 shows the velocity trajectories for this scenario, where it can be seen that the TC yields velocities outside the boundaries, which means that the TC does not inherently satisfy the actuator constraints, while the DWC continues to stay inside the boundaries. Additionally, the TC also has some unwanted oscillations.

VII. CONCLUSION

This paper has proposed the use of a simplified dynamic window algorithm as a way to ensure that the actuator constraints of a ship are satisfied. This algorithm has been used as a dynamic window-based controller (DWC) to guarantee

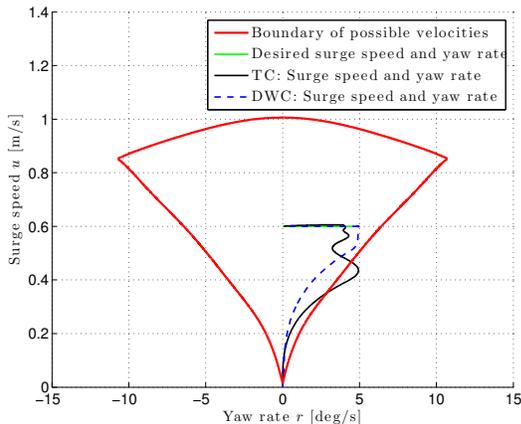


Fig. 12: Velocity trajectories in the set of possible velocities V_p , where the target heading and target speed are changed to $\psi_t = 90$ deg and $u_t = 0.6$ m/s while keeping the control gains unchanged.

that that ship velocities remain within a feasible set. An existing nonlinear dynamic model of a ship was modified to make it more physically realistic. Additionally, a DWC was evaluated against a heading and speed controller using a traditional design approach. Both methods were compared through numerical simulations. Two performance metrics were used to compare the behaviour of the controllers. The simulation results showed good tracking performance of the considered controllers, and that the dynamic window-based controller was able to inherently handle actuator magnitude constraints.

Future work will include introducing model uncertainties and unknown disturbances. It is also relevant to consider actuator rate constraints in addition to magnitude constraints.

ACKNOWLEDGEMENT

This work was supported by the Research Council of Norway through the Centres of Excellence funding scheme, project number 223254.

REFERENCES

- [1] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Wiley, 2011.
- [2] T. Lauvdal, *Stabilization of Linear Systems with Input Magnitude and Rate Saturations*. PhD thesis, Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway, 1998.
- [3] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.
- [4] B.-O. H. Eriksen, M. Breivik, K. Y. Pettersen, and M. S. Wiig, "A modified dynamic window algorithm for horizontal collision avoidance for AUVs," in *Proceedings of the IEEE Multi-Conference on Systems and Control, Buenos Aires, Argentina*, 2016.
- [5] M. E. N. Sørensen and M. Breivik, "Comparing combinations of linear and nonlinear feedback terms for motion control of marine surface vessels," in *Proceedings of the 10th IFAC Conference on Control Applications in Marine Systems, Trondheim, Norway*, 2016.
- [6] R. Skjetne, *The Maneuvering Problem*. PhD thesis, Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway, 2005.
- [7] J. E. Refsnes, *Nonlinear Model-Based Control of Slender Body AUVs*. PhD thesis, Department of Marine Technology, Norwegian University of Science and Technology, Trondheim, Norway, 2008.
- [8] B.-O. H. Eriksen and M. Breivik, *Modeling, Identification and Control of High-Speed ASVs: Theory and Experiments*, pp. 407–431. Cham: Sensing and Control for Autonomous Vehicles: Applications to Land, Water and Air Vehicles, Springer International Publishing, 2017.
- [9] T. I. Fossen, M. Breivik, and R. Skjetne, "Line-of-sight path following of underactuated marine craft," in *Proceedings of the 6th IFAC Conference on Manoeuvring and Control of Marine Craft, Girona, Spain*, 2003.
- [10] P. Ögren and N. E. Leonard, "A convergent dynamic window approach to obstacle avoidance," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 188–195, 2005.
- [11] M. E. N. Sørensen and M. Breivik, "Comparing nonlinear adaptive motion controllers for marine surface vessels," in *Proceedings of the 10th IFAC Conference on Manoeuvring and Control of Marine Craft, Copenhagen, Denmark*, 2015.