A Modified Dynamic Window Algorithm for Horizontal Collision Avoidance for AUVs

Bjørn-Olav H. Eriksen, Morten Breivik, Kristin Y. Pettersen, Martin S. Wiig

***Abstract*— Much research has been done on the subject of collision avoidance (COLAV). However, few results are presented that consider vehicles with second-order nonholonomic constraints, such as autonomous underwater vehicles (AUVs). This paper considers the dynamic window (DW) algorithm for reactive horizontal COLAV for AUVs, and uses the HUGIN 1000 AUV in a case study. The DW algorithm is originally developed for vehicles with first-order nonholonomic constraints and is hence not directly applicable for AUVs without resulting in degraded performance. This paper suggests further developments of the DW algorithm to make it better suited for use with AUVs. In particular, a new method for predicting AUV trajectories using a linear approximation which accounts for second-order nonholonomic constraints is developed. The new prediction method, together with a modified search space, reduces the mean square prediction error to about one percent of the original algorithm. The performance and robustness of the modified DW algorithm is evaluated through simulations using a nonlinear model of the HUGIN 1000 AUV.**

I. Introduction

Collision avoidance (COLAV) systems are necessary for autonomous operation of vehicles, including autonomous underwater vehicles (AUVs). AUVs are often engaged in long term operations in deep waters with limited communication possibilities, which increase the reliability requirements of the COLAV system. It is clear that if a collision occurs and the AUV is immobilized, a salvage operation will be both costly and time consuming. In addition, a delayed operation may potentially have large economical consequences.

The topic of COLAV may be split in two main areas [1]:

- *Obstacle detection*, which focuses on detecting obstacles, usually based on sonar data regarding AUVs.
- *Obstacle avoidance*, which consists of generating appropriate steering commands in order to avoid collisions with detected obstacles.

A *COLAV system* must include both obstacle detection and avoidance in order to avoid collisions. This paper will only focus on obstacle avoidance. For details on how obstacle detection can be handled, see [2] and the references therein.

There exists a number of both *reactive* (local) and *deliberate* (global) COLAV algorithms. Reactive algorithms base their actions only on real-time sensor data, which makes the algorithms computationally inexpensive and suitable for reacting to sudden changes in the environment. Using no a priori information can, however, make the vehicle sensitive to local minima or traps, which can make it unable to reach the goal. In comparison, deliberate algorithms include a priori information to plan future actions. The provided actions are more likely to make the vehicle converge towards the goal, at the cost of increased computational time and reduced ability to react rapidly. Reactive and deliberate algorithms are often combined in *hybrid* architectures, where they are executed in parallel at different sampling frequencies [3], [4].

One of the existing reactive[1] algorithms is the dynamic window (DW) algorithm [5], which was originally intended for vehicles with first-order nonholonomic constraints and constant acceleration limits. AUVs, however, have second-order nonholonomic constraints, and also nonlinear responses which result in time-varying acceleration limits. Applying the original DW algorithm to AUVs will thus result in degraded performance. This paper therefore suggests a number of modifications to the original DW algorithm to increase the performance when applied to vehicles with second-order nonholonomic constraints and time-varying acceleration limits. In particular, a new trajectory prediction method taking second-order nonholonomic constraints into account is developed. This, together with a modified search space, reduces the mean square prediction error to about one percent of the original algorithm. The generality of the algorithm is also improved to facilitate a more layered architecture.

A comparison between the modified and original DW algorithm is obtained through simulations for various static obstacle environments using a nonlinear model of the HUGIN 1000 AUV [6], shown in Figure 1. The simulations show a significant performance improvement when applying the modified DW algorithm.

In Section II, a 3 degrees-of-freedom (DOF) control model is presented. Section III contains the DW algorithm and the proposed modifications. Simulation results are presented in Section IV, while Section V concludes the paper.

II. Control-oriented AUV model

In this section, a generic 3 DOF AUV control model is defined. The model is based on the following assumptions:

Assumption 1: The AUV model describes the motion of the *pivot point* of the vehicle.

Bjørn-Olav H. Eriksen, Morten Breivik and Kristin Y. Pettersen 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: {bjorn-olav.h.eriksen, morten.breivik}@ieee.org and kristin.y.pettersen@itk.ntnu.no

Martin S. Wiig is with the Centre for Autonomous Marine Operations and Systems, Department of Engineering Cybernetics, NTNU, NO-7491 Trondheim, Norway and the Norwegian Defence Research Establishment, NO-2027 Kjeller, Norway. Email: Martin-Syre.Wiig@ffi.no

[1] One may argue that the DW algorithm is not strictly reactive since it plans trajectories in time. It does, however, only rely on real-time sensor data, thus it is considered to be a reactive algorithm.

Fig. 1. The HUGIN 1000 AUV, courtesy of the Norwegian Defence Research Establishment.

Remark 1: When the body-fixed coordinate system is positioned in the pivot point, the rudders do not affect the sway dynamics directly. For vehicles which are controllable in yaw it is always possible to transform a model represented in an arbitrary point on the AUV to the pivot point [7].

Assumption 2: The vehicle is neutrally buoyant and the center of gravity (CG) is located below the center of buoyancy (CB) on a vertical line.

Assumption 3: Heave speed and the roll and pitch angles are assumed equal to zero.

Remark 2: Assuming zero roll angle is a common assumption for slender body vehicles such as AUVs [8]. The CG is located below the CB on a vertical line, which passively stabilizes the roll motion. Further, by equally utilizing the top and bottom rudders, and the port and starboard rudders, no roll moment is generated by the rudders. For an AUV equipped with a depth controller, close to zero heave speed and pitch angle is achieved when the AUV is traveling in a horizontal plane.

Assumption 3 allows a control model to be formulated in 3 DOF (surge, sway and yaw), while Assumption 2 implies that no restoring forces are applied. Using the SNAME [9] notation, the 3 DOF control model is therefore given as:

$$\dot{\boldsymbol{\eta}}^n_{b/n} = \boldsymbol{R}(\boldsymbol{\eta}^n_{b/n})\boldsymbol{\nu}^b_{b/n} \tag{1a}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}}^b_{b/n} + \boldsymbol{C}(\boldsymbol{\nu}^b_{b/n})\boldsymbol{\nu}^b_{b/n} + \boldsymbol{D}(\boldsymbol{\nu}^b_{b/n})\boldsymbol{\nu}^b_{b/n} = \boldsymbol{B}\boldsymbol{f}^b_b, \tag{1b}$$

where $\boldsymbol{\eta}^n_{b/n} = \begin{bmatrix} x & y & \psi \end{bmatrix}^T \in \mathbb{R}^2 \times SO(2)$ is the position and orientation of the body-fixed frame $\{b\}$ represented in the north-east-down-fixed inertial frame $\{n\}$. Further, $\boldsymbol{\nu}^b_{b/n} = \begin{bmatrix} u & v & r \end{bmatrix}^T \in \mathbb{R}^3$ is the velocity of $\{b\}$ with respect to $\{n\}$ represented in $\{b\}$, and $\boldsymbol{f}^b_b = \begin{bmatrix} X & N \end{bmatrix}^T \in \mathbb{R}^2$ is the actuator input in $\{b\}$. It should be noted that the model (1) does not account for external forces such as ocean currents, wind and waves. For notational simplicity, $\boldsymbol{\eta}^n_{b/n}, \boldsymbol{\nu}^b_{b/n}, \boldsymbol{f}^b_b$ are further denoted as $\boldsymbol{\eta}, \boldsymbol{\nu}, \boldsymbol{f}$. The transformation matrix from $\{b\}$ to $\{n\}$ is given as:

$$\boldsymbol{R}(\boldsymbol{\eta}) = \boldsymbol{R}(\psi) = \begin{bmatrix} c(\psi) & -s(\psi) & 0 \\ s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2a}$$

where $c(\cdot) = \cos(\cdot)$ and $s(\cdot) = \sin(\cdot)$. The vessel dynamics matrices are given as:

$$\boldsymbol{M} = \begin{bmatrix} m_{11} & 0 & 0 \\ 0 & m_{22} & m_{23} \\ 0 & m_{23} & m_{33} \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} b_{11} & 0 \\ 0 & b_{22} \\ 0 & b_{32} \end{bmatrix}$$

$$\boldsymbol{C}(\boldsymbol{\nu}) = \begin{bmatrix} 0 & 0 & -m_{22}v - m_{23}r \\ 0 & 0 & m_{11}u \\ m_{22}v + m_{23}r & -m_{11}u & 0 \end{bmatrix} \tag{2b}$$

$$\boldsymbol{D}(\boldsymbol{\nu}) = -\begin{bmatrix} X_u + X_{|u|u}|u| & 0 & 0 \\ 0 & Y_v & Y_r \\ 0 & N_v & N_r \end{bmatrix} \frac{|u'|}{u_0}.$$

Here, the term $\frac{|u'|}{u_0}$, where $u_0 > 0$ is the nominal surge speed, is used for speed-scaling of the damping coefficients. The term $|u'| \triangleq \max(|u|, \mu)$, where $\mu > 0$ is an arbitrary constant, guarantees some damping for low surge speeds. The constants in $\boldsymbol{B}$ are given as $b_{11} = b_{32} = 1$, while b_{22} captures the coupling from the yaw torque to the sway force in the actuator model, given from $Y = -\frac{1}{l_x}N \triangleq b_{22}N$. The force vector $\boldsymbol{f}$ is modeled as:

$$\boldsymbol{f} = \begin{bmatrix} X \\ N \end{bmatrix} = \begin{bmatrix} T_{|n|n}|n_p|n_p + T_{un}un_p \\ -Y_{\delta u^2} l_x \delta_\psi \end{bmatrix}, \tag{3}$$

where n_p and δ_ψ are the propeller speed and rudder deflection angle, respectively, while $T_{|n|n}$, T_{un} and $Y_{\delta u^2}$ are propeller and rudder coefficients, and l_x is the rudder lever arm along the x-axis. The actuators are limited by both saturation and rate limitations:

$$n_p \in [n_{p_{\min}}, n_{p_{\max}}], \ \|\delta_\psi\|_\infty \leq \delta_{\max}, \ \left\|\dot{\delta}_\psi\right\|_\infty \leq \dot{\delta}_{\max}. \tag{4}$$

The actuator limitations are not considered in the mathematical model (1) since $\boldsymbol{f}$ is selected as the control input, but are, however, to be included in the DW algorithm. For positive surge speeds, and requiring that the DW algorithm only specifies feasible commands, it is always possible to calculate n_p and δ_ψ given a force vector $\boldsymbol{f}$. Hence, a unique inverse function $\boldsymbol{f}^{-1}$ exists.

It should be noted that when Assumption 1 is satisfied, the following property holds [7]:

$$\boldsymbol{M}^{-1}\boldsymbol{B}\boldsymbol{f} = \begin{bmatrix} \frac{b_{11}}{m_{11}}X \\ 0 \\ \frac{m_{22}b_{32} - m_{23}b_{22}}{m_{22}m_{33} - m_{23}^2}N \end{bmatrix}. \tag{5}$$

III. The Dynamic Window Algorithm

A. Introduction

The DW algorithm is a velocity space method, intended to prohibit infeasible control commands by specifying a desired *velocity pair* consisting of a desired forward speed and a desired rotational rate as reference signals for the vehicle speed controllers. The algorithm was originally designed for a car-like mobile robot with first-order nonholonomic constraints, moving in 3 DOF [5]. The original paper predicts vehicle trajectories as circular arcs with radii of $M^i_R = \frac{u_i}{r_i}$ (and straight lines for $r_i = 0$) for a discrete set of desired velocity pairs (u_i, r_i). The original trajectory prediction is

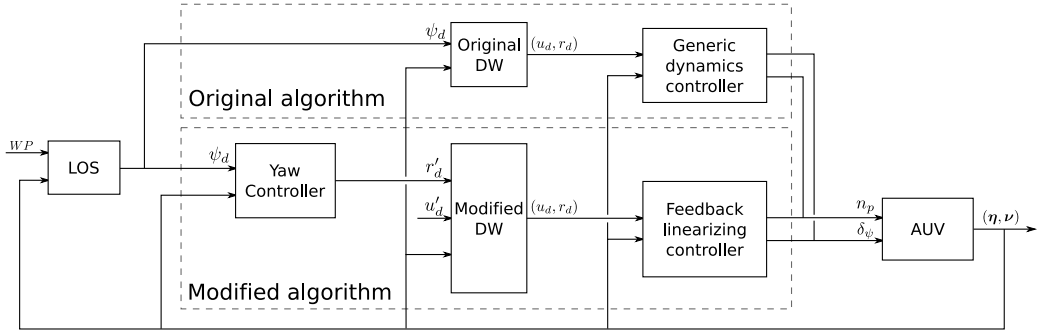

Fig. 2. Architecture overview. The modified algorithm facilitates a more modular architecture than the original algorithm.

quite accurate for vehicles with only first-order nonholonomic constraints, since they have no sideways motion.

As seen in Figure 2, the original DW algorithm inputs a desired heading ψ_d to guide the vehicle towards a goal. In contrast, the modified algorithm inputs a desired surge speed and yaw rate (u'_d, r'_d), which facilitates a more modular architecture in addition to allowing the surge speed to be externally guided. Notice that (u'_d, r'_d) is a desired velocity pair used as an *input* to the modified algorithm, while (u_d, r_d) is the *output* of the algorithm used as the reference for the vehicle dynamics controller.

Here, the DW algorithm is combined with a line-of-sight (LOS) guidance law, given as [8]:

$$\psi_{LOS} = \alpha_p - \arctan\left(\frac{e}{\Delta}\right), \tag{6}$$

where α_p is the path heading and e is the cross track error. The tuning parameter $\Delta > 0$ is the lookahead-distance, given in meters. To generate a desired yaw rate for the modified DW algorithm, the following yaw controller is proposed:

$$r'_d = -k_\psi \left(\psi - \psi_d\right) + \dot{\psi}_d, \tag{7}$$

where $k_\psi > 0$ is a constant gain and $\psi_d = \psi_{LOS}$.

For an implementation of the DW algorithm with an integral line-of-sight (ILOS) guidance law for compensation of ocean currents, also including a proof of convergence to a straight line path for the combined system, see [2]. It should be noted that the case presented in this paper is a special case of the case considered in [2], hence the convergence proof also applies to the system presented here.

B. The original dynamic window algorithm

Three 2D search spaces in forward (surge) speed and rotational (yaw) rate accounts for the kinematic and kinetic constraints of the vehicle. The *dynamic window* allows a time interval T (usually smaller than the sample time) for acceleration of the vehicle:

$$V_d = \Big\{(u,r) \in \mathbb{R}\times\mathbb{R} \Big| u \in [u^* + \dot{u}_{\min}T, u^* + \dot{u}_{\max}T] \wedge r \in [r^* - \dot{r}_{\max}T, r^* + \dot{r}_{\max}T]\Big\}, \tag{8}$$

where u^*, r^* are the current forward speed and rotational rate and $\dot{u}_{\min} < 0, \dot{u}_{\max} > 0, \dot{r}_{\max} > 0$ are the vehicle accelerations limits. Note that the original algorithm assumes that the yaw acceleration limits are symmetric, hence $\dot{r}_{\min} = -\dot{r}_{\max}$. The set of *possible velocities* is:

$$V_s = \Big\{(u,r) \in \mathbb{R}\times\mathbb{R} \Big| u \in [0, u_{\max}] \wedge r \in [-r_{\max}, r_{\max}]\Big\}, \tag{9}$$

where $u_{\max}$ and $r_{\max}$ are the maximum forward speed and rotational rate. The dynamic window and the set of possible velocities account for the actuator limitations. Finally, the set of *admissible velocities* ensures that the vehicle is able to stop before it collides with an obstacle:

$$V_a = \Big\{(u,r) \in \mathbb{R}\times\mathbb{R} \Big| u \le \sqrt{2\cdot \text{dist}(u,r)\cdot|\dot{u}_{\min}|} \wedge |r| \le \sqrt{2\cdot \text{dist}(u,r)\cdot \dot{r}_{\max}}\Big\}, \tag{10}$$

where $\text{dist}(u,r)$ expresses the distance which the vehicle can travel along the trajectory given the velocity pair (u,r) without colliding with an obstacle.

The optimal velocity pair is selected through maximizing an objective function over the resulting search space $V_r = V_d \cap V_s \cap V_a$:

$$\begin{aligned} \max_{(u,r)} \; & G(u,r) = \sigma\left(\alpha\cdot\text{heading}(u,r) + \beta\cdot\text{dist}(u,r) + \gamma\cdot\text{velocity}(u,r)\right) \\ \text{s.t. } & (u,r) \in V_r, \end{aligned} \tag{11}$$

where $\alpha, \beta, \gamma > 0$ are tuning parameters and $\text{heading}(u,r)$ measures the alignment between the trajectory corresponding to the velocity pair (u,r) and a desired heading. The term $\text{velocity}(u,r)$ favors holding a high surge speed, while σ is a low-pass filter to reduce fluctuations in the control output. The optimization problem (11) is solved by numerically computing the objective value for all $(u,r) \in V_r$ (discretized) and selecting the one with the highest objective value.

The original algorithm has two significant limitations when applied to AUVs:

- The lack of modeling the sideways motion of vehicles with second-order nonholonomic constraints results in inaccurate trajectory predictions.
- Using a rectangular search space which does not include any actuator modeling can cause infeasible control references to be specified.

Furthermore, the *heading* term in (11) compares a desired heading input with a resulting heading for the trajectories, requiring the algorithm to include a mapping function to compute the resulting heading for the different velocity pairs.

Several modifications to the original DW algorithm have been proposed. Some of these include global information in order to handle local minima in the environment, see [10], [11], [12], [13]. In particular, [13] combines the focussed D* (FD*) [14] global planner with the DW algorithm in a hybrid architecture. The *heading* and *velocity* terms are replaced by a measure of alignment to the FD* path, hence also removing the required mapping from angle to angular rate. To reduce the prediction error of the vehicle trajectories, [15] uses clothoid curves instead of circular arcs and straight lines to model the vehicle trajectories. A simplified set of equations is used in [4] to simulate trajectories for an autonomous surface vehicle (ASV), accounting for an estimated lateral speed.

C. A new search space and objective function

In this section we present a modified search space to better suit the nonlinear responses of AUVs. The objective function is also changed to facilitate a more layered architecture.

To ensure feasible steering commands for the AUV, the search space is modified to account for the actuator model (3) and limitations (4). Allowing a small time interval $T_a < T$ to be used for changing the control inputs, the feasible actuator commands are:

$$\begin{aligned} \delta_\psi &\in \text{sat}\left(\left[\delta_\psi^* - T_a\dot{\delta}_{\max}, \delta_\psi^* + T_a\dot{\delta}_{\max}\right], \delta_{\max}\right) \\ n_p &\in \left[n_{p_{\min}}, n_{p_{\max}}\right], \end{aligned} \tag{12}$$

where δ_ψ^* denotes the current rudder deflection angle, and $\text{sat}(\cdot)$ is a saturation function. It should be noted in general that rate limitation can also be imposed on the propeller speed. By denoting $\boldsymbol{Bf} = \boldsymbol{\tau}(\boldsymbol{\nu}, \delta_\psi, n_p)$, and since the actuators are linearly independent, the acceleration limits can be found by solving:

$$\dot{\boldsymbol{\nu}}_i = \boldsymbol{M}^{-1}\left(\boldsymbol{\tau}_i - \boldsymbol{C}(\boldsymbol{\nu}^*)\boldsymbol{\nu}^* - \boldsymbol{D}(\boldsymbol{\nu}^*)\boldsymbol{\nu}^*\right), \tag{13}$$

where $i \in \{\min, \max\}$, $\boldsymbol{\tau}_{\min} \triangleq \boldsymbol{\tau}(\boldsymbol{\nu}^*, \max(\delta_\psi), \min(n_p))$, $\boldsymbol{\tau}_{\max} \triangleq \boldsymbol{\tau}(\boldsymbol{\nu}^*, \min(\delta_\psi), \max(n_p))$ and $\boldsymbol{\nu}^*$ denotes the current vehicle velocity. It is worth noticing that a positive rudder deflection results in negative yaw moment. The dynamic window (8) is then modified as:

$$\begin{aligned} V_d = \Big\{(u,r) \in \mathbb{R}\times\mathbb{R} \Big| u \in [u^* + \dot{u}_{\min}T, u^* + \dot{u}_{\max}T] \\ \wedge r \in [r^* + \dot{r}_{\min}T, r^* + \dot{r}_{\max}T]\Big\}. \end{aligned} \tag{14}$$

Notice that in contrast to the original algorithm the yaw rate acceleration limit is no longer assumed to be symmetric.

By defining a function $g(u,r)$ which is positive semidefinite for feasible velocities with respect to actuator saturations, the set of possible velocities (9) can be generalized as:

$$V_s = \Big\{(u,r) \in \mathbb{R}\times\mathbb{R} \Big| g(u,r) \geq 0\Big\}. \tag{15}$$

The function $g(u,r)$ is approximated by numerically calculating the boundaries of the possible steady state solutions of the kinetics (1b), see [2] for details. The sets V_d, V_s and the dynamically feasible velocity set $V_f = V_d \cap V_s$ are illustrated in Figure 3.

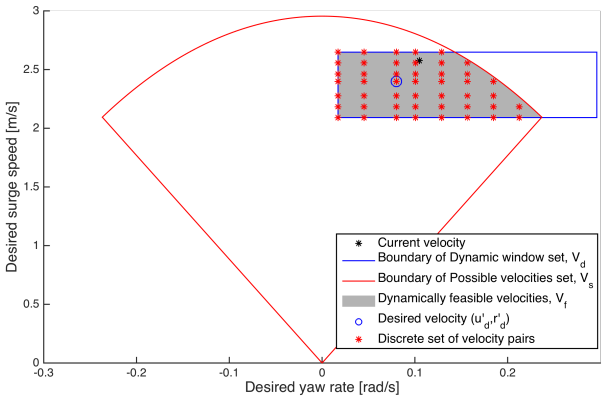

Fig. 3. The dynamically feasible velocity set V_f (in grey), together with the boundaries of the dynamic velocity window (in blue) and the possible velocity set (in red). The dynamically feasible velocity set is discretized uniformly, and note that the desired velocity pair (u'_d, r'_d) is included in the discrete search space.

To simplify the implementation, the configuration space of the AUV is reduced from $\mathbb{R}^2 \times SO(2)$ to $\mathbb{R}^2$ by approximating the AUV footprint as a circle. This is done by expanding the detected obstacles with the maximum AUV radius, and representing the AUV as a particle. Inspired by [16], two regions are defined on the detected obstacles to control the obstacle clearance; the "avoidance region"

$$\Omega \triangleq \left\{\boldsymbol{p} \in \mathcal{C} \middle| \|\boldsymbol{p} - \boldsymbol{p}_{obs}\|_2 \leq \bar{r}\right\}, \tag{16}$$

and the "antitarget region"

$$\mathcal{T} \triangleq \left\{\boldsymbol{p} \in \mathcal{C} \middle| \|\boldsymbol{p} - \boldsymbol{p}_{obs}\|_2 \leq r^*\right\} \tag{17}$$

where $\boldsymbol{p}_{obs} \in \mathbb{R}^2$ is the position of obstacles, $\mathcal{C} = \mathbb{R}^2$ is a collapsed, heading independent configuration space and $\bar{r} > r^* > 0$ are scalars defining the size of Ω and $\mathcal{T}$. In particular, r^* is the maximum radius of the AUV corresponding to approximating the AUV footprint as a circle. The antitarget region is interpreted as the region where a collision may occur, while the avoidance region is interpreted as a safety region that is not desirable to enter. The regions are illustrated in Figure 4.

The set of admissible velocities (10) is modified as:

$$\begin{aligned} V_a = \Big\{(u,r) \in \mathbb{R}\times\mathbb{R} \Big| u \leq \sqrt{2\rho'(u,r)|\dot{u}_{\min}|} \\ \wedge |r| \leq \left\{ \begin{array}{ll} \sqrt{2\rho'(u,r)|\dot{r}_{\max}|} & , r < 0 \\ \sqrt{2\rho'(u,r)|\dot{r}_{\min}|} & , r \geq 0 \end{array} \right\}. \end{aligned} \tag{18}$$

The function $\rho'(u,r)$ expresses the remaining distance the AUV can travel along the resulting trajectory at the next iteration without entering the antitarget region $\mathcal{T}$:

$$\rho'(u,r) = \max(\rho(u,r) - \Delta_s, 0), \tag{19}$$

where $\rho(u,r)$ expresses the distance the AUV can travel along the resulting trajectory before it enters $\mathcal{T}$ and Δ_s expresses the distance the AUV travels until the next iteration.

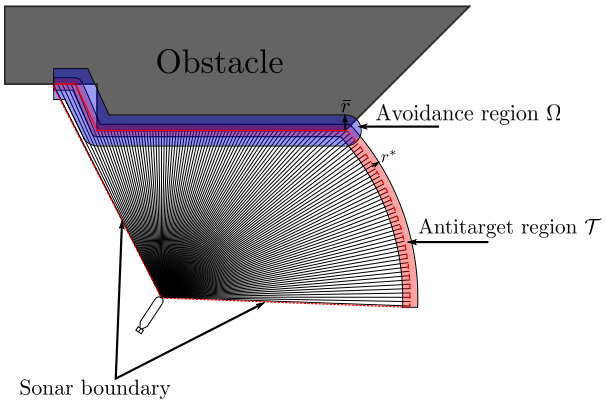

Fig. 4. Obstacle regions. Notice that the antitarget region is extended along the forward sonar boundary to account for possible obstacles just outside of the sonar range. Black lines illustrate sonar range measurements.

To improve the generality of the algorithm and remove the need for computing resulting headings for the velocity pairs, the *heading* term of the objective function is replaced with a term taking a desired yaw rate as input. This is inspired by [17] and [18]. In addition, inspired by [13], the $dist(u,r)$ term is scaled by the trajectory velocity, resulting in a term which approximates the time until collision. To motivate the algorithm to keep out of the avoidance region, $\text{dist}(u,r)$ expresses the time until the AUV enters Ω. The algorithm input is assumed to be smooth, so the low-pass filter is omitted. The objective function in (11) is thus modified as:

$$G(u,r) = \alpha\cdot\text{yawrate}(u,r,r'_d) + \beta\cdot\text{dist}(u,r) + \gamma\cdot\text{velocity}(u,r,u'_d), \quad (20)$$

where u'_d and r'_d are inputs to the algorithm, and

$$\text{yawrate}(u,r,r'_d) = 1 - \frac{|r'_d - r|}{\max\limits_{r\in V_r}(|r'_d - r|)}, \quad (21)$$

$$\text{velocity}(u,r,u'_d) = 1 - \frac{|u'_d - u|}{\max\limits_{u\in V_r}(|u'_d - u|)}, \quad (22)$$

$$\text{dist}(u,r) = \frac{\bar{\rho}(u,r)}{\frac{1}{T}\int_0^T \|\boldsymbol{\chi}(u,r,t)\|_2 \,\mathrm{dt}}, \quad (23)$$

where $\bar{\rho}(u,r)$ is the distance the AUV can travel along the trajectory specified by the velocity pair (u,r) until it enters Ω, and $\boldsymbol{\chi}(u,r,t)$ is the predicted AUV surge and sway speed along the trajectory specified by the velocity pair (u,r).

D. A new trajectory prediction method

To account for second-order nonholonomic constraints in the trajectory prediction, we propose to use partial feedback linearization to linearize the surge and yaw dynamics, while leaving the sway motion uncontrolled. The closed loop dynamics are then derived and used for predicting the AUV trajectories, hence including both sway and controller dynamics in the AUV trajectory prediction. This approach is similar to the one presented in [19], but does not require a linear model and is hence more flexible. In contrast to the approach suggested by [4], the actual equations of motion are used and the kinetics are solved analytically. This makes the prediction more accurate and requires less computations.

By solving (1b) for $\dot{\boldsymbol{\nu}}$, the system can be described as:

$$\dot{\boldsymbol{\nu}} = \boldsymbol{M}^{-1}\boldsymbol{B}\boldsymbol{f} - \boldsymbol{n}(\boldsymbol{\nu}), \quad (24)$$

where $\boldsymbol{n}(\boldsymbol{\nu}) = \boldsymbol{M}^{-1}\left(\boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu}\right)$.

To formulate the control law, the system is divided into two parts. This is done by using the matrices $\boldsymbol{\Gamma}_1$ and $\boldsymbol{\Gamma}_2$:

$$\boldsymbol{\Gamma}_1 \triangleq \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{\Gamma}_2 \triangleq \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, \quad (25)$$

that satisfies $\boldsymbol{\Gamma}_1^T\boldsymbol{\Gamma}_1 + \boldsymbol{\Gamma}_2^T\boldsymbol{\Gamma}_2 = \mathbf{I}$. The system (24) can be written as:

$$\begin{aligned}\dot{\boldsymbol{\nu}} &= \left(\boldsymbol{\Gamma}_1^T\boldsymbol{\Gamma}_1 + \boldsymbol{\Gamma}_2^T\boldsymbol{\Gamma}_2\right)\left(\boldsymbol{M}^{-1}\boldsymbol{B}\boldsymbol{f} - \boldsymbol{n}(\boldsymbol{\nu})\right) \\ &= \boldsymbol{\Gamma}_1^T\left(\boldsymbol{\Gamma}_1\boldsymbol{M}^{-1}\boldsymbol{B}\boldsymbol{f} - \boldsymbol{\Gamma}_1\boldsymbol{n}(\boldsymbol{\nu})\right) - \boldsymbol{\Gamma}_2^T\boldsymbol{\Gamma}_2\boldsymbol{n}(\boldsymbol{\nu}).\end{aligned} \quad (26)$$

Notice that $\boldsymbol{\Gamma}_2\boldsymbol{M}^{-1}\boldsymbol{B}\boldsymbol{f} = 0$. Hence, the system is divided in two parts where $\boldsymbol{\Gamma}_1^T\boldsymbol{\Gamma}_1$ maps dynamics to surge and yaw, while $\boldsymbol{\Gamma}_2^T\boldsymbol{\Gamma}_2$ maps dynamics to sway. To remove the nonlinearities in surge and yaw, we select the feedback linearizing control law:

$$\boldsymbol{f} = \left(\boldsymbol{\Gamma}_1\boldsymbol{M}^{-1}\boldsymbol{B}\right)^{-1}\left(\boldsymbol{\Gamma}_1\boldsymbol{n}(\boldsymbol{\nu}) + \boldsymbol{a}_{1d}\right), \quad (27)$$

where $\boldsymbol{a}_{1d} = \begin{bmatrix}\dot{u}_d & \dot{r}_d\end{bmatrix}^T$ is the desired acceleration.

Remark 3: By construction, it is shown in [2] that $\boldsymbol{\Gamma}_1\boldsymbol{M}^{-1}\boldsymbol{B}$ is of full rank and hence invertible.

Inserting (27) into (26) gives:

$$\dot{\boldsymbol{\nu}} = \boldsymbol{\Gamma}_1^T\boldsymbol{a}_{1d} - \boldsymbol{\Gamma}_2^T\boldsymbol{\Gamma}_2\boldsymbol{n}(\boldsymbol{\nu}). \quad (28)$$

The desired acceleration is selected using a proportional controller with a reference feedforward:

$$\begin{aligned}\boldsymbol{a}_{1d} &= \dot{\boldsymbol{\nu}}_{1d} - \boldsymbol{K}_p\left(\boldsymbol{\nu}_1 - \boldsymbol{\nu}_{1d}\right) \\ &= \dot{\boldsymbol{\nu}}_{1d} - \boldsymbol{K}_p\left(\boldsymbol{\Gamma}_1\boldsymbol{\nu} - \boldsymbol{\nu}_{1d}\right),\end{aligned} \quad (29)$$

where $\boldsymbol{K}_p = \begin{bmatrix} k_u & 0 \\ 0 & k_r \end{bmatrix} > 0$ is a gain matrix, $\boldsymbol{\nu}_1 = \begin{bmatrix} u & r \end{bmatrix}^T$ and $\boldsymbol{\nu}_{1d} = \begin{bmatrix} u_d & r_d \end{bmatrix}^T$.

Inserting (29) into (28), and defining

$$\tilde{\boldsymbol{\nu}} = \begin{bmatrix} \tilde{u} \\ v \\ \tilde{r} \end{bmatrix} \triangleq \boldsymbol{\nu} - \boldsymbol{\Gamma}_1^T\boldsymbol{\nu}_{1d}, \quad (30)$$

results in the dynamics:

$$\dot{\tilde{\boldsymbol{\nu}}} = -\boldsymbol{\Gamma}_1^T\boldsymbol{K}_p\boldsymbol{\Gamma}_1\tilde{\boldsymbol{\nu}} - \boldsymbol{\Gamma}_2^T\boldsymbol{\Gamma}_2\boldsymbol{n}(\boldsymbol{\nu}). \quad (31)$$

Note that $\boldsymbol{n}(\boldsymbol{\nu})$ takes $\boldsymbol{\nu}$ as its argument. Also, it is important to notice that (31) is linear in both surge and yaw:

$$\dot{\tilde{u}} = -k_u\tilde{u}, \quad \dot{\tilde{r}} = -k_r\tilde{r}, \quad \dot{v} = -n_2(\boldsymbol{\nu}), \quad (32)$$

where $n_2(\boldsymbol{\nu})$ is the contribution from the Coriolis-centripetal and damping matrices in sway, given as:

$$\begin{aligned}n_2(\boldsymbol{\nu}) = \frac{1}{m_{22}m_{33} - m_{23}^2}\Big(&\left(m_{33}d_{22} - m_{23}d_{32}\right)v \\ &- m_{23}\left(m_{22} - m_{11}\right)uv + \left(m_{33}m_{11} - m_{23}^2\right)ur \\ &+ \left(m_{33}d_{23} - m_{23}d_{33}\right)r\Big),\end{aligned} \quad (33)$$

where $m_{ij} = M_{i,j}$ and $d_{ij} = D_{i,j}$.

By approximating the last term of (31) using a first-order Taylor series, the dynamics of the AUV is also linear in sway. The Taylor approximation is given as:

$$\begin{aligned} \boldsymbol{n}(\boldsymbol{\nu}) &\approx \boldsymbol{n}(\boldsymbol{\nu}^*) + \underbrace{\left.\frac{\mathrm{d}\boldsymbol{n}(\boldsymbol{\nu})}{\mathrm{d}\boldsymbol{\nu}}\right|_{\boldsymbol{\nu}=\boldsymbol{\nu}^*}}_{\boldsymbol{N}} (\boldsymbol{\nu} - \boldsymbol{\nu}^*) \\ &= \boldsymbol{n}(\boldsymbol{\nu}^*) + \boldsymbol{N}\boldsymbol{\nu} - \boldsymbol{N}\boldsymbol{\nu}^* \\ &= \boldsymbol{N}\boldsymbol{\nu} + \boldsymbol{b}(\boldsymbol{\nu}^*), \end{aligned} \tag{34}$$

where the current velocity $\boldsymbol{\nu}^*$ is selected as the linearization point and $\boldsymbol{b}(\boldsymbol{\nu}^*) = \boldsymbol{n}(\boldsymbol{\nu}^*) - \boldsymbol{N}\boldsymbol{\nu}^*$ is a constant term. The matrix $\boldsymbol{N}$ is stated in [2].

Inserting (34) into (31) yields:

$$\dot{\tilde{\boldsymbol{\nu}}} = \boldsymbol{A}\tilde{\boldsymbol{\nu}} + \boldsymbol{\beta}\boldsymbol{\nu}_{1d} + \boldsymbol{G}, \tag{35}$$

where:

$$\begin{aligned} \boldsymbol{A} &= -\left(\boldsymbol{\Gamma}_1^T \boldsymbol{K}_p \boldsymbol{\Gamma}_1 + \boldsymbol{\Gamma}_2^T \boldsymbol{\Gamma}_2 \boldsymbol{N}\right) \\ \boldsymbol{\beta} &= -\boldsymbol{\Gamma}_2^T \boldsymbol{\Gamma}_2 \boldsymbol{N} \boldsymbol{\Gamma}_1^T \\ \boldsymbol{G} &= -\boldsymbol{\Gamma}_2^T \boldsymbol{\Gamma}_2 \boldsymbol{b}(\boldsymbol{\nu}^*). \end{aligned} \tag{36}$$

This is a linear time-invariant system perturbed by a nonvanishing perturbation $\boldsymbol{G}$ (hence $\tilde{\boldsymbol{\nu}} = \boldsymbol{0}$ and $\boldsymbol{\nu}_{1d} \equiv \boldsymbol{0}$ does not imply $\dot{\tilde{\boldsymbol{\nu}}} = \boldsymbol{0}$). The time evolution of (35) is:

$$\begin{aligned} \tilde{\boldsymbol{\nu}}(t) = \mathrm{e}^{\boldsymbol{A}(t-t_0)}\tilde{\boldsymbol{\nu}}(t_0) \\ + \int_{t_0}^{t} \mathrm{e}^{\boldsymbol{A}(t-\sigma)} \left(\boldsymbol{\beta}\boldsymbol{\nu}_{1d}(\sigma) + \boldsymbol{G}\right) \mathrm{d}\sigma. \end{aligned} \tag{37}$$

The desired surge speed and yaw rate are considered constant for each trajectory, hence $\boldsymbol{\nu}_{1d}$ is constant for each trajectory. By letting $t_0 = 0$ s, (37) can be expressed as [20]:

$$\tilde{\boldsymbol{\nu}}(t) = \mathrm{e}^{\boldsymbol{A}t}\tilde{\boldsymbol{\nu}}(0) - \boldsymbol{A}^{-1}\left(\boldsymbol{I} - \mathrm{e}^{\boldsymbol{A}t}\right)\left(\boldsymbol{\beta}\boldsymbol{\nu}_{1d} + \boldsymbol{G}\right). \tag{38}$$

The kinematics (1a) are simulated in discrete time using the modified Euler method [21]:

$$\begin{aligned} \boldsymbol{\eta}(t_{n+1}) &= \boldsymbol{\eta}(t_n) + h\boldsymbol{k}_2 \\ \boldsymbol{k}_1 &= \boldsymbol{R}(\boldsymbol{\eta}(t_n))\boldsymbol{\nu}(t_n) \\ \boldsymbol{k}_2 &= \boldsymbol{R}(\boldsymbol{\eta}(t_n) + \frac{h}{2}\boldsymbol{k}_1)\boldsymbol{\nu}(t_n + \frac{h}{2}), \end{aligned} \tag{39}$$

where h is the integration time step, and $\boldsymbol{\nu}(t)$ is computed from (38) and (30). It should be noted that the absolute position is not required in the DW implementation, as the sonar measurements are given in $\{b\}$. Hence, the AUV prediction can be done in $\{b\}$ by selecting $\boldsymbol{\eta} = \begin{bmatrix} 0 & 0 & \psi \end{bmatrix}^T$.

IV. Simulation Results

A number of simulations have been conducted to compare the modified DW algorithm with the original DW algorithm, in order to evaluate the performance of the two algorithms. A 6 DOF nonlinear model of the HUGIN 1000 AUV with a horizontally oriented forward looking sonar, developed by the Norwegian Defence Research Establishment and implemented in SIMULINK, has been used for testing the algorithms. The HUGIN 1000 AUV model satisfies Assumptions 1-3 given in Section II.

TABLE I
Simulation parameters:

Parameter	Value	Description
k_u	$1\ \mathrm{s}^{-1}$	Surge controller gain
k_r	$1\ \mathrm{s}^{-1}$	Yaw rate controller gain
k_ψ	$0.2\ \mathrm{s}^{-1}$	Yaw controller gain
Δ	8 m	LOS lookahead distance
α	1	DW yaw rate scaling constant
β	$9\ \mathrm{s}^{-1}$	DW distance scaling constant
γ	3	DW surge speed scaling constant
u_d'	$2\ \mathrm{m\,s}^{-1}$	Desired surge speed
ΔT_{DW}	1 s	Dynamic window algorithm sampling time
$\bar{r}$	6 m	Size of the avoidance region Ω
r^*	3.5 m	Size of the antitarget region $\mathcal{T}$

The control system was implemented in MATLAB, with parameters as in Table I. The model parameters are not stated due to confidentiality reasons. Further details about the simulator are given in [2].

To illustrate the improvement of predicting the AUV trajectories using the proposed linear approximation compared to the original approach, a set of AUV trajectories are predicted using a search space consisting of three desired surge velocities and three desired yaw rates. Assuming that V_s does not impose any limitations on the search space, this results in nine velocity pairs. The initial velocity is chosen as $\boldsymbol{\nu}(0) = \begin{bmatrix} 2 & 0 & 0 \end{bmatrix}^T$ and the trajectories are predicted for 30 s. Figure 5 shows the actual AUV trajectories together with predicted trajectories using both the original prediction and the new linear approximation, for three of the velocity pairs (the other six trajectories look similar, see [2]). Table II

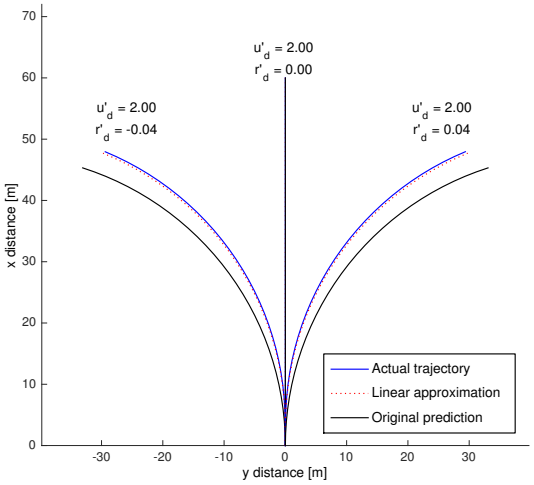

Fig. 5. Actual and approximated AUV trajectories, given initial velocity $\boldsymbol{\nu}(0) = [2 \quad 0 \quad 0]^T$ and $u_d' = 2\ \mathrm{m\,s}^{-1}$.

shows the average prediction error using both the original prediction and the new linear approximation. From Table II and Figure 5 it is clear that the linear approximation is much

TABLE II

MEAN SQUARE ERROR OF PREDICTED AUV TRAJECTORIES:

Time span	Original prediction	Linear approximation	Linear approximation vs. original prediction
$t = [0, 5]$ s	0.100 m^2	0.000576 m^2	0.576 %
$t = [0, 30]$ s	4.67 m^2	0.045 m^2	0.964 %

more accurate than the original prediction method, especially for the initial part of the trajectories.

Simulations in various environments all indicates that the modified algorithm achieves a more consistent and secure clearance to obstacles. One of the comparisons is shown in Figure 6. Following the straight line path between the three waypoints would result in a collision, and an alternative path is therefore found by the DW algorithm. In this case, the modified algorithm chooses a shorter route than the original algorithm. The modified algorithm achieves a larger obstacle clearance, and makes the AUV stay well clear of the antitarget region at all times. In contrast, the original algorithm makes the AUV enter the antitarget region during the simulation. See Figure 7 and Table III for details.

The robustness of the COLAV system is assessed through a Monte Carlo simulation with 1500 samples. The AUV COLAV system is simulated in environments generated by filtering and thresholding matrices of normal distributed random elements, to represent the environments as obstacle grids. Further, based on simulated sonar ranges, estimates of $\mathcal{T}$ and Ω are generated as shown in Figure 4. As shown in Table IV, 17.2 % of the simulations of the modified algorithm came closer than 3 m to an obstacle, and hence may have caused a collision (recall that the AUV is represented as a particle, and the radius of the AUV is approximately

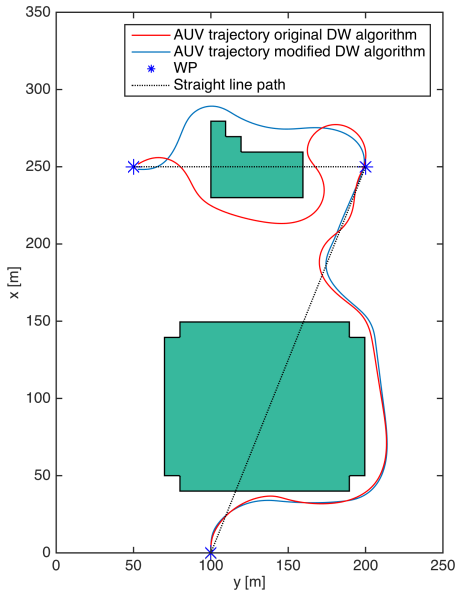

Fig. 6. AUV trajectories using the modified and original DW algorithms. The AUV starts at $(0, 100)$ m.

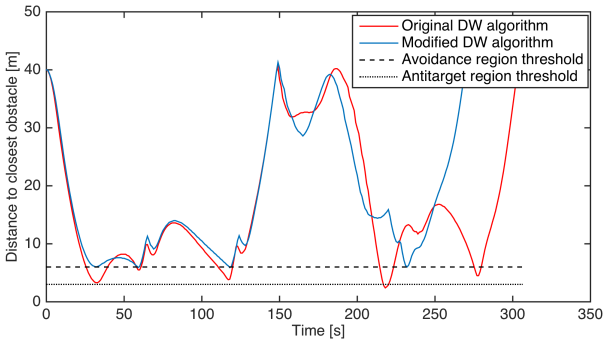

Fig. 7. Distance to closest obstacle.

3 m). Some trajectories even resulted in a minimum distance of 0.1 m, surely causing a collision. A closer inspection of the trajectories reveals that when the AUV reduces the surge speed to avoid collisions in local minima, the speed scaling of the damping causes the AUV to slide sideways for a long time after the surge speed reaches zero. This is considered to be a simulation artifact, since the model (1) does not capture the correct vehicle dynamics at low surge speeds. For further elaboration of the simulation artifact, see [2]. Notice, however, that 71.3 % of the simulations of the original algorithm came closer than 3 m, demonstrating a large improvement with regards to the original algorithm. From Table IV and Figure 8 it is clear that the modified DW algorithm consistently achieves a larger obstacle clearance. An interesting result is that the modified DW algorithm made the AUV reach the final waypoint only in 31.9 % of the simulations, hence the AUV got trapped in local minima in 68.1 % of the simulations. This demonstrates the need for adapting the global path underway for example by using a deliberate planner together with the DW algorithm in a hybrid architecture, to avoid local minima. However, the modified DW algorithm performed better than the original DW algorithm which only reached the final WP in 28 % of the simulations.

V. CONCLUSION

We have in this paper proposed a number of modifications to the dynamic window (DW) algorithm to make it suitable for vehicles with second-order nonholonomic constraints and time-varying acceleration limitations.

Based on simulations, a new AUV trajectory prediction method accounting for second-order nonholonomic con-

TABLE III

TRAJECTORY DATA, ORIGINAL AND MODIFIED DW ALGORITHM:

Parameter	Original algorithm	Modified algorithm
Trajectory length to end WP	602 m	539 m
Trajectory time to end WP	307 s	273 s
Average surge speed	1.94 m s^{-1}	1.95 m s^{-1}
Minimum obstacle clearance	2.4 m	5.9 m

TABLE IV

SUMMARY OF MONTE CARLO SIMULATION:

Min. obs. clearance	Original algorithm		Modified algorithm	
	Perc. of simulations	Perc. of which reached goal	Perc. of simulations	Perc. of which reached goal
$[0, 1]$ m	33.9 %	0.8 %	1.8 %	0 %
$(1, 2]$ m	4.1 %	17.7 %	5.1 %	0 %
$(2, 3]$ m	33.3 %	20.0 %	10.3 %	0 %
$(3, 4]$ m	26.5 %	68.8 %	38.3 %	8.7 %
$(4, 5]$ m	0.5 %	87.5 %	5.3 %	52.5 %
$(5, 6]$ m	0.6 %	100.0 %	31.0 %	62.2 %
$(6, \infty)$ m	1.0 %	100.0 %	8.2 %	79.7 %
All	100.0 %	28.0 %	100.0 %	31.9 %

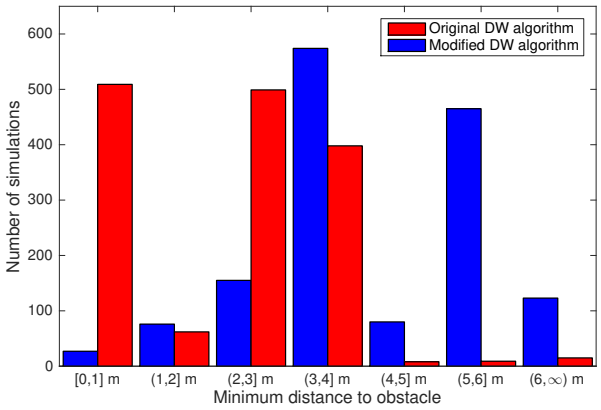

Fig. 8. Minimum obstacle clearance in the Monte Carlo simulation.

straints reduces the mean square prediction error to about one percent of the original method. Together with a modified search space, this improves the performance of the DW algorithm in terms of obstacle clearance and accuracy when applied to AUVs. Based on a Monte Carlo simulation of 1500 samples, the modified DW algorithm is believed to be robust with respect to obstacle configurations. This should however be investigated further. Due to the new prediction method, the computational cost of the modified algorithm is moderately larger than that of the original algorithm. On the other hand, the increased prediction accuracy makes it possible to run the DW algorithm at a lower sampling frequency.

In order to develop a practical COLAV system, further research will be put into combining the reactive DW algorithm with deliberate planning algorithms to ensure global convergence. The suitability of the DW algorithm for use with ASVs will also be evaluated.

ACKNOWLEDGMENT

This work was supported by the Research Council of Norway through project number 244116, and through the Centres of Excellence funding scheme with project number 223254.

REFERENCES

[1] C. S. Tan, R. Sutton, and J. Chudley, "Collision avoidance systems for autonomous underwater vehicles part A: A review of obstacle detection," *Journal of Marine Science and Environment*, vol. C2, pp. 39–50, 2004.

[2] B.-O. H. Eriksen, "Horizontal collision avoidance for autonomous underwater vehicles," Master's thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2015. [Online]. Available: http://hdl.handle.net/11250/2352502

[3] C. S. Tan, R. Sutton, and J. Chudley, "Collision avoidance systems for autonomous underwater vehicles part B: A review of obstacle avoidance," *Journal of Marine Science and Environment*, vol. C2, pp. 51–62, 2004.

[4] Ø. A. G. Loe, "Collision avoidance for unmanned surface vehicles," Master's thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2008. [Online]. Available: http://www.diva-portal.org/smash/record.jsf?pid=diva2:347606

[5] 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.

[6] Kongsberg Maritime. (2014) Autonomous underwater vehicle - HUGIN. Accessed: 2014-10-06. [Online]. Available: http://www.km.kongsberg.com/ks/web/nokbg0240.nsf/AllWeb/B3F87A63D8E419E5C1256A68004E946C?OpenDocument

[7] W. Caharija, K. Y. Pettersen, J. T. Gravdahl, and E. Børhaug, "Integral LOS guidance for horizontal path following of underactuated autonomous underwater vehicles in the presence of vertical ocean currents," in *Proc. of American Control Conference*, Montréal, Canada, 2012, pp. 5427–5434.

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

[9] SNAME, "Nomenclature for treating the motion of a submerged body through a fluid," The Society of Naval Architects and Marine Engineers, New York, USA, Tech. Rep., 1950.

[10] O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," in *Proc. of IEEE International Conference on Robotics and Automation*, Detroit, Michigan, 1999.

[11] P. Ögren and N. E. Leonard, "A tractable convergent dynamic window approach to obstacle avoidance," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, 2002, pp. 595–600.

[12] I. Tusseyeva, S.-G. Kim, and Y.-G. Kim, "3D global dynamic window approach for navigation of autonomous underwater vehicles," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 13, pp. 91–99, 2013.

[13] M. Seder, K. Macek, and I. Petrovic, "An integrated approach to real-time mobile robot control in partially known indoor environments," in *Proc. of the 31st Annual Conference of IEEE Industrial Electronics Society*, Raleigh, North Carolina, USA, 2005, pp. 1785–1790.

[14] A. Stentz, "The focussed D* algorithm for real-time replanning," in *Proc. of the International Joint Conference on Artificial Intelligence*, Pittsburgh, Pennsylvania, USA, 1995.

[15] C. Schröter, M. Höchemer, and H.-M. Gross, "A particle filter for the dynamic window approach to mobile robot control," in *Proc. of the 52nd International Scientific Colloquium*, Ilmenau, Germany, 2007, pp. 425–430.

[16] E. J. Rodríguez-Seda, C. Tang, M. W. Spong, and D. M. Stipanovic, "Trajectory tracking with collision avoidance for nonholonomic vehicles with acceleration constraints and limited sensing," *The International Journal of Robotics Research*, vol. 33, pp. 1569–1592, 2014.

[17] H. Berti, A. Sappa, and O. Agamennoni, "Improved dynamic window approach by using Lyapunov stability criteria," *Latin American Applied Research*, vol. 38, no. 4, pp. 289–298, 2008.

[18] P. Inñigo-Blasco, F. Díaz-del Río, S. Vicente Díaz, and D. Cagigas Muñiz, "The shared control dynamic window approach for non-holonomic semi-autonomous robots," in *Proc. of the 45th International Symposium on Robotics*, Munich, Germany, 2014, pp. 355–360.

[19] D. Kiss and G. Tevesz, "Advanced dynamic window based navigation approach using model predictive control," in *Proc. of the 17th International Conference on Methods and Models in Automation and Robotics*, Miedzyzdroje, Poland, 2012, pp. 148–153.

[20] J. P. Hespanha, *Linear Systems Theory*. Princeton University Press, 2009.

[21] O. Egeland and J. T. Gravdahl, *Modeling and Simulation for Automatic Control*. Marine Cybernetics, 2003.