Robotics Research and Technology

Underwater Vehicles

Design and Applications

George M. Roman
Editor

NOVA

ROBOTICS RESEARCH AND TECHNOLOGY

UNDERWATER VEHICLES

DESIGN AND APPLICATIONS

ROBOTICS RESEARCH AND TECHNOLOGY

Additional books and e-books in this series can be found on Nova's website under the Series tab.

ROBOTICS RESEARCH AND TECHNOLOGY

UNDERWATER VEHICLES

DESIGN AND APPLICATIONS

GEORGE M. ROMAN
EDITOR

nova
science publishers
New York

Library of Congress Cataloging-in-Publication Data

ISBN: 978-1-53618-967-4(eBook)

Published by Nova Science Publishers, Inc. † New York

CONTENTS

Preface

Underwater Vehicles: Design and Applications first explores the application of the adaptive Kalman filter algorithm to the estimation of high speed autonomous underwater vehicle dynamics.

The authors investigate the performances of different control schemes, from non-model-based to model-based and adaptive model-based, implemented on a low-inertia underwater vehicle for three-dimensional helical trajectory tracking.

Control laws for collision avoidance in three-dimensional environments are introduced, considering scenarios where a vehicle detects arbitrarily shaped and nonconvex obstacles using sensors.

Fading Kalman Filter (AFKF) with correction of process noise covariance (Q-Adaptation) for estimation of AUV Dynamics. The proposed approach in Chapter 1 is based on the adaptation scheme for the conventional KF algorithm, change in the noise covariance is detected and corrected via single or multiple fading factors that are introduced. The proposed AFKF algorithms give accurate estimation results despite the system uncertainty. The presented AFKF algorithms are simple for practical implementation and their computational burden is not heavy. These characteristics make introduced AFKF algorithms extremely important in terms of supplying reliable parameter estimation for the control system of the high speed AUV. Keeping in mind the harsh environments where AUVs are generally used, it is highly probable to come across/obtain a fault in the system inputs/parameters and so preferring the proposed AFKF algorithms instead of the conventional KF may bring a significant advantage.

Chapter 2 investigates the performances of different control schemes, from non-model-based (proportional-integral-derivative control, PID) to model-based (computed torque control, CT) as well as adaptive model-based (adaptive proportional-derivative plus control, APD+), implemented on a low-inertia underwater vehicle for three-dimensional (3D) helical trajectory tracking. Then, the asymptotic stability of the resulting closed-loop dynamics for each control scheme is proven based on the Lyapunov direct method. The performances of the control schemes, implemented on the Leonard underwater vehicle for 3D helical trajectory tracking, are then demonstrated through scenarios-based numerical simulations. The proposed simulations are conducted under the influences of the vehicle's buoyancy and damping changes, parametric variations; sensor noise, internal vehicle's perturbations; and water current, external disturbances rejection. Moreover, the authors demonstrate the task of transporting an object by the vehicle during underwater missions. The obtained simulation results show the effectiveness and robustness of the APD+ control scheme for tracking control of the low-inertia underwater vehicle in marine applications, outperforming the other controllers.

Collision avoidance in 3D environments is important to the problem of planning safe trajectories for an autonomous vehicle. Existing literature on collision avoidance assumed that obstacle shapes are known a priori and modeled obstacles as spheres or bounding boxes. However, in 3D environments, an obstacle shape is unknown to the autonomous vehicle, and the vehicle detects an obstacle boundary using 3D sensors, such as 3D sonar. In Chapter 3, the authors introduce control laws for collision avoidance, considering scenarios where a vehicle detects arbitrarily shaped and non-convex obstacles using sensors. Moreover, the control laws are designed considering motion constraints, such as the maximum turn rate and the maximum speed rate of the vehicle. The effectiveness of our control laws is verified using MATLAB simulations.

In: Underwater Vehicles
Editor: George M. Roman
ISBN: 978-1-53618-876-9

Chapter 1

ADAPTİVE ADJUSTMENT OF PROCESS NOİSE COVARİANCE İN KALMAN FİLTER FOR ESTİMATİON OF AUV DYNAMİCS

Chingiz Hajiyev[1,*], Sıtkı Yenal Vural[1] and Ulviye Hacizade[2]

[1]Aeronautics and Astronautics Faculty,
Istanbul Technical University, Istanbul, Turkey
[2]Faculty of Engineering, Halic University, Istanbul, Turkey

ABSTRACT

Fading Kalman Filter (AFKF) with correction of process noise covariance (Q-Adaptation) for estimation of AUV Dynamics. The proposed approach is based on the adaptation scheme for the conventional KF algorithm, change in the noise covariance is detected and corrected via single or multiple fading factors that are introduced. The proposed AFKF algorithms give accurate estimation results despite the system uncertainty. The presented AFKF algorithms are simple for practical implementation and their computational burden is not heavy. These characteristics make introduced AFKF algorithms extremely important in terms of supplying reliable parameter estimation for the control system of the high speed AUV. Keeping in mind the harsh environments where AUVs are generally used,

[*] Corresponding Author's Email: cingiz@itu.edu.tr.

it is highly probable to come across/obtain a fault in the system inputs/parameters and so preferring the proposed AFKF algorithms instead of the conventional KF may bring a significant advantage.

Keywords: autonomous underwater vehicle, adaptive Kalman filter, state estimation, fading factor, system uncertainty

1. Introduction

The research on underwater systems has gained an immense interest during the last decades with novel applications taking place in many fields. Therefore, significant number of autonomous underwater vehicles (AUVs) have been developed for solving problems involving a wide spectrum of scientific and applied tasks of sea research and development in the world.

AUVs require a precise navigation system for localization, positioning, path tracking, guidance, and control during a long period of duty cycle [1]. In order to develop an accurate and robust navigation and control system for an AUV, it is necessary to derive an adaptive algorithm for estimation of AUV dynamics.

Since it was proposed, Kalman filter has been widely used as an AUV motion dynamics parameters estimation technique and different Kalman filter types have been developed with that purpose [2, 3]. As a known fact; motion dynamics parameters estimation of an AUV cannot be solved by linear Kalman filters because of the inherent nonlinear dynamics and kinematics. In such a case Extended Kalman Filter (EKF) may be used instead. By using EKF, it is possible to estimate motion dynamics parameters of an AUV, which has a typical navigation sensor outfit such as compass, pressure depth sensor, and some class of inertial navigation system (INS) [4].

In the normal operation conditions of AUV, conventional Kalman filter gives sufficiently good estimation results. However, if the measurements are not reliable because of any kind of malfunction in the estimation system, KF gives inaccurate results and diverges by time. The conventional KF has no capability to adapt itself to the changing conditions of the measurement system. Malfunctions such as abnormal measurements, increases in the background noise and the like affect instantaneous filter outputs and as a result the process may end up with the failure of the filter. In order to avoid from such a condition, the filter must be operated robustly.

KF can be made adaptive and hence insensitive to the prior measurements or system uncertainties by using various different techniques. Multiple Model Based Adaptive Estimation (MMAE), Innovation Based Adaptive Estimation (IAE) and Residual Based Adaptive Estimation (RAE) are three of the basic approaches to the adaptive Kalman filtering. In the first approach, various filters run parallel under different models for satisfying filter's true statistical information. However, that can only be achieved if the sensor/actuator faults are known. Also, this approach requires several parallel Kalman filters to run and the processing time may increase in this case [5]. In IAE or RAE methods, adaptation is applied directly to the covariance matrices of the measurement and/or system noises in accordance with the differentiation of the residual or innovation sequence. To realize these methods, the innovation or residual vectors must be known for m epoch which causes an increment in the storage burden, as well as creating a requirement to know the width of the moving window [6]. Besides, in order to estimate the covariance matrix of the measurement noise based on the innovation or residual vector, the number, type and distribution of measurements must be consistent for all epochs within a window.

Kalman filter may also be made adaptive by using fuzzy logic based techniques. When the theoretical and real innovation values of covariance matrices of the measurement or process noises are compared, and a variable which characterizes the discrepancy level between them is defined, then by the fuzzy logic rules, process or measurement covariance matrices can be adjusted [7, 8]. However, the essence of these kinds of fuzzy methods are human experience and heuristic information; in cases where such experience is lacking, they may not work.

Another concept is to scale the noise covariance matrix by multiplying it with a time dependent variable. One of the methods for constructing such an algorithm is to use a single adaptive factor as a multiplier to the process or measurement noise covariance matrices [5, 9]. This algorithm, which may be named as Adaptive Fading Kalman Filter (AFKF), can be used when the information about the dynamic process or the priori measurements is absent

[10]. However, when the point at issue is the recent measurements, another technique to scale measurement noise covariance matrix and to make filter robust (insensitive to recent measurement faults) should be proposed. Therefore, if there is a malfunction in the measurement system, Robust Kalman Filter (RKF) algorithm with correction of measurement noise covariance (R-adaptation) can be utilized. In this case, by using a measurement noise scale factor (MNSF) as a multiplier on the measurement noise covariance matrix, insensitiveness of the filter to the current measurement faults can be satisfied. Consequently, via a correction applied to the filter gain, good estimation behaviour of the filter will be secured without being affected from faulty current measurements [6, 11].

One important problem that needs to be addressed in using KF is how to properly set up the covariance matrices of process noise (i.e., Q) and measurement noise (i.e., R). Note that the performance of the KF is highly affected by Q and R [12]. Improper choice of Q and R may significantly degrade the performance of KF and even make the filter diverge.

The paper [13] proposes an estimation approach to adaptively adjust Q and R at each step of the EKF to improve the dynamic state estimation accuracy. An innovation-based method is used to adaptively adjust Q. A residual-based method is used to adaptively adjust the R.

In Ref. [14] the problem of unknown system noises and uncertain measurement noises inherent in underwater cooperative navigation is solved via a variational Bayesian-based Adaptive Extended Kalman Filter. The Inverse Wishart distribution is used to model the predicted error covariance and measurement noise covariance matrix. The state, together with the predicted error covariance and measurement noise covariance, are adaptively estimated based on variational Bayesian approximation.

In [15], the Unscented Kalman Filter (UKF) algorithm that is used for estimating the dynamics of an AUV is adapted against changes in the environment and provides accurate estimation results even in such cases. The main aim is to make the algorithm adaptive against the changes in the process noise covariance (Q-adaptation). The Adaptive UKF estimates the AUV's dynamics. The numerical results confirm that the adaptive UKF gives better results than the regular UKF in cases where there are changes in the environment. The results of this study show that it is not possible to get precise estimation results by the regular UKF if the model for the process noise covariance disagrees with the real value as a result of changes occurring in the environment.

An adaptive extended Kalman filter (AEKF) is proposed in [16] by estimating the covariance matrix of prediction error and covariance matrix of measurement noise adaptively based on an online expectation-maximization approach. The presented AEKF does not require a window of data because the predicted error covariance matrix is estimated instead of the process noise covariance matrix, which makes it suitable for the case of unknown and time-varying noise covariance matrices.

In this study, AFKF algorithms with correction of process noise covariance (Q-adaptation) are introduced and applied for the motion dynamics parameters estimation process of an AUV. Two types of Q-adaptation procedures are presented; with single fading factor (SFF) and with multiple fading factor (MFF). The proposed AFKFs are considerably simpler than the existing and may be preferred, especially for the AUV motion dynamics estimation. Throughout the study, results of these proposed algorithms are compared using different types of system malfunctions (actuator faults) and the recommendations regarding their applications are discussed.

The paper proceeds as follows; AUV motion dynamics mathematical model is given in Section 2. The novel AFKF algorithms are presented in Section 3. The simulation of testing algorithms as part of an AUV motion dynamics parameters estimation procedure is presented in Section 4. Section 5 gives a brief summary of the obtained results and the conclusions.

2. Mathematical Model of Autonomous Underwater Vehicle Dynamics

AUV modeling is fairly complicated, and an exact analysis is only possible by including the underlying infinite dimensional dynamics of the surrounding fluid (sea water). While this can be done using partial differential equations in Computational Fluid Dynamics (CFD) computer tools, it still involves a formidable computational burden, infeasible for most practical applications.

AUVs move in 6 degrees of freedom (6DOF) since six independent coordinates are necessary to determine the position and orientation of a rigid body. The first three coordinates and their time derivatives are of translational motion along the x, y and z-axes, while the last three coordinates (ϕ, θ, ψ) and time derivatives are used to describe the orientation and rotational motion.

Instead of the sample AUV, the linearized model of REMUS torpedo will be used in calculations. Six different motion variables help to determine position and orientation. First three coordinates (x, y, z) are used to determine the position. Time derivatives of three coordinates (u, v, w) define transitions along x, y and z. Euler angles show the orientation. Time derivatives of Euler angles (p, q, r) express the rotational motion.

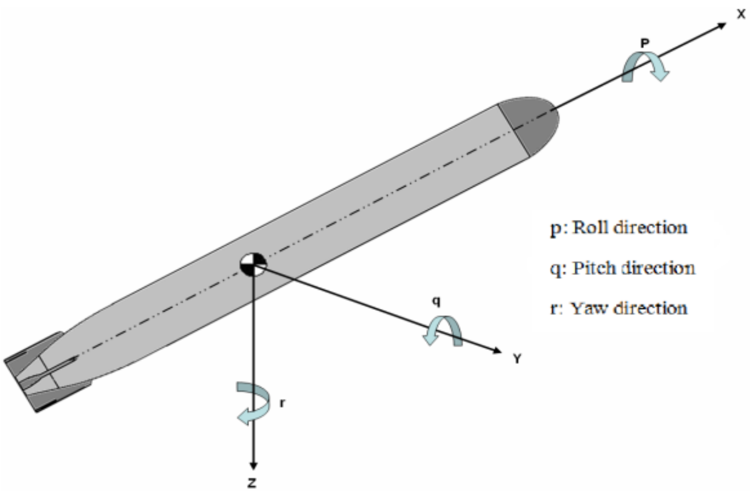

Figure 1. 6-DOF AUV Angular and Translational Motions.

2.1. Diving Subsystem of Sample AUV Model

Basically, diving subsystem includes heave velocity w, angular velocity q in pitch direction, pitch angle θ, depth z and bending of stern surface (deflection) δ_s. Diving subsystem neglects sway velocity v, roll rate of rotation r, heading angle ψ, rotation mode (p, φ) and initial horizontal movements of X and Y.

Vehicle is assumed to move with constant u_0 velocity with respect to water and zero pitch angle. Linearized equations of motions in direction of Heave and Pitch angles are given below [17];

$$\begin{bmatrix} m - Z_{\dot{w}} & mx_G - Z\dot{q} & 0 & 0 \\ mx_G - M_{\dot{w}} & I_y - M\dot{q} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{w} \\ \dot{q} \\ \dot{\theta} \\ \dot{z} \end{bmatrix} =$$

$$\begin{bmatrix} Z_w & Z_q - mu & 0 & 0 \\ M_w & M_q - mx_G u_0 & -\overline{BG_Z}W & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & -u_0 & 1 \end{bmatrix} \begin{bmatrix} w \\ q \\ \theta \\ z \end{bmatrix} + \begin{bmatrix} Z_\delta \\ M_\delta \\ 0 \\ 0 \end{bmatrix} \delta_s, \tag{1}$$

Equation (1) uses the hydrodynamic to which a linear decrease and deflection of stern surface to define external forces and moments is added. In addition, vertical distance between mass center z_G and buoyancy center z_B model the moment from $\overline{BG_z}$.

2.1.1. Discretization for Diving Subsystem

Diving subsystem matrices are given below;

$$M = \begin{bmatrix} m - Z_{\dot{w}} & mx_G - Z\dot{q} & 0 & 0 \\ mx_G - M_{\dot{w}} & I_y - M\dot{q} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

$$A_D = \begin{bmatrix} Z_w & Z_q - mu & 0 & 0 \\ M_w & M_q - mx_G u_0 & -\overline{BG_Z}W & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & -u_0 & 1 \end{bmatrix}, \quad B_D = \begin{bmatrix} Z_\delta \\ M_\delta \\ 0 \\ 0 \end{bmatrix}, \tag{3}$$

The mathematical model of the diving subsystem can be rewritten in the matrix form as;

$$\dot{X}_D(t) = M^{-1}A_D X_D(t) + M^{-1}B_D \delta_s(t), \tag{4}$$

where

$$X_D(t) = [w(t); q(t); \theta(t); z(t)] \tag{5}$$

is the state vector.

After discretization we obtain the diving subsystem model in the following form:

$$X_D(k+1) = A_D^* \times X_D(k) + B_D^* \times U_D(k), \tag{6}$$

where

$$A_D^* = I + \Delta t \times M^{-1} \times A_D ; B_D^* = \Delta t \times M^{-1} \times B_D , \tag{7}$$

$U_D(k)$ is the control input coming from deflection [18].

2.2. Steering Subsystem of Sample AUV

Steering subsystem equations are shown below;

$$\begin{bmatrix} m - Y_{\dot{V}_r} & -Y_{\dot{r}} & 0 \\ -N_{\dot{V}_r} & Izz - N_{\dot{r}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{v}_r \\ \dot{r} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} Y_{V_r} & Y_r - mU_0 & 0 \\ N_{v_r} & N_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ r \\ \psi \end{bmatrix} + \begin{bmatrix} Y_\delta \\ N_\delta \\ 0 \end{bmatrix} \delta_r(t), \tag{8}$$

$$M = \begin{bmatrix} m - Y_{\dot{V}_r} & -Y_{\dot{r}} & 0 \\ -N_{\dot{V}_r} & Izz - N_{\dot{r}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

If the inverse of M matrix in (9) is calculated and both sides are multiplied with M^{-1} in (8), equation transforms to;

$$\begin{bmatrix} \dot{v}_r \\ \dot{r} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} m - Y_{\dot{V}_r} & -Y_{\dot{r}} & 0 \\ -N_{\dot{V}_r} & Izz - N_{\dot{r}} & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} Y_{V_r} & Y_r - mU_0 & 0 \\ N_{v_r} & N_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ r \\ \psi \end{bmatrix} + \begin{bmatrix} m - Y_{\dot{V}_r} & -Y_{\dot{r}} & 0 \\ -N_{\dot{V}_r} & Izz - N_{\dot{r}} & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} Y_\delta \\ N_\delta \\ 0 \end{bmatrix} \delta_r(t), \tag{10}$$

2.2.1. Discretization of Steering Subsystem

A and B matrix are defined as below in equation (10):

$$A_S = \begin{bmatrix} m - Y_{\dot{V}_r} & -Y_{\dot{r}} & 0 \\ -N_{\dot{V}_r} & Izz - N_{\dot{r}} & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} Y_{V_r} & Y_r - mU_0 & 0 \\ N_{v_r} & N_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{11}$$

$$B_S = \begin{bmatrix} m - Y_{\dot{V}_r} & -Y_{\dot{r}} & 0 \\ -N_{\dot{V}_r} & Izz - N_{\dot{r}} & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} Y_\delta \\ N_\delta \\ 0 \end{bmatrix}, \tag{12}$$

If A_S and B_S matrices are defined as (11) and (12), A_S^* and B_S^* matrices are also defined for discretization as below;

$$A_S^* = I + \Delta t \times A_S ; B_S^* = \Delta t \times B_S , \tag{13}$$

Let us define the state vector as $X_S = \begin{bmatrix} v_r & r & \psi \end{bmatrix}^T$. Then the mathematical model of the steering subsystem can be written in the discrete form as:

$$X_S(k+1) = A_S^* \times X_S(k) + B_S^* \times U_S(k), \tag{14}$$

Here, $U_S(k)$ is control input by rudders. Discretized model (14) will be used for Kalman applications.

3. Kalman Filter for Estimation of AUV Dynamics

3.1. Optimum Linear Kalman Filter Equations

Consider the following linear discrete dynamic system:

$$X(k+1) = \Phi(k+1,k)X(k) + B(k)u(k) + G(k+1,k)W(k), \tag{15}$$

$$z(k) = H(k)X(k) + v(k), \tag{16}$$

where $X(k)$ is the m-dimensional state vector of the system at time t_k, $\Phi(k+1,k)$ is the $m \times m$ transition matrix of the system, $B(k)$ is the $m \times p$ control distribution matrix, $u(k)$ is the $p \times 1$ control vector; $W(k)$ is the r-dimensional random Gaussian noise vector (system noise) with zero mean and known covariance structure, $G(k+1,k)$ is the $m \times r$ transition matrix of the system noise, $z(k)$ is the s-dimensional measurement vector at time t_k, $H(k)$ is the $s \times m$ dimensional measurement matrix of the system, and $v(k)$ is the s-dimensional measurement noise vector with zero mean and known covariance structure.

There is no correlation between the system noise $W(k)$ and the measurement noise $v(k)$.

Apparently, the optimum Kalman filter (OKF) that estimates the state vector of the system (15) is expressed with the following recursive equations system [19]:

Equation of the estimation value,

$$\hat{X}\left(k/k\right) = \hat{X}(k/k-1) + K(k)\left[z(k) - H(k)\hat{X}(k/k-1)\right], \tag{17}$$

where;

$$\hat{X}(k/k-1) = \Phi(k,k-1)\hat{X}(k-1/k-1) + B\left(k-1\right)u\left(k-1\right) \tag{18}$$

is the extrapolation value, $K(k)$ is the gain matrix of the optimum linear Kalman filter:

$$K(k) = P(k/k-1)H^T(k)x\left[H(k)P(k/k-1)H^T(k) + R(k)\right]^{-1} \tag{19}$$

$R(k)$ is the covariance matrix of measurement noise.

The covariance matrix of the filtering error is,

$$P(k/k) = \left[I - K(k)H(k)\right]P(k/k-1), \tag{20}$$

where I is the identity matrix.

The covariance matrix of the extrapolation error is,

$$P(k/k-1) = \Phi(k,k-1)P(k-1/k-1)\Phi^T(k,k-1) + G(k,k-1)Q(k-1)G^T(k,k-1), \tag{21}$$

where $Q(k-1)$ is the covariance matrix of system noise.

3.2. Adaptive Fading Kalman Filter

In case of normal operation, where the model for the process noise covariance matches the real values, regular KF works without any divergence problem if it is tuned correctly for the issued problem. However, when a change occurs in the process noise covariance, the filter fails, and the estimation outputs become faulty.

Hence, an adaptive algorithm must be introduced. So, as an adaptation on process noise covariance, Q-adaptation is performed, and the estimations of the filter are corrected without affecting good estimation characteristic of the remaining process.

3.2.1. Adaptive Fading KF with Single Fading Factor

In case of system malfunctions (for example, an actuator fault in the system that results with changes in the control distribution matrix), an approach for Q adaptation can be followed. The covariance matrix of the innovation can be written as:

$$
\begin{aligned}
P_{\Delta}(k) &= H(k)P(k/k-1)H^{T}(k)+R(k) = H(k)\left[\Phi(k,k-1)P(k-1/k-1)\Phi^{T}(k,k-1)\right] \\
&\times H^{T}(k)+H(k)\left[\Lambda(k)G(k,k-1)Q(k-1)G^{T}(k,k-1)\right]H^{T}(k)+R(k)
\end{aligned}
\tag{22}
$$

in which adaptive factor (weight coefficient) $\Lambda(k)$ is calculated from the innovation sequence

$$\Delta(k) = z(k) - H(k)\hat{X}(k/k-1) \tag{23}$$

analysis results.

According to the proposed approach the gain matrix is changed when the following condition is valid [20]:

$$
\begin{aligned}
& tr\left\{\Delta(k)\Delta^T(k)\right\} \geq tr\left\{E\left[\Delta(k)\Delta^T(k)\right]\right\} \\
& = tr\left\{E\left[H(k)\left(X(k)-\hat{X}(k/k-1\right)+v(k)\right]\times\left[H(k)\left(X(k)-\hat{X}(k)\right)+v(k)\right]^T\right\} \\
& = tr\left\{H(k)P(k/k-1)H^T(k)+R(k)\right\}
\end{aligned} \tag{24}
$$

When a significant change in the conditions of operation of the measurement system occurs, the prediction of observations $H\left(k\right)\hat{X}\left(k/k-1\right)$ will considerably differ from the observation results $z(k)$. Consequently, the sum of the discrepancy squares on the left side of (24) will characterize the real filtration error, while the right side determines the theoretical accuracy of the innovation sequence, obtained on the basis of a priori information. If condition (24) is met, then the real filtration error exceeds the theoretical error. Therefore, it is necessary to correct the filter gain matrix beginning from this moment. In this case, by substituting (21) in (24) the following expression can be obtained;

$$
\begin{aligned}
& tr\left\{\Delta(k)\Delta^T(k)\right\} = \\
& tr\left\{H(k)\left[\Phi(k,k-1)P(k-1/k-1)\Phi^T(k,k-1)\right]H^T(k)\right\} \\
& +tr\left\{H(k)\left[\Lambda(k)G(k,k-1)Q(k-1)G^T(k,k-1)\right]H^T(k)\right\}+tr\left\{R(k)\right\}
\end{aligned} \tag{25}
$$

Hence taking the expression $tr\left\{\Delta(k)\Delta^T(k)\right\}=\Delta^T(k)\Delta(k)$ into consideration, the following formula for the fading (adaptive) factor $\Lambda(k)$ is obtained:

$$
\Lambda(k)=\frac{\Delta^T(k)\Delta(k)-tr\left\{H(k)\left[\Phi(k,k-1)P(k-1/k-1)\Phi^T(k,k-1)\right]H^T(k)\right\}-tr\left\{R(k)\right\}}{tr\left\{H(k)\left[G(k,k-1)Q(k-1)G^T(k,k-1)\right]H^T(k)\right\}} \tag{26}
$$

Apart from that point, if there is a fault in the system, $\Lambda(k)$ must be put into process as,

$$P(k/k-1) = \Phi(k,k-1)P(k-1/k-1)\Phi^T(k,k-1) + \Lambda(k)G(k,k-1)Q(k-1)G^T(k,k-1), \quad (27)$$

If the left side of the expression (24) is greater than the right side, the fading factor value $\Lambda(k)$ will increase. This corresponds to the beginning of adaptation of filter. Consequently, the covariance matrix of innovation sequence $P_\Delta(k)$ (22) increases, and the filter gain matrix $K(k)$ increases too, which will cause strengthening of the corrective influence of innovation sequence in the estimation algorithm and decrease the difference between the estimation value $\hat{X}(k/k)$ and the actual value $X(k)$. This will lead to the decrease of innovation sequence $\Delta(k)$ and adaptive factor $\Lambda(k)$, weakening the corrective influence of innovation sequence. The final expressions of the proposed adaptive (Q-adaptation) filtration algorithm with the single fading factor can be presented via the formulas (17), (18), (19), (20), (26) and (27).

In contrast to the standard optimal filtration algorithm, in which the filter gain $K(k)$ is changed by program, current measurements in the proposed algorithm have larger weight, since the coefficients of matrix $K(k)$ are corrected by the results of each observation. This algorithm is adapted to the system operation conditions by the approximation of theoretical covariance matrix $P_\Delta(k)$ to the real covariance matrix of innovation sequence, by applying the changing adaptive factor $\Lambda(k)$. The mentioned change is accomplished using the matrix $\Delta(k)\Delta^T(k)$, which characterizes the real filtration error. The presented adaptive KF will ensure guaranteed adaptation of the filter to the change of system operation conditions.

3.2.2. Adaptive Fading KF with Multiple Fading Factor

In this case, again, the real and theoretical values of the innovation covariance matrix must be compared. When there is a system fault, the real error will exceed the theoretical one. Hence, if a fading matrix Λ_k , built of fading factors, is added into the algorithm as given below [20],

$$\begin{aligned}&\frac{1}{\mu}\sum_{j=k-\mu+1}^{k}\tilde{\Delta}(k)\tilde{\Delta}^T(k)=H(k)\times\\&\left[\Phi(k,k-1)P(k-1/k-1)\Phi^T(k,k-1)+\Lambda(k)G(k,k-1)Q(k-1)G^T(k,k-1)\right]H^T(k)\\&+R(k)\end{aligned}\tag{28}$$

Then the fading matrix can be determined as,

$$\begin{aligned}\Lambda(k)=&\left[\frac{1}{\mu}\sum_{j=k-\mu+1}^{k}\Delta(k)\Delta^T(k)-H(k)\Phi(k,k-1)P(k-1/k-1)\Phi^T(k,k-1)H^T(k)-R(k)\right]\\&\times\left[H(k)G(k,k-1)Q(k-1)G^T(k,k-1)H^T(k)\right]^{-1}\end{aligned}\tag{29}$$

The gained fading matrix should be diagonalized since the Q matrix must be a diagonal, positive definite matrix.

$$\Lambda^* = diag\left(\lambda_1^*, \lambda_2^*, ..., \lambda_n^*\right), \tag{30}$$

where,

$$\lambda_i^* = \max\{1, \Lambda_{ii}\} \quad i = 1, n\,. \tag{31}$$

where, Λ_{ii} represents the i[th] diagonal element of the matrix Λ . Apart from that point, if there is a fault in the system, Λ_k^* must be put into process as,

$$\begin{aligned}P(k/k-1)=\Phi\left(k,k-1\right)P\left(k-1/k-1\right)\Phi^{\mathrm{T}}\left(k,k-1\right)+\\\Lambda^*(k)G\left(k,k-1\right)Q\left(k-1\right)G^{\mathrm{T}}\left(k,k-1\right)\end{aligned}\tag{32}$$

Remark that, due to the scale matrix or fading matrix the covariance of the estimation error of AFKF increases in comparison with OKF. Therefore, adaptive algorithm is operated only when there is a system fault and in all other cases procedure is run optimally with regular Kalman filter. Process is controlled by the use of statistical information. At that point, the following two hypotheses may be introduced: γ_0 - the system is normally operating; γ_1 - there is a malfunction in the system. To detect failures, a statistical function may be defined as:

$$\beta(k) = \sum_{j=k-m+1}^{k} \Delta^T(j)\left[HP(j/j-1)H^T + R\right]^{-1}\Delta(j) \tag{33}$$

where m is the width of the moving window [21].

This statistical function has χ^2 distribution with s degree of freedom where s is the dimension of innovation vector.

If the level of significance, α , is selected as,

$$P\left\{\chi^2 > \chi^2_{\alpha,s}\right\} = \alpha; \;\; 0 < \alpha < 1, \tag{34}$$

the threshold value, $\chi^2_{\alpha,s}$ can be found. Hence, when the hypothesis γ_1 is correct, the statistical value of $\beta(k)$ will be greater than the threshold value $\chi^2_{\alpha,s}$, i.e.,:

$$\begin{aligned} &\gamma_0 : \beta(k) \le \chi^2_{\alpha,s} \;\; \forall k \\ &\gamma_1 : \beta(k) > \chi^2_{\alpha,s} \;\; \exists k \end{aligned} \tag{35}$$

4. Simulation Results

The coefficients of REMUS torpedo will be used instead of sample AUV in calculations. The coefficients for steering subsystem of REMUS torpedo are given in [22]. Simulation results are presented in Figures 2-7. Figures are outputs of MATLAB program codes used for this purpose.

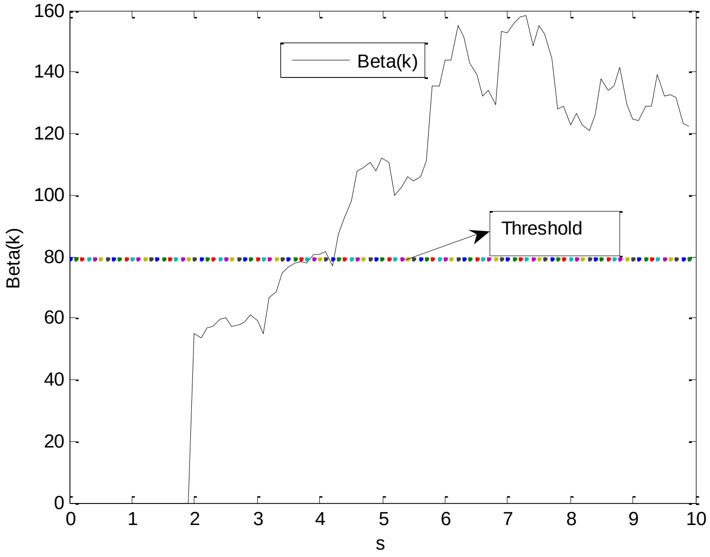

Figure 2. Actuator fault detection results when OKF is used.

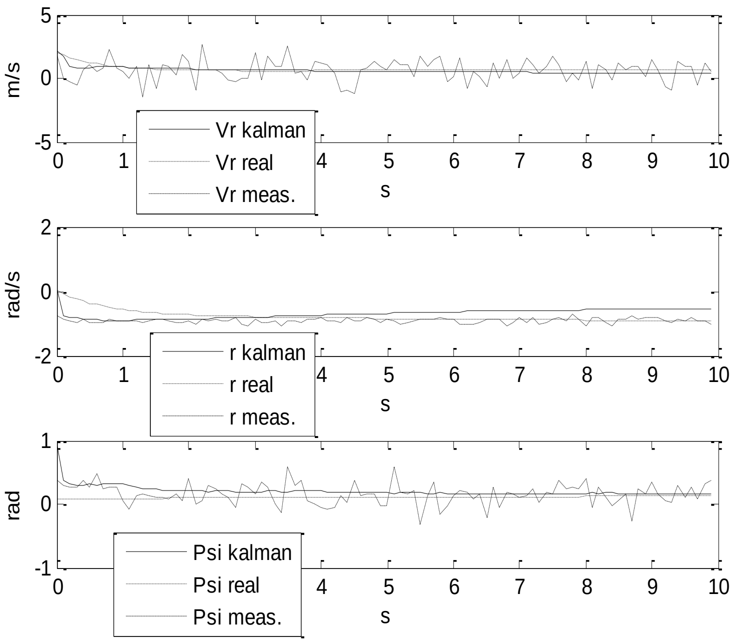

Figure 3. OKF estimation results in the case of actuator failure.

4.1. OKF Results

Steering subsystem model is used to simulate the actuator malfunction case. It is shown that with OKF the mentioned statistical tests can be used to detect the actuator faults in the system. The control distribution matrix of the system is changed to simulate the actuator fault case. The results when fault is present in actuator channel are given in the Figures 2-4.

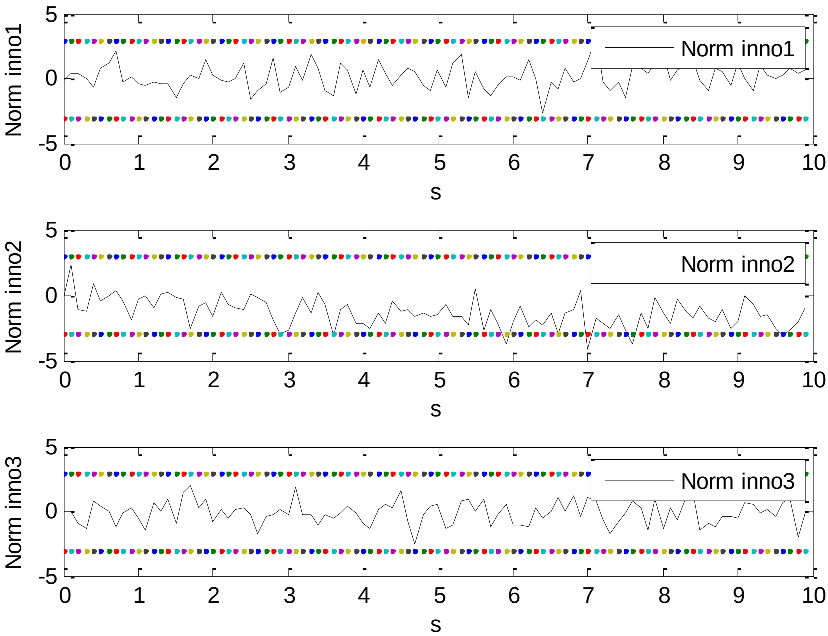

Figure 4. Normalized innovations of OKF in the case of actuator failure.

The statistical tests using the values determined in OKF can detect the actuator faults as we can observe. Figures 2-4 show that the OKF is affected from actuator fault and diverges by time.

4.2. AFKF with SFF Simulation Results

The control distribution matrix B is changed and a simulation is made using AFKF with SFF. As seen from Figures 5-7, AFKF with SFF is insensitive to actuator faults. It is observed that the adaptive filter gives better results compared to that of the OKF.

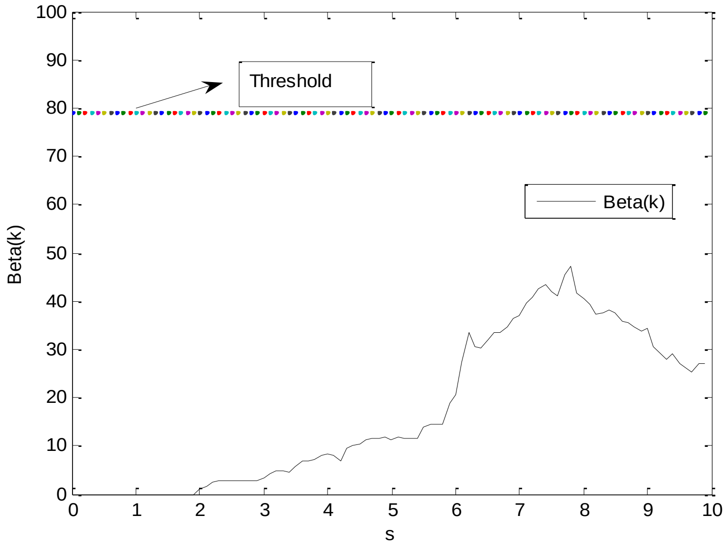

Figure 5. Actuator fault detection results when AFKF with SFF is used.

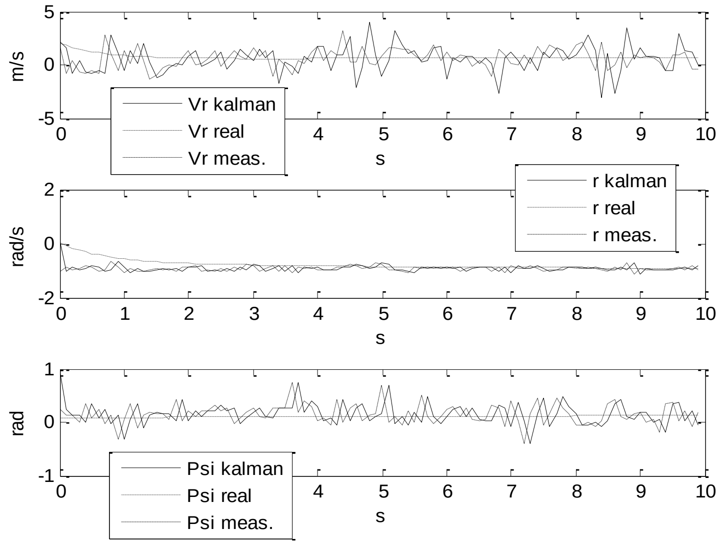

Figure 6. AFKF with SFF estimation results in the case of actuator failure.

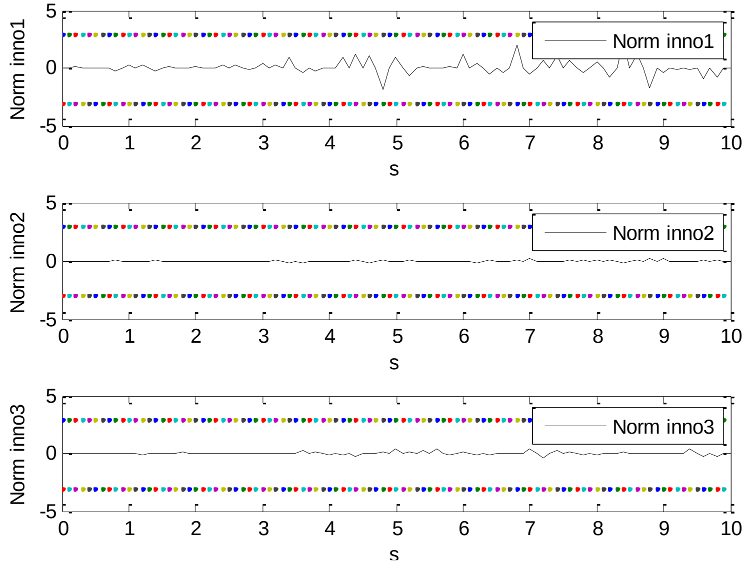

Figure 7. Normalized innovations of AFKF with SFF in the case of actuator failure.

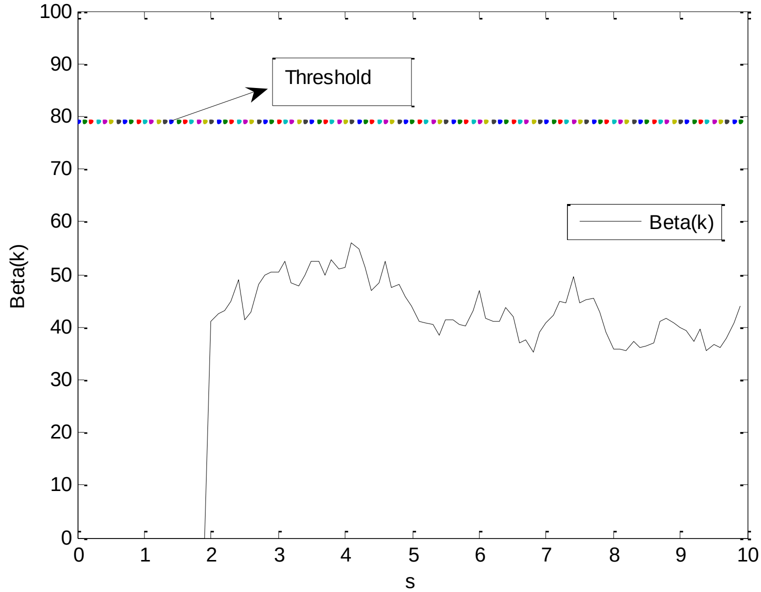

Figure 8. Actuator fault detection results when AFKF with MFF is used.

4.3. AFKF with MFF Simulation Results

AFKF with MFF is also simulated in case of actuator failure. As seen in Figure 8, AFKF with MFF is insensitive to actuator faults. Figure 9 shows that, the results of the adaptive filter with multiple factor are also better than that of the OKF.

As we can observe AFKF with MFF gives accurate results. The results of both the single and multiple fading factors filters are better than the results of the OKF. The adaptive filter estimation values converge to the real values and fault detection algorithm results stay under the threshold values. We can also conclude that the results obtained from AFKF with MFF are better than the AFKF with SFF results.

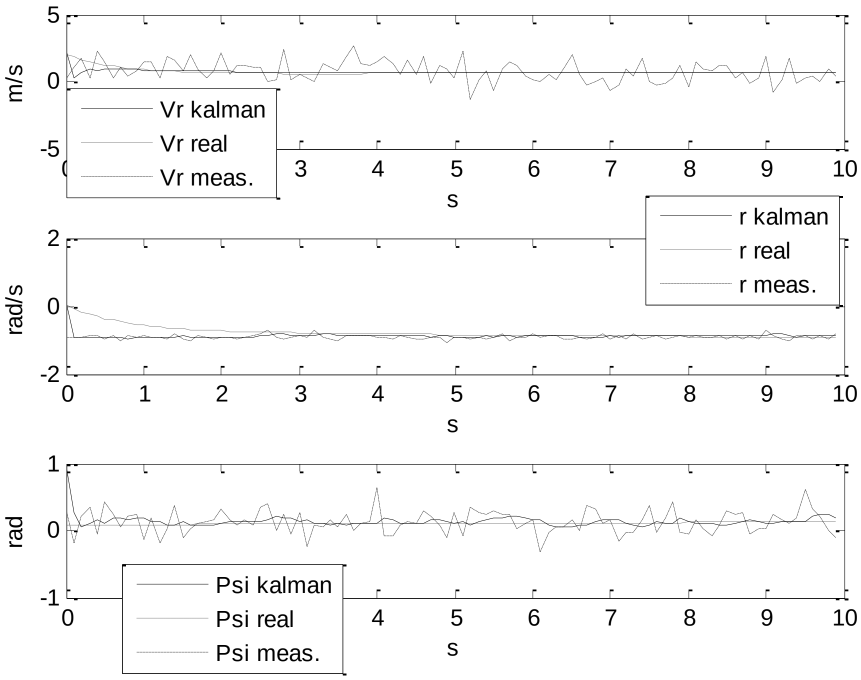

Figure 9. AFKF with MFF estimation results in the case of actuator failure.

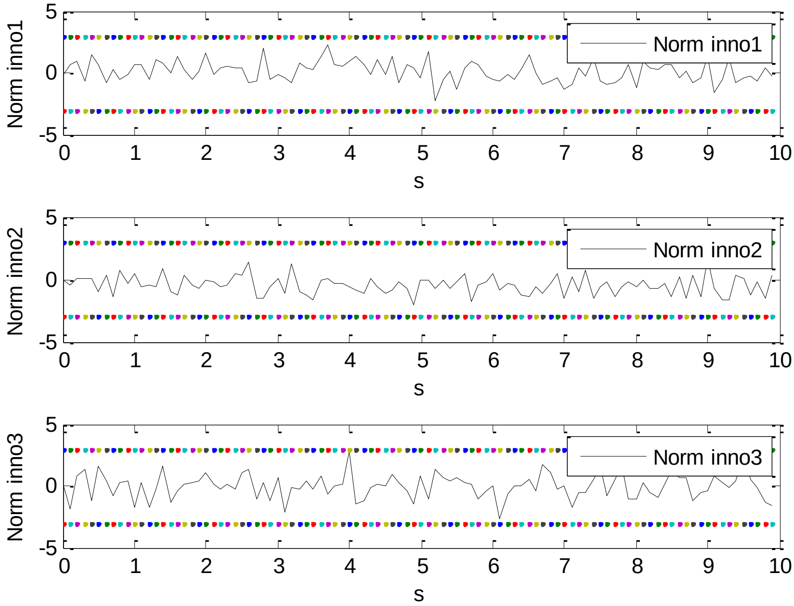

Figure 10. Normalized innovations of AFKF with MFF in the case of actuator failure.

CONCLUSION

This study is mainly focused on the application of the adaptive Kalman filter algorithm to the estimation of high speed autonomous underwater vehicle dynamics. In the normal operation conditions of AUV, conventional Kalman filter gives sufficiently good estimation results. However, if any kind of malfunction occurs in the system, KF gives inaccurate results and diverges with time. This study introduces adaptive fading Kalman filter algorithm with the filter gain correction in the case of system malfunctions. By using defined variables named as single and multiple fading factors (adaptive factors), the estimations are corrected without affecting the characteristic of the accurate ones. In this algorithm, by changing the adaptive factor, theoretical covariance matrix can be made adaptive to system operation conditions by converging to actual covariance matrix of innovation sequence.

Application of the proposed adaptive filters to AUV dynamics shows that it provides the adaptation to changes in operating conditions of system and provides correct results for both the regular and system failure conditions. Furthermore, the estimation system errors can be corrected without affecting the good estimation behavior. Simulation results show that the application of proposed algorithm to AUV fault tolerant steering and diving control system is beneficial.

The presented AFKF algorithms are simple for practical implementation and their computational burden is not heavy. This approach does not require knowledge of the a priori statistical characteristics of the faults and can be used for both linear and nonlinear systems.

The proposed AFKF algorithms may play an important role for the AUV control systems since it gives accurate estimation results despite the system faults. Given the harsh environments where the AUVs are generally used, it is highly possible/probable to encounter a fault, therefore preferring the proposed AFKF algorithms instead of the conventional KF may bring a significant advantage.

ACKNOWLEDGMENTS

This work was supported in part by TUBITAK (The Scientific and Technological Research Council of Turkey) under Grant 109M702.

REFERENCES

[1] Dinc, M. and Hajiyev, Ch. 2011. "Development of autonomous underwater vehicle navigation system based on Kalman filtering technique." *Proc. of the 18th St. Petersburg International Conference on Integrated Navigation Systems (ICINS- 2011)*, St. Petersburg, Russia, pp. 238-242.

[2] Lammas, A., Sammut, K. and He, F. 2010. "6-DOF navigation systems for autonomous underwater vehicles." In *Mobile robots navigation,* edited by A. Barrera, 457-483. InTech Publishing.

[3] Geng, Y., Martins, R. and Sousa, J. 2010. "Accuracy analysis of DVL/IMU/magnetometer integrated navigation system using different

IMUs in AUV." *Proc. of the 8th IEEE International Conference on Control and Automation*, pp. 516-521.

[4] Stutters, L., Liu, H., Tiltman, C. and Brown, D.J. 2008. "Navigation technologies for autonomous underwater vehicles." *IEEE Transactions on Systems, Man, and Cybernetics: Applications and Reviews*, 38(4), 581-589.

[5] Hide, C., Moore, T. and Smith, M. 2004. "Adaptive Kalman filtering algorithms for integrating GPS and low cost INS." *Proc. of the Position Location and Navigation Symposium*, Monterey, USA, pp. 227-233.

[6] Hajiyev, C. 2007. "Adaptive filtration algorithm with the filter gain correction applied to integrated INS/Radar altimeter." *Proceedings of the IMechE, Part G: Journal of Aerospace Engineering*, 221, 847-885.

[7] Escamilla-Ambrosio, P. J. and Mort, N. 2003. "Hybrid Kalman filter-fuzzy logic adaptive multisensor data fusion architecture." *Proceedings of. 42nd IEEE Conference on Decision and Control*, Hawaii, USA, Vol. 5, pp. 5215-5220.

[8] Jwo, D. and Cho, T. 2007. "A practical note on evaluating Kalman filter performance optimality and degradation." *Applied Mathematics and Computation*, 193, 482-505.

[9] Hu, C., Chen, W., Chen, Y. and Liu, D. 2003. "Adaptive Kalman filtering for vehicle navigation." *Journal of Global Positioning Systems*, 2, 42-47.

[10] Kim, K. H., Lee, J. G. and Park, C. G. 2006. "Adaptive two-stage Kalman filter in the presence of unknown random bias." *International Journal of Adaptive Control and Signal Processing*, 20, 305-319.

[11] Hajiyev, Ch. and Soken, H. E. 2012. "Robust estimation of UAV dynamics in the presence of measurement faults." *Journal of Aerospace Engineering*, 25(1), 80-89.

[12] Mohamed, A. H. and Schwarz, K. P. 1999. "Adaptive Kalman filtering for INS/GPS," *J. Geodesy*, vol. 73, no. 4, pp. 193-203, 1999.

[13] Akhlaghi, Sh., Zhou, N., Huang, Zh. 2017. "Adaptive adjustment of noise covariance in Kalman filter for dynamic state estimation." *2017 IEEE Power & Energy Society General Meeting*, 16-20 July 2017, Chicago, IL, USA.

[14] Sun, Ch., Zhang, Y., Wang, G. and Gao, W. 2018. “A New Variational Bayesian Adaptive Extended Kalman Filter for Cooperative Navigation.” *Sensors*, 18, 2538; doi:10.3390/s18082538.

[15] Hajiyev, C., Soken, H. E. (2018). Q-Adaptation of UKF Algorithm for Estimation of the Autonomous Underwater Vehicles Dynamics. Proc. of the 5th International Conference of Control, Dynamic Systems, and Robotics (CDSR’18, Niagara Falls, Canada – June 7-9, 2018.

[16] Huang, Y., Zhang, Y., Xu, B., Wu, Z., Chambers, J. A. 2018. A New Adaptive Extended Kalman Filter for Cooperative Localization. *IEEE Transactıons on Aerospace and Electronıc Systems,* Vol. 54, No. 1, pp. 353-368.

[17] Jalving, B. 1994. “The NDRE – AUV flight control system.” *IEEE Journal of Oceanic Engineering*, 19(4), 497-501.

[18] Lingli, N. 2001. “Fault-tolerant control of unmanned underwater vehicles.” PhD diss., Virginia Polytechnic Institute, Blacksburg, Virginia.

[19] Kalman, R. E. 1960. “A new Approach to linear Filtering and Prediciton Problems.” *Transactions of the ASME Journal of Basic Engineering*, 82 (Series D), 35-45.

[20] Hajiyev, Ch., Vural, S. Y. and Hajiyeva, U. 2012. “Adaptive fading Kalman filter with Q-adaptation for estimation of AUV dynamics.” *Preprints of the 20th Mediterranean Conference on Control & Automation (MED),* Barcelona, Spain, 2012, pp. 697-702.

[21] Haciyev, Ch., and Caliskan, F. 2003. *Fault diagnosis and reconfiguration in flight control systems*. Boston: Kluwer Academic Publishers.

[22] Fodrea, L. 2002. *Obstacle avoidance control for the REMUS autonomous underwater vehicle*. California: Storming Media.

BIOGRAPHICAL SKETCHES

Chingiz Hajiyev

Affiliation: Istanbul Technical University.

Education: Moscow Aviation Institute.

Business Address: Aeronautics and Astronautics Faculty, Istanbul Technical University, Maslak, 34469, Istanbul, Turkey.

Research and Professional Experience: In 1981 he graduated from the Faculty of Automatic Control Systems of Flying Apparatus, Moscow Aviation University (Moscow, Russia) with honour diploma. He received the PhD and DS. (Eng.) degrees in Process Control from Superior Certifying Commission at the Council of Ministers of the USSR from Azerbaijanian Scientific and Production Association (ASPA) "Neftgazavtomat" (Sumgait, Azerbaijan), in 1987 and 1993, respectively.

Professional Appointments: From 1987 to 1994 he worked as a Scientific Worker, Senior Scientific Worker, Chief of the Information-Measurement Systems Dept. at the ASPA "Neftgazavtomat." Since 1994 to 1996 he was a Leading - Scientific Worker at the Institute of Cybernetics of the Academy of Sciences of Azerbaijan Republic. He was also a Professor in the Department of Electronically-Calculated System Design, Azerbaijan Technical University, where he had been teaching 1995-1996.

Since 1996 he has been with Department of Aeronautics and Astronautics, Istanbul Technical University (Istanbul, Turkey), where he is currently a Professor. From December 2016 he is head of the Aeronautical Department of ITU.

Honors:

1981: Honors Certificate at Moscow Aviation Institute (Russia).

1994: Award from the International Science Foundation (USA).

2012: Best Paper Award, 13th International Carpathian Control Conference 2012, High Tatras, Slovak Republic.

2016: Sustainable Aviation Research Society (SARES) Science Award.

2016: Award for the Best Paper Presented of the 3rd IEEE International Workshop on Metrology for Aerospace, Florence, Italy, 2016.

2016: Recognized Reviewer Award - Ocean Engineering Journal, (UK).

2016: Recognized Reviewer Award - Aerospace Science and Technology Journal, (France).

2017: Outstanding Reviewer Award - Aerospace Science and Technology Journal, (France).

2019: Project Performance Award - The Scientific and Technological Research Council of Turkey (TUBITAK), (Turkey).

Publications from the Last Three Years:

[1] Contreras, A. M., Hajiyev C. (2018). Integration of Baro-Inertial-GPS Altimeter via Complementary Kalman Filter. In: Karakoç T., Colpan C., Şöhret Y. (eds) *Advances in Sustainable Aviation.* Springer, Cham, pp. 251-268. doi: https://doi.org/10.1007/978-3-319-67134-5_18.

[2] Hajiyev, C., Hacizade U. (2018). Testing the Determinant of the Innovation Covariance Matrix Applied to Aircraft Sensor and Actuator/Surface Fault Detection. In: Karakoç T., Colpan C., Şöhret Y. (eds) *Advances in Sustainable Aviation.* Springer, Cham, pp. 269-284. doi: https://doi.org/10.1007/978-3-319-67134-5_19.

[3] Cilden-Guler, D., Kaymaz, Z., Hajiyev, C. (2018). Evaluation of geomagnetic field models using magnetometer measurements for satellite attitude determination system at low earth orbits: Case studies. *Advances in Space Research*, 61(1), pp. 513-529.

[4] Cilden-Guler, D., Hajiyev, C., Dasdemir, J. (2018). Fault tolerant control of attitude dynamics of nanosatellites. *International Journal of Sustainable Aviation*, Vol. 4, Iss. 2, pp. 99-113.

[5] Hajiyev, Ch. and Sofyali, A. (2018). Spacecraft localization by indirect linear measurements from a single antenna. *Aircraft Engineering and Aerospace Technology*, Vol. 90, No. 5, pp. 734-742. doi: 10.1108/AEAT-12-2015-0245.

[6] Hajiyev, C. and Kamiloğlu, D. B. (2018). Design of loosely-coupled INS/GPS integration system applied to Boeing-747 aircraft model, *Int. J. Sustainable Aviation*, Vol. 4, Nos. 3/4, pp. 163-177.

[7] Cilden-Guler, D., Soken, H. E., Hajiyev, C. (2018). SVD-Aided UKF for Estimation of Nanosatellite Rotational Motion Parameters. *WSEAS Transactions on Signal Processing,* Vol. 14, pp. 27-35.

[8] Hajiyev, C., Cilden-Guler D. (2018). Gyroless Nanosatellite Attitude Estimation in Loss of Sun Sensor Measurements. *IFAC-PapersOnLine*, 51(30), 89-94.

[9] Cilden-Guler, D., Hajiyev C. (2018). UD Factorization Based Non-Traditional Attitude Estimation of Nanosatellite with Rate Gyros FDI. *IFAC-PapersOnLine*, 51(30), 95-100.

[10] Hajiyev C. (2018). An Innovation Approach Based Model Change Detection Applied to UAV Actuator/Surface FDI. *IFAC-PapersOnLine*, 51(30), 77-82.

[11] Hajiyev, C., Vural, S. Y., Shumsky A., Zhirabok, A. (2018). Robust Kalman Filter Based Estimation of AUV Dynamics. *IFAC-PapersOnLine*, 51(30), 424-429.

[12] Mammadov, H., Hajiyev C. (2018). Reconfigurable Fault Tolerant Flight Control for UAV with Pseudo-Inverse Technique. *IFAC-PapersOnLine*, 51(30), 83-88.

[13] Vural, S. Y., Janset Dasdemir J., Hajiyev C. (2018). Passive Fault Tolerant Lateral Controller Design for an UAV. *IFAC-PapersOnLine*, 51(30), 446-451.

[14] Somov, Ye., Hajiyev, Ch. (2018). Satellite Guidance and Control During Operative Optoelectronic Imagery for Disaster Management. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* Volume XLII-3/W4, 2018 GeoInformation For Disaster Management (Gi4DM), 18-21 March 2018, Istanbul, Turkey, pp. 475-482. https://doi.org/10.5194/isprs-archives-XLII-3-W4-475-2018.

[15] Hajiyev, Ch., Soken, H. E. (2018). Q-Adaptation of UKF Algorithm for Estimation of the Autonomous Underwater Vehicles Dynamics. *Proceedings of the 5th International Conference of Control, Dynamic Systems, and Robotics (CDSR'18),* Niagara Falls, Canada - June 7 - 9, 2018, Paper No. 103, doi: 10.11159/cdsr18.103.

[16] Hajiyev, C., Cilden-Guler, D., Hacizade, U. (2019). Two-Stage Kalman Filter for Estimation of Wind Speed and UAV States by using GPS, IMU and Air Data System. *WSEAS Transactions On Electronics*, Vol. 10, pp. 60-65.

[17] Hajiyev, C., Soken, H. E., Cilden-Guler, D. (2019). Nontraditional Attitude Filtering with Simultaneous Process and Measurement Covariance Adaptation. *Journal of Aerospace Engineering (ASCE)*, 32(5): 04019054.

[18] Cilden-Guler, D., Raitoharju, M., Piche, R., Hajiyev, C. (2019). Nanosatellite attitude estimation using Kalman-type filters with non-Gaussian noise. *Aerospace Science and Technology*, Vol. 92, pp. 66-76.

[19] Contreras A. M., Hajiyev C. (2019). Comparison of Conventional and Robust Adaptive Kalman Filters Based Integrated Altimeters. Proc. of the 20th International Carpathian Control conference (ICCC-2019), Krakow, Wieliczka, Poland, 26-29 May, 2019, *IEEE*, 6P.

[20] Contreras, A. M., Hajiev, C. (2019). Comparison of Linear and Nonlinear Kalman Filters Based Integrated Inertial, Baro and GPS Altimeters. Chapter in the book "*Advances in Engineering Research*," Editor V. M. Petrova, Vol. 31. Nova Science Publishers, Inc., NY, USA, pp. 117-155.

[21] Contreras, A. M., Hajiyev, C. (2019). Robust Kalman Filter Based Fault Tolerant Integrated Baro-Inertial-GPS Altimeter. *Metrology and Measurement Systems,* Vol. 26, No. 4., 673-686.

[22] Cilden-Guler, D., Kaymaz, Z., Hajiyev, C. (2019). Assessment of Magnetic Storm Effects under Various Magnetometer Noise Levels for Satellite Attitude Estimation. *Proc. 9th International Conference on Recent Advances in Space Technologies "Space for the Sustainable Development Goals" (RAST-2019),* 11-14 June 2019, Istanbul, Turkey, *IEEE*, pp. 769-773.

[23] Soken, H. E., Hacizade, C. (2019). Tuning the Attitude Filter: A Comparison of Intuitive and Adaptive Approaches. *Proc. 9th International Conference on Recent Advances in Space Technologies "Space for the Sustainable Development Goals" (RAST-2019),* 11-14 June 2019, Istanbul, Turkey, *IEEE*, pp.747-752.

[24] Bagci, M., C. Hajiyev, C. (2019). Measurement Conversion Based RKF for Satellite Localization via GPS. *Proc. 9th International Conference on Recent Advances in Space Technologies "Space for the Sustainable Development Goals" (RAST-2019),* 11-14 June 2019, Istanbul, Turkey, *IEEE*, pp. 861-868.

[25] Turan, E., Hajiyev, C. (2019). Performance Comparison of Guidance Algorithms for Planetary Landing, Asteroid Intercept and Rendezvous. *Proc. 9th International Conference on Recent Advances in Space Technologies "Space for the Sustainable Development Goals" (RAST-2019)*, 11-14 June 2019, Istanbul, Turkey, *IEEE*, pp. 919-925.

[26] Hajiyev, C., Cilden-Guler, D., Hacizade, U. (2019). EKF for Wind Speed Estimation and Sensor Fault Detection Using Pitot Tube Measurements. *Proc. 9th International Conference on Recent Advances in Space Technologies "Space for the Sustainable Development Goals" (RAST-2019)*, 11-14 June 2019, Istanbul, Turkey, *IEEE*, pp. 887-893.

[27] Hajiyev, C., Cilden-Guler, D., Hacizade, U. (2019). Two-Stage Kalman Filter for Estimation of Wind Speed and UAV Flight Parameters Based on GPS/INS and Pitot Tube Measurements. *Proc. 9th International Conference on Recent Advances in Space Technologies "Space for the Sustainable Development Goals" (RAST-2019)*, 11-14 June 2019, Istanbul, Turkey, *IEEE*, pp. 875-880.

[28] Haciyev, C., Hacizade, U., Cilden-Guler, U. (2019). Data Fusion for Integrated Baro/GPS Altimeter. *Proc. 9th International Conference on Recent Advances in Space Technologies "Space for the Sustainable Development Goals" (RAST-2019)*, 11-14 June 2019, Istanbul, Turkey, *IEEE*, pp. 881-885.

[29] Hajiyev, C., Vural, S. Y. (2019). Active Fault Tolerant Lateral Control Against Actuator Faults Applied to AUV Dynamics. *Proc. 4th Conference on Control and Fault Tolerant Systems (SysTol),* Casablanca, Morocco, September 18-20, 2019, pp. 153-158.

[30] Cilden-Guler, D., Hajiyev, C. (2019). SVD-aided EKF attitude estimation with UD factorized measurement noise covariance. *Asian Journal of Control*, Vol. 21, pp. 1423–1432.

[31] Dökme, Ö., Hajiyev C. (2019). *Assessment of Fault Tolerant Estimation for Light Utility Helicopter Dynamics. 8th Asian/Australian Rotorcraft Forum*, Ankara, Turkey, Oct. 30 - Nov. 2, 2019, 9P.

[32] Mammadov, H., and Hajiyev, C. (2019). Active fault tolerant flight control for UAV actuator failures. *Mathematıcs In Engıneerıng, Scıence And Aerospace*, Vol. 10, No. 4, pp. 657-670.

[33] Hajiyev, C., Vural, S. Y. (2020). "Reconfigurable Fault Tolerant Control Against Sensor/Actuator Faults Applied to Autonomous Underwater Vehicle Dynamics." In *A Closer Look at Fault Tolerant Control*, edited by Jeremy M. Hutton, 173–209. New York- Nova Science Publishers Inc.

[34] Hajiyev, C., Cilden-Guler, Hacizade, U. (2020). Two-Stage Kalman Filter for Fault Tolerant Estimation of Wind Speed and UAV Flight Parameters. *Measurement Scıence Revıew*, Vol. 20, No. 1, pp. 35-42. doi: 10.2478/msr-2020-0005.

Sitki Yenal Vural

Affiliation: Istanbul Technical University.

Education: Istanbul Technical University.

Business Address: Aeronautics and Astronautics Faculty, Istanbul Technical University, Maslak, 34469, Istanbul, Turkey.

Research and Professional Experience: He is a PhD student at the Aeronautical Engineering Department at Istanbul Technical University. He graduated from the Faculty of Mechanical Engineering, Istanbul Technical University in 2002. He received his MSc degree in Aerospace Engineering from Istanbul Technical University in 2008. His research interests include unmanned aerial vehicles (UAV), flight control techniques, Kalman filtering applications and fault-tolerant flight control.

Professional Appointments: Sitki Yenal Vural is a PhD student at the Aeronautical Engineering Department at Istanbul Technical University. He worked in Turkish Aerospace Industries (TAI) as an Aeronautics-Avionics Systems Design Engineer between 2010 and 2012. He has been working in Turkish Airlines since then.

Honors: 2012: Tubitak Young Researcher Scholarship, 2012 (Tubitak project under grant 109M702).

Publications from the Last 3 Years:

[1] Hajiyev, C., Vural, S. Y., Shumsky A., Zhirabok, A. (2018). Robust Kalman Filter Based Estimation of AUV Dynamics. *IFAC-PapersOnLine*, 51(30), 424-429.

[2] Vural, S. Y., Janset Dasdemir J., Hajiyev C. (2018). Passive Fault Tolerant Lateral Controller Design for an UAV. *IFAC-PapersOnLine*, 51(30), 446-451.

[3] Hajiyev, C., Vural, S. Y. (2019). Active Fault Tolerant Lateral Control Against Actuator Faults Applied to AUV Dynamics. *Proc. 4th Conference on Control and Fault Tolerant Systems (SysTol),* Casablanca, Morocco, September 18-20, 2019, pp. 153-158.

[4] Hajiyev, C., Vural, S. Y. (2020). "Reconfigurable Fault Tolerant Control Against Sensor/Actuator Faults Applied to Autonomous Underwater Vehicle Dynamics." In *A Closer Look at Fault Tolerant Control,* edited by Jeremy M. Hutton, 173–209. NewYork- Nova Science Publishers Inc.

Ulviye Hacizade

Affiliation: Halic University, Istanbul, Turkey.

Education: Azerbaijan State Pedagogical University.

Business Address: Department of Computer Engineering, Faculty of Engineering, Halic University, Imrahor Str. No: 82, 34445 Beyoglu, Istanbul, Turkey.

Research and Professional Experience: In 1995 she graduated from the Computer Engineering Department, Azerbaijan State Pedagogical University (Baku, Azerbaijan) with honor diploma. He received the PhD degree in Process Control from Azerbaijan Technical University (Baku, Azerbaijan), in 2003.

Professional Appointments: From 1995 to 2005 she worked as a Scientific Worker at the Azerbaijan State Petrol Academy. Since 2006 Hajiyeva has been with the Department of Computer Engineering, Halic University (Istanbul, Turkey), where she is currently an Assistant Professor. From 2016 she is Deputy Head of Computer Engineering Department of the Halic University.

Honors: 1995: Honors Certificate at Azerbaijan State Pedagogical University (Baku, Azerbaijan).

Publications from the Last 3 Years:

[1] Hajiyev, C., Hacizade U. (2018). Testing the Determinant of the Innovation Covariance Matrix Applied to Aircraft Sensor and Actuator/Surface Fault Detection. In: Karakoç T., Colpan C., Şöhret Y. (eds) *Advances in Sustainable Aviation.* Springer, Cham, pp. 269-284. DOI: https://doi.org/10.1007/978-3-319-67134-5_19.

[2] Hacizade, U. (2019). Genetic Algorithm Based Sensor Calibration. *Pamukkale University Journal of Engineering Sciences,* Vol. 25, Iss. 5, pp. 929-933.

[3] Hajiyev, C., Cilden-Guler, D., Hacizade, U. (2019). Two-Stage Kalman Filter for Estimation of Wind Speed and UAV States by using GPS, IMU and Air Data System. *WSEAS Transactions On Electronics,* Vol. 10, pp. 60-65.

[4] Hajiyev, C., Cilden-Guler, D., Hacizade, U. (2019). EKF for Wind Speed Estimation and Sensor Fault Detection Using Pitot Tube Measurements. *Proc. 9th International Conference on Recent Advances in Space Technologies "Space for the Sustainable Development Goals" (RAST-2019),* 11-14 June 2019, Istanbul, Turkey, *IEEE,* pp. 887-893.

[5] Hajiyev, C., Cilden-Guler, D., Hacizade, U. (2019). Two-Stage Kalman Filter for Estimation of Wind Speed and UAV Flight Parameters Based on GPS/INS and Pitot Tube Measurements. *Proc. 9th International Conference on Recent Advances in Space Technologies "Space for the Sustainable Development Goals" (RAST-2019),* 11-14 June 2019, Istanbul, Turkey, *IEEE,* pp. 875-880.

[6] Haciyev, C., Hacizade, U., Cilden-Guler, U. (2019). Data Fusion for Integrated Baro/GPS Altimeter. *Proc. 9th International Conference on Recent Advances in Space Technologies "Space for the Sustainable*

Development Goals" (RAST-2019), 11-14 June 2019, Istanbul, Turkey, *IEEE*, pp. 881-885.

[7] Hacizade, U. (2019). Comparison of the Genetic Algorithm and the Least Squares Method for Determining the Calibration Function of a Differential Pressure Sensor. *Measurement Techniques*, Vol. 61, No. 11, pp. 1081-1085.

[8] Hacizade, U. (2019). Genetic Algorithm and Least Square Method Based Calibration of Measurement Apparatus. *IET Science Measurement and Technology*, Vol. 13, Iss. 5, pp. 715-721.

[9] Hajiyev, C., Cilden-Guler, Hacizade, U. (2020). Two-Stage Kalman Filter for Fault Tolerant Estimation of Wind Speed and UAV Flight Parameters. *Measurement Science* Review, Vol. 20, No. 1, pp. 35-42. DOI: 10.2478/msr-2020-0005

In: Underwater Vehicles
Editor: George M. Roman

ISBN: 978-1-53618-876-9

Chapter 2

FROM NON-MODEL-BASED TO ADAPTIVE MODEL-BASED TRACKING CONTROL OF LOW-INERTIA UNDERWATER VEHICLES

Auwal Shehu Tijjani and Ahmed Chemori*
LIRMM, University of Montpellier, CNRS, Montpellier, France

Abstract

This chapter investigates the performances of different control schemes, from non-model-based (proportional-integral-derivative control, PID) to model-based (computed torque control, CT) as well as adaptive model-based (adaptive proportional-derivative plus control, APD+), implemented on a low-inertia underwater vehicle for three-dimensional (3D) helical trajectory tracking. Then, the asymptotic stability of the resulting closed-loop dynamics for each control scheme is proven based on the Lyapunov direct method. The performances of the control schemes, implemented on the Leonard underwater vehicle for 3D helical trajectory tracking, are then demonstrated through scenarios-based numerical simulations. The proposed simulations are conducted under the influences of the vehicle's buoyancy and damping changes, parametric variations; sensor noise, internal vehicle's perturbations; and water current, external disturbances rejection. Moreover, we demonstrate the task of transporting an object by the vehicle during underwater missions. The obtained simulation results show the effectiveness and robustness of the APD+ control

*Corresponding Author's Email: Ahmed.Chemori@lirmm.fr.

scheme for tracking control of the low-inertia underwater vehicle in marine applications, outperforming the other controllers.

Keywords: non-model-based control, model-based control, adaptive model-based control, computed torque control, PID, APD+, stability analysis, low-inertia underwater vehicle

1. INTRODUCTION

1.1. Context

The high demand for raw materials on the land surface due to the rapid technological advances in industry and research activities broaden the exploration and exploitation of subsea environments. These raw materials includes: crude oil and natural gas, solid minerals (nickel, silver, copper, gold, cobalt etc), aquatic plants and animals [1]. Being an alternative natural source of raw materials currently and in the near future, approximately less than 10% of the subsea is explored and exploited by human either for civil or military purposes [2], [3], [4]. Some of the characteristics that make the subsea challenging to explore and exploit by the human despite its abundance deposit of raw materials include: poor visibility especially at higher depths, poor or impossible electromagnetic transmission which hinders online communications, highly dynamical and unstructured nature of the environment, as well as the impact of waves and water currents [5].

Being motivated by the challenges of the subsea as well as the high demand for raw materials, research communities proposed using divers for exploring and exploiting this environment. The proposed solution was associated to inherent challenges such as putting the lives of the divers at risk, the expensive cost, the time-consuming, the low efficiency, etc. Based on these limitations, other groups of research communities proposed using manned underwater vehicles (MUVs) for exploring and exploiting the subsea environment. Although the proposed idea was a big step forward, it was also associated with other challenges. For instance, when MUVs got stuck in a confined environment, the lives of the personnel inside will be at risk such as the real-life scenario happen with an MUV carrying seven personnel identified as $AS-28$. The vehicle was trapped 15 years ago by underwater radar cables in the pacific ocean at a depth of approximately 250m from the surface [5]. In view of the challenges faced by

MUVs, with the recent technological advances in computational power of microprocessors, sensors, battery systems and vision system, unmanned underwater vehicles (UUVs) are becoming the ultimate tool for exploring and exploiting subsea environments [6].

In general, UUVs can be classified into remotely operated underwater (ROVs) and autonomous underwater vehicles (AUVs) [6]. During the exploration and exploitation of the subsea, either for the civil or military purposes, the mission may involve operations such as seafloor mapping, drilling, monitoring, inspection, debris cleaning, search and rescue, etc. These missions may require the ability of the vehicle to make autonomously an intelligent decision. The autonomous behaviours for instance could be station keeping, spatial trajectory tracking, collision avoidance, desired velocity profile regulation, and so on [7], [8]. The presence of intelligent behaviour in AUV broadens its operational context in subsea missions [7]. In this chapter, we focus on the spatial trajectory tracking case. Even though designing an onboard control scheme combining several autonomous behaviours for AUVs remains a challenging task and an open research problem.

1.2. Related Work

Despite, the challenge of designing an onboard control scheme for AUVs, various contributions have been proposed by several research communities to resolve the problems of station keeping, spatial trajectory tracking, collision avoidance, desired velocity profile regulation and classical path-following. Focusing on trajectory tracking and station keeping problems, some of the proposed classical non-model-based controllers include: classical PD and PID control schemes for position and velocity regulation, respectively, of a fully actuated AUVs, that have been proposed in [9]. Even though the authors focused the work on control design and the stability analysis, the analytical stability analysis of the control schemes designed are not conducted. Similarly, in [10], the authors demonstrated the application of a PID control scheme for depth motion control of micro-AUVs swamps through simulation and real-time experiment. However, the obtained results show that the PID control scheme is oscillatory at steady-state. For this reason, the authors proposed a bounded PD control scheme to deal with this effect, and the proposed control scheme was validated through simulations and real-time experiments. A real-time station keeping problem was also addressed in [11] using classical PID control. Moreover, the depth and heading control using a classical PID control scheme for Amogh AUV has been proposed in [12].

Although PID control scheme demonstrated some level of AUV's tracking control performance in the literature, keeping in mind the dynamical nature of the subsea and the vehicle's dynamics nonlinearity, this non-model-based control scheme will certainly not be able to solve all the trajectory tracking and station keeping problems for AUVs especially in high precision applications. For this reason, improved non-model-based control schemes such as fuzzy logic-based PID [13, 14], GA-based PID [15], saturation-based nonlinear PD/PID [16], classical RISE control, etc. have been proposed. To improve the performance of the classical PID controller for tracking control of mini-ROVs in subsea applications, the idea of auto-adjustment of the feedback gains using neural networks has been proposed in [17]. Although the authors demonstrated the performance of the control scheme through simulations and real-time experiments, the neural networks are always associated with long training time and high computational cost. Also, in [18], a PID control approach has been proposed for depth and yaw tracking of UUVs. To improve the performance of the PID, the authors proposed using fuzzy gain scheduling to design the controller at various operating points with optimal gains.

So far, the improved non-model-based controllers show superior performance over classical non-model-based controllers, such as classical PD and PID in trajectory tracking and station keeping for UUVs; however, having some knowledge about the AUVs dynamics will certainly help to improve the performance of the designed control scheme for these vehicles. Consequently, model-based classical control (saturation-based nonlinear fractional order PD, nonlinear PD based on variable saturation function, etc.) as well as model-based robust control (nonlinear RISE, sliding mode, high-order sliding mode, etc.) have been investigated in [19], [20] and [21], [22] respectively. Also, in [23] exact linearisation and nonlinear model-based controllers have been proposed for set-point regulation and trajectory tracking tasks. The performances of the proposed control schemes are evaluated through real-time experiments using Johns Hopkins University remotely operated vehicle (JHU-ROV). However, the computational time of the proposed controllers can be reduced using desired compensation in the control schemes, which could be computed offline. Similarly, a nonlinear model-based controller for six degrees of freedom position and velocity tracking has been proposed in [24]. The control scheme has been implemented on the fully actuated JHU-ROV through both numerical simulations and real-time experiments. The obtained results show the better performance of the nonlinear model-based controller when compared with a non-model-based controller. A robust fuzzy controller for ROVs has been proposed by [25]. In the control scheme, the membership functions are adjusted using genetic algorithms, which modified the gains of the controller based on the task complexity assigned to the ROV. Similarly, an optimised fuzzy controller for path tracking of an underwater vehicle has been proposed in [26], and validated experimentally on Sea-Dog underwater vehicle. Also, three-dimensional spatial tracking control of a hybrid AUV under the influence of underwater currents has been addressed in [27].

In spite of the notable performances of model-based robust control schemes, in some subsea missions, their performances may be degraded drastically due to inherent uncertainties in subsea environments, as well as in the vehicles themselves. To deal with these effects, several research communities proposed using control schemes able to dynamically adjust themselves in real-time. Indeed, the proposed idea opens another interesting field of research known as adaptive control; based on the notion that the auto-adjustment will not only maintain but also improve the desired control system performance.

In the context of underwater vehicles, adaptive control schemes have been proposed by several research studies. For instance, an adaptive thruster fault tolerant region tracking control with prescribed transient performance has been investigated in [28]. Even though factors such as thruster fault, measurement noise, parameter uncertainties and underwater currents were considered; additional cases could be added to ascertain the effectiveness of the proposed scheme. Similarly, adaptive tracking control and its improvement using a disturbance observer for underwater vehicles have been proposed in [29] and [8] respectively. Besides, a variable forgetting factor model-free adaptive control for surface unmanned vehicles has been studied in [30]; this scheme could be extended to the case of UUVs. Output constraints fuzzy-based adaptive tracking control for autonomous underwater vehicles was investigated in [31]; where numerical simulations were carried out to show the effectiveness of the proposed scheme. An adaptive formation control based on output-feedback for an underactuated surface vehicle has been proposed in [32]. However, this scheme does not consider measurement noise. In [33], an indirect adaptive control scheme for intervention operations of AUV has been proposed. The robustness of the control scheme is enhanced with an extended Kalman filter (EKF), which is used to take care of external disturbances, parametric uncertainties, payload variations, sensor noise and actuator nonlinearity. Despite the proposed complex adaptive control schemes in the literature, still classical adaptive schemes are dominating most of the real-time marine applications.

1.3. Chapter Contribution and Organisation

In this chapter, we propose to investigate design, stability analysis and effectiveness of tracking control schemes, from non-model-based to model-based as well as adaptive model-based and their application to control a low-inertia underwater vehicle for marine missions. These vehicles are characterised with high power to weight ratio, which makes them vulnerable towards any slight variation in the system parameter.

The remaining parts of this chapter are organised as follow. In Section 2, the low-inertia underwater vehicle description as well as its six-degree-of-freedom modelling are introduced. Then, Section 3 is devoted to the proposed tracking control schemes and their stability analysis. Numerical simulation results are presented and discussed in Section 4, while Section 5 finalises the chapter with some concluding remarks and future works.

2. Vehicle Description and Modelling

2.1. Vehicle Description

To validate our proposed investigations in this chapter, we perform numerical simulation using a LIRMM's underwater vehicle known as Leonard. Even though some specific features of this vehicle are well described in [34] and [8]; we recall some of these essential features again to facilitate kinematics and dynamics formulations of the vehicle in this chapter. The vehicle can be categorised as a low-inertia hybrid underwater vehicle, that is, having both remote and autonomous operation capabilities. Additionally, being a holonomic system can be suitable for various marine missions. The vehicle's translational and rotational motions are determined by its thrusters' allocation illustrated in Figure 1.

Besides, the vehicle is equipped with six thrusters, energy consumption is minimised by keeping neutrally both the vehicle's pitch and roll close to zero with respect to the horizontal. Table 1 summarises some of the vehicle's hardware components, as well as its parameters.

Figure 1. View of Leonard underwater vehicle thrusters' allocation, which produces forces responsible for the navigation of the vehicle.

Table 1. The main technical specifications of Leonard low-inertia underwater vehicle

Hardware Components and Parameters	**Descriptions**
Attitude Sensor	Sparkfun MPU 9250, MEMS 9-axes gyrometer, accelerometer and magnetometer microprocessor.
Depth Sensor	MS5803-14BA (Pressure Sensor).
Dimensions	0.75m (l) $\times$ 0.55m (w) $\times$ 0.45m (h).
Floatability	9N.
Mass	28kg.
Maximal Depth	100m.
Power	48V - 600W.
Sampling Period	0.05s.
Tether Length	150m.
Thrusters	6-Seabotix BTD150.

2.2. Vehicle Modelling in Six Degrees of Freedom

The kinematics and dynamics of a low-inertial underwater vehicle such as Leonard can be derived with respect to 3D reference frames. These frames are the earth-fixed and the body-fixed frames. Figure 2 illustrates the frames

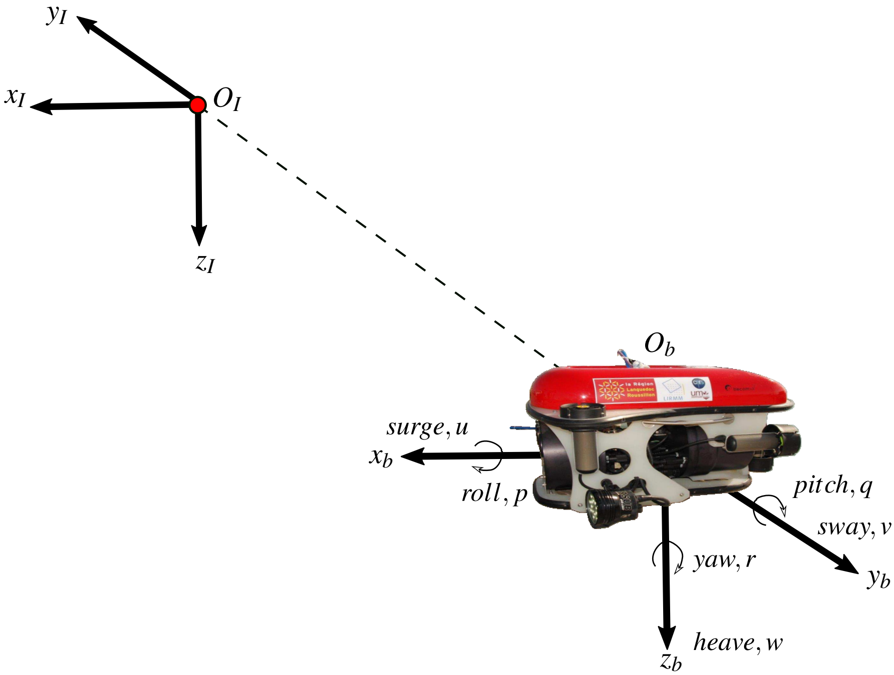

Figure 2. Illustration of the earth-fixed frame (O_I, x_I, y_I, z_I) and the body-fixed frame (O_b, x_b, y_b, z_b) frames assignment for kinematic and dynamic modelling.

assignment, for guidance and navigation of Leonard underwater vehicle, using SNAME (Society of Naval Architects and Marine Engineers) standard [34].

2.2.1. Vehicle Kinematics in Six Degrees of Freedom

For a rigorous kinematic formulation based on Figure 2, we can express the time derivatives of the vehicle's position and orientation in the earth-fixed frame with respect to its linear and angular velocities in vehicle's body-fixed frame as follows:

$$\dot{\eta} = J(\eta)\nu \tag{1}$$

where $\nu = [\nu_1\ \nu_2]^T$ is the vector of linear and angular velocities in the body-fixed frame, $\nu_1 = [u\ v\ w] \in \mathbb{R}^{3\times1}$ and $\nu_2 = [p\ q\ r] \in \mathbb{R}^{3\times1}$, $\eta = [\eta_1\ \eta_2]^T$ denotes the vector of position and orientation in the earth-fixed frame, $\eta_1 = [x\ y\ z] \in \mathbb{R}^{3\times1}$ and $\eta_2 = [\phi\ \ \theta\ \ \psi] \in \mathbb{R}^{3\times1}$, while $J(\eta) \in \mathbb{R}^{6\times6}$ is a matrix of the 3D spatial transformation between the earth-fixed frame and body-fixed frame.

This transformation matrix $J(\eta)$ is given by [35]:

$$J(\eta) = \begin{bmatrix} J_1(\eta_2) & 0_{3\times 3} \\ 0_{3\times 3} & J_2(\eta_2) \end{bmatrix} \tag{2}$$

where $J(\eta_1)$ and $J(\eta_2)$ are given by (3) and (4) respectively, as follows (see [35] for further details):

$$J_1(\eta_2) = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix} \tag{3}$$

$$J_2(\eta_2) = \begin{bmatrix} 1 & s\psi t\theta & c\psi t\theta \\ 0 & c\psi & -s\psi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix} \tag{4}$$

with *c* angle, *s* angle and *t* angle representing *cos* angle, *sin* angle and *tan* angle functions respectively, where angle = ϕ = θ = ψ.

2.2.2. Vehicle Dynamics in Six Degrees of Freedom

Many research studies have well described the dynamics of an underwater vehicle [34], [5]. Inspired by these research studies and representation proposed by [35], the dynamics describing the motion of our underwater vehicle, based on SNAME notations in the vehicle's body-fixed frame, can be written as follows:

$$M\dot{\nu} + C(\nu)\nu + D(\nu)\nu + g(\eta) = \tau + w_{ext}(t) \tag{5}$$

where $M \in \mathbb{R}^{6\times 6}$ defines the inertia matrix including the added mass effects, $C(\nu) \in \mathbb{R}^{6\times 6}$ represents the Coriolis and centripetal matrix, $D(\nu) \in \mathbb{R}^{6\times 6}$ is the hydrodynamic damping matrix including both linear and quadratic effects, $g(\eta) \in \mathbb{R}^{6\times 1}$ defines the vector of restoring forces and moments, $\tau \in \mathbb{R}^{6\times 1}$ represents the vector of control inputs and $w_{ext}(t) \in \mathbb{R}^{6\times 1}$ is the vector of time-varying external disturbances.

Additionally, the matrices and vectors defined in the vehicle's dynamics (5) are described as follows:

The total contributions of the vehicle's rigid-body inertia M_{RB} and the inertia of the added mass M_A constitute the so-called inertia matrix M. This matrix can be written as:

$$M = M_{RB} + M_A \tag{6}$$

Based on the assumption that we consider the motion of the vehicle at low-speed, the matrix M can be simplified as follows:

$$M = diag\{m+X_{\dot{u}},\ m+Y_{\dot{v}},\ m+Z_{\dot{w}},\ I_{xx}+K_{\dot{p}},\ I_{yy}+M_{\dot{q}},\ I_{zz}+N_{\dot{r}}\} \tag{7}$$

where m is the mass of the vehicle, $\{I_{xx}, I_{yy}, I_{zz}\}$ are the vehicle's rigid-body moments of inertia and $\{X_{\dot{u}}, Y_{\dot{v}}, Z_{\dot{w}}, K_{\dot{p}}, M_{\dot{q}}, N_{\dot{r}}\}$ are the hydrodynamics added masses.

Similarly, the Coriolis and centripetal matrix is usually expressed as (see [35] for more details):

$$C(\nu) = C_{RB}(\nu) + C_A(\nu) \tag{8}$$

where $C_{RB}(\nu)$ and $C_A(\nu)$ denote the Coriolis and centripetal (rigid-body and hydrodynamics) matrices, which are given by (9) and (10) as follows:

$$C_{RB}(\nu) = \begin{bmatrix} 0 & -mr & mq & 0 & 0 & 0 \\ mr & 0 & -mp & 0 & 0 & 0 \\ -mq & mp & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{zz}r & -I_{yy}q \\ 0 & 0 & 0 & -I_{zz}r & 0 & I_{xx}p \\ 0 & 0 & 0 & I_{yy}q & -I_{xx}p & 0 \end{bmatrix} \tag{9}$$

$$C_A(\nu) = \begin{bmatrix} 0 & 0 & 0 & 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v \\ 0 & 0 & 0 & Z_{\dot{w}}w & 0 & -X_{\dot{u}}u \\ 0 & 0 & 0 & -Y_{\dot{v}}v & X_{\dot{u}}u & 0 \\ 0 & -Z_{\dot{w}}w & Y_{\dot{v}}v & 0 & -N_{\dot{r}}r & M_{\dot{q}}q \\ Z_{\dot{w}}w & 0 & -X_{\dot{u}}u & N_{\dot{r}}r & 0 & -K_{\dot{p}}p \\ -Y_{\dot{v}}v & X_{\dot{u}}u & 0 & -M_{\dot{q}}q & K_{\dot{p}}p & 0 \end{bmatrix} \tag{10}$$

The detail step-by-step process of obtaining approximate values of the hydrodynamics elements (X_u, Y_v, Z_w, K_p, M_q, N_r) of the vehicle's $D(\nu)$ matrix is addressed in [34]. Considering the low-speed motion of the vehicle, we can approximate the damping matrix $D(\nu)$ as follows:

$$D(\nu) = diag\{X_u,\ Y_v,\ Z_w,\ K_p,\ M_q,\ N_r\} \tag{11}$$

Concerning the restoring forces and moments $g(\eta)$, we assume that the centre of gravity coincides with the centre of the vehicle; as a result, the vector $g(\eta)$

can be written as:

$$g(\eta) = \begin{bmatrix} f_b s\theta \\ -f_b c\theta s\psi \\ -f_b c\theta c\psi \\ -z_{cb} B c\theta s\psi \\ -z_{cb} B s\theta \\ 0 \end{bmatrix} \tag{12}$$

where B = Buoyancy, while f_b and z_{cb} are the buoyancy force and the position of the centre of buoyancy of the vehicle, respectively.

We finalise the description of the vehicle's dynamics terms with τ, which is a control inputs vector responsible for the translational and rotational motions of the vehicle. A particular motion pattern is possible through actuating a precise vehicle's thrusters configuration. The control input vector τ can be written as follows:

$$\tau = B^{\star} \cdot F^{\star} \tag{13}$$

where $B^{\star} \in \mathbb{R}^{6\times 6}$ is the thrusters' allocation matrix, which maps all the control inputs to their corresponding forces and moments for translational and rotational motions of the vehicle, and $F^{\star} = [F_1 \;\; F_2 \;\; F_3 \;\; F_4 \;\; F_5 \;\; F_6]^T$ is a vector of the forces generated by the six thrusters of the vehicle.

3. Proposed Control Solutions and Their Stability Analysis

3.1. Control Solution 1: A Non-Model-Based Tracking Control Scheme

A control scheme that can be designed based only on the system states is referred to as a non-model-based control scheme [36]. It does not require any prior information on the system dynamics. There are many control schemes proposed in the literature based on non-model-based structures. However, the most famous scheme, widely used in industry, is the conventional proportional-integral-derivative (PID) control scheme. Besides its implementation simplicity, this approach works satisfactorily in many industrial applications [37]. Regarding low-inertia underwater water vehicles, PID and PD control schemes have widely been used in most of the real-time marine applications [38], [16]; therefore in this section, we focus on the control structure based on the PID algorithm as our non-model-based control scheme case study.

3.1.1. Background on PID Control Scheme

The classical PID control scheme has the following structure:

$$U(t) = K_p e(t) + K_i \int_0^t e(\sigma) d\sigma + K_d \left[\frac{de(t)}{dt} \right] \tag{14}$$

where $U(t)$ is the control signal, $e(t)$ defines the error signal, which is obtained as the difference between the reference signal $r(t)$ and output to be controlled $y(t)$, while K_p, K_i and K_d are respectively the proportional, integral and derivative feedback gains of the controller. Even though the feedback gains can be selected easily during the implementation of the control scheme, the selection of optimal feedback gains is a nontrivial task. Where each gain has a particular effect on the system's behaviour; for instance, an optimal value of K_p decreases the response time and steady-state error of the closed-loop system, optimal K_i value removes steady-state error and K_d improves the stability through increasing the damping of the resulting closed-loop dynamics.

On the other hand, non-optimal gains selection may lead to the instability of the resulting closed-loop dynamics. Several techniques have been proposed in the literature for a relevant tuning of these feedback gains (see for instance, [39]).

3.1.2. Application of the PID on Leonard Underwater Vehicle

We proposed to apply the classical PID structure given by (14) to our nonlinear coupled six degrees of freedom underwater vehicle described in (5). The controller is aimed to guide the vehicle to track the desired trajectories defined as follows:

$$\eta_d(t) = [x_d(t),\ y_d(t),\ z_d(t),\ \phi_d(t),\ \theta_d(t),\ \psi_d(t)]^T \tag{15}$$

If we write the vehicle's trajectories as:

$$\eta(t) = [x(t),\ y(t),\ z(t),\ \phi(t),\ \theta(t),\ \psi(t)]^T \tag{16}$$

Then, the tracking error $e(t)$ can be expressed as follows:

$$e(t) = \eta(t) - \eta_d(t) \tag{17}$$

where $e(t)$ is a vector of the tracking errors of all the six degrees of freedom, and is expressed as, $e(t) = [e_1(t), e_2(t), ..., e_6(t)]^T$, while $\eta_d(t)$ and $\eta(t)$ are the desired and actual trajectories given by (15) and (16), respectively.

The control input vector τ to be applied to our underwater vehicle is designed as follows:

$$\tau = -J^T(\eta)[\tau_{PID}] \tag{18}$$

where, the PID control law τ_{PID} can be expressed as follows:

$$\tau_{PID} = K_p e(t) + K_i \int_0^t e(\sigma)d\sigma + K_d \left[\frac{de(t)}{dt}\right] \tag{19}$$

where $\tau = [\tau_x,\ \tau_y,\ \tau_z,\ \tau_\phi,\ \tau_\theta,\ \tau_\psi]^T$ is the vector of the control inputs for all the six degrees of freedom, $e(t)$ is the vector of the tracking errors, while $K_p = diag\{k_{1p}, k_{2p}, ..., k_{6p}\} > 0$, $K_i = diag\{k_{1i}, k_{2i}, ..., k_{6i}\} > 0$ and $K_d = diag\{k_{1d}, k_{2d}, ..., k_{6d}\} > 0$ are the PID feedback gains matrices.

The above designed PID-based control scheme can be illustrated by the block diagram of Figure 3.

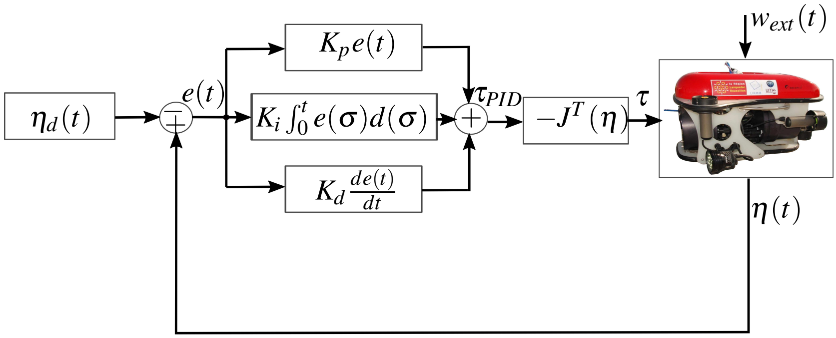

Figure 3. Block diagram of the non-model-based PID control scheme implemented on Leonard underwater vehicle.

3.1.3. Stability Analysis

To facilitate the stability analysis, let us consider the transformation of (5) into the earth-fixed frame (O_I, x_I, y_I, z_I) using (1) as follows:

$$M^\star(\eta)\ddot{\eta} + C^\star(\nu,\eta)\dot{\eta} + D^\star(\nu,\eta)\dot{\eta} + g^\star(\eta) = \tau^\star(\eta) + w_{ext}(t) \tag{20}$$

where

$M^\star(\eta) = J^{-T}(\eta) M J^{-1}(\eta),$
$C^\star(\nu,\eta) = J^{-T}(\eta)[C(\nu) - MJ^{-1}(\eta)\dot{J}(\eta)]J^{-1}(\eta),$
$D^\star(\nu,\eta) = J^{-T}(\eta)D(\nu)J^{-1}(\eta),$
$g^\star(\eta) = J^{-T}(\eta)g(\eta),$
$\tau^\star(\eta) = J^{-T}(\eta)\tau$

Assumption 1. The external disturbance $w_{ext}(t)$, including water waves and currents, is assumed to be Lipschitz continuous. Also, its time derivative exists and is bounded: $\left|\dot{w}_{i_{ext}}(t)\right| \leq L_i, \;\; i = \overline{1,6}.$

Substituting (18) into (20), yields:

$$M^\star(\eta)\ddot{\eta} = -C^\star(\nu,\eta)\dot{\eta} - D^\star(\nu,\eta)\dot{\eta} - g^\star(\eta) + w_{ext}(t) - \tau_{PID} \tag{21}$$

Before substituting (19) into (21), the integral term of τ_{PID} introduces an auxiliary state variable, which leads to the modification of (19) as follows:

$$\tau_{PID} = K_p e + K_i \zeta + K_d \dot{e} \tag{22}$$

where, $\zeta = \int_0^t e(\sigma)d\sigma$ is the auxiliary state variable and $\dot{e}$ is the time derivative of (17). Then, we can adopt the following change of variable [40]:

$$z = a\zeta + e \tag{23}$$

where $a > 0$ and $z = [z_1, z_2, ..., z_6]^T$.

Using this change of variable, (22) can be rewritten as follows:

$$\tau_{PID} = K_p^\star e + K_i^\star z + K_d \dot{e} \tag{24}$$

where $K_p^\star = K_p - \frac{1}{a}K_i$ and $K_i^\star = \frac{1}{a}K_i$.

By substituting (24) into (21), the resulting vehicle's closed-loop dynamics can be rewritten as follows:

$$\ddot{\eta} = M^\star(\eta)^{-1}[-C^\star(\nu,\eta)\dot{\eta} - D^\star(\nu,\eta)\dot{\eta} - g^\star(\eta) + w_{ext}(t) - K_p^\star e - K_i^\star z - K_d \dot{e}] \tag{25}$$

Then, (25) can be written in state-space form with a unique equilibrium point as follows:

$$\frac{d}{dt}\begin{bmatrix} e \\ \dot{e} \\ z \end{bmatrix} = \begin{bmatrix} \dot{e} \\ M^{\star}(\eta)^{-1}[-C^{\star}(\nu,\eta)\dot{\eta} - D^{\star}(\nu,\eta)\dot{\eta} - g^{\star}(\eta) + w_{ext}(t) - K_p^{\star}e - K_i^{\star}z - K_d\dot{e}] - \ddot{\eta}_d \\ ae + \dot{e} \end{bmatrix} \tag{26}$$

To guarantee the stability of the unique equilibrium point of this state-space model, we propose to use the Lyapunov direct method by considering the following Lyapunov candidate function:

$$V(e,\dot{e},z) = \frac{1}{2}\dot{e}^T M^{\star}(\eta)\dot{e} + [g^{\star T}(\eta) + \dot{\eta}_d^T D^{\star}(\nu,\eta)]e + \int_0^e z^T K_p^{\star} dz + \int_0^e z^T K_i^{\star} dz \tag{27}$$

To prove that $V(e,\dot{e},z)$ is a positive definite function and radially unbounded, the term $\frac{1}{2}\dot{e}^T M^{\star}(\eta)\dot{e}$ is positive definite, since $M^{\star}(\eta)$ is a positive definite matrix; also, in the second term $D^{\star}(\nu,\eta) > 0$ and it is possible to design η_d such that $\dot{\eta}_d > 0$. For the integral terms, we consider the following arguments [16]:

$$\int_0^e z^T K_p^{\star} dz = \int_0^{e_1} z_1^T k_{1p}^{\star} dz_1 + \int_0^{e_2} z_2^T k_{2p}^{\star} dz_2 + \ldots + \int_0^{e_6} z_6^T k_{6p}^{\star} dz_6 \tag{28}$$

$$\int_0^e z^T K_p^{\star} dz > 0, \quad \forall \ e \neq 0 \in \mathbb{R}^n \tag{29}$$

where $K_p^{\star} = diag\{k_{1p}^{\star}, k_{2p}^{\star}, \ldots, k_{6p}^{\star}\}$.

From the arguments (28) and (29), we can deduce that:

$$\int_0^e z^T K_p^{\star} dz \to \infty \quad \text{as } \ \|e\| \to \infty \tag{30}$$

Similarly, it is possible to apply the same above arguments to the second integral term of $V(e,\dot{e},z)$ as follows:

$$\int_0^e z^T K_i^{\star} dz > 0, \quad \forall \ e \neq 0 \in \mathbb{R}^n \tag{31}$$

leading to

$$\int_0^e z^T K_i^{\star} dz \to \infty \quad \text{as } \ \|e\| \to \infty \tag{32}$$

From (27), the time derivative of $V(e,\dot{e},z)$ can be expressed as follows:

$$\dot{V}(e,\dot{e},z) = \dot{e}^T M^{\star}(\eta)\ddot{e} + \frac{1}{2}\dot{e}^T \dot{M}^{\star}(\eta)\dot{e} + e^T K_p^{\star}\dot{e} + z^T K_i^{\star}\dot{e} + g^{\star T}(\eta)\dot{e} + \dot{\eta}_d^T D^{\star}(\nu,\eta)\dot{e} \tag{33}$$

Injecting the closed-loop state-space dynamics (26) into (33), yields:

$$\begin{aligned}\dot{V}(e,\dot{e},z) = -\dot{e}^T C^{\star}(\nu,\eta)\dot{\eta} - \dot{e}^T D^{\star}(\nu,\eta)\dot{\eta} - \dot{e}^T g^{\star}(\eta) + \dot{e}^T w_{ext}(t) - \dot{e}^T K_p^{\star} e \\ -\dot{e}^T K_i^{\star} z - \dot{e}^T K_d \dot{e} - \dot{e}^T M^{\star}(\eta)\ddot{\eta}_d + \frac{1}{2}\dot{e}^T \dot{M}^{\star}(\eta)\dot{e} + e^T K_p^{\star}\dot{e} + z^T K_i^{\star}\dot{e} \\ + g^{\star T}(\eta)\dot{e} + \dot{\eta}_d^T D^{\star}(\nu,\eta)\dot{e}\end{aligned} \tag{34}$$

Assumption 2. In this work, we consider that our vehicle moves at a low speed.

Based on Assumption 2, $\dot{M}^{\star}(\eta) = 0$ and $C^{\star}(\nu,\eta) \approx 0$, therefore (34) can be rewritten as follows:

$$\dot{V}(e,\dot{e},z) = -\dot{e}^T D^{\star}(\nu,\eta)\dot{e} - [\dot{e}^T K_d \dot{e} + \dot{e}^T M^{\star}(\eta)\ddot{\eta}_d - \dot{e}^T w_{ext}(t)] \tag{35}$$

From (35), if we consider Assumption 1, it is always possible to design K_d of the controller to compensate for the effect of $w_{ext}(t)$ as follows:

$$K_{id} > \frac{\|w_{ext}(t)\| - \|M^{\star}(\eta)\ddot{\eta}_d\|}{\|\dot{e}\|} \quad i = \overline{1,6} \tag{36}$$

where $K_d = diag\{k_{1d}, k_{2d}, ..., k_{6d}\}$

Also, from (35) we can deduce that $\dot{e}^T D^{\star}(\nu,\eta)\dot{e} > 0$, since $D^{\star}(\nu,\eta) > 0$. Therefore, if (36) is satisfied, then $\dot{e}^T D^{\star}(\nu,\eta)\dot{e} > 0$ will dominate the right-hand side of (35). Consequently, we can conclude that $\dot{V}(e,\dot{e},z)$ in (35) is negative semidefinite. In accordance with the LaSalle's invariance principle, the origin of the resulting closed-loop dynamics is asymptotically stable [34], [40].

Remark 1. Even though the PID controller proposed here is non-model-based, the process of tuning of its feedback gains is a nontrivial task. It can be noticed in (36) that having some knowledge of the system dynamics (for instance, inertia matrix $M^{\star}(\eta)$ in our case) may help to select better PID feedback gains, which could improve the overall performance of the controller.

3.2. Control Solution 2: A Model-Based Tracking Control Scheme

In various marine missions, the performance of non-model-based controllers is degraded due to external disturbances and parametric variations. Certainly, integrating the system dynamics (partially or entirely) into a non-model-based controller structure will help to improve its performance. Therefore, when a non-model-based control scheme contains the dynamics (partially or entirely) of the system it is known as a model-based control scheme [36]. However, obtaining accurate and simple dynamics of a system, having all the properties of the real system remains a challenging task. Concerning the control of our low-inertia underwater vehicle, we propose to focus our study on the computed torque (CT) control as an example of a model-based control scheme, which is based on the full knowledge of the vehicle's dynamics.

3.2.1. Background on the CT Control and Its Application on Leonard Underwater Vehicle

The majority of the real systems are represented mathematically by nonlinear differential equations which mainly result in a nonlinear closed-loop dynamics, when controlled with model-based controllers. However, the CT control scheme has the advantage of transforming the closed-loop dynamics of the nonlinear system into a linear closed-loop dynamics. As a result, we can use linear systems design tools to analyse the resulting linear closed-loop dynamics. Additionally, the CT controller can fulfil the tracking control objective without necessary an optimal tuning of the feedback gains [40]. For the tracking control of Leonard underwater vehicle, we propose to design the CT controller as follows:

$$\tau = J^T(\eta)\left[M^\star(\eta)\ddot{\eta}_d + C^\star(\nu,\eta)\dot{\eta} + D^\star(\nu,\eta)\dot{\eta} + g^\star(\eta) - M^\star(\eta)\left[K_p e(t) + K_d \frac{de(t)}{dt}\right]\right] \quad (37)$$

where $\tau = [\tau_x,\ \tau_y,\ \tau_z,\ \tau_\phi,\ \tau_\theta,\ \tau_\psi]^T$ is the vector of the control inputs for all the six degrees of freedom of the vehicle, $J(\eta)$ is the transformation matrix, $e(t)$ is the vector of the tracking errors, $K_p = diag\{k_{1p}, k_{2p}, ..., k_{6p}\} > 0$ and $K_d = diag\{k_{1d}, k_{2d}, ..., k_{6d}\} > 0$ are the feedback gains, $M^\star(\eta)$ defines the inertia matrix including the added mass effects, $C^\star(\nu,\eta)$ represents the Coriolis and centripetal matrix, $D^\star(\nu,\eta)$ is the hydrodynamic damping matrix including both linear and quadratic effects, and $g^\star(\eta)$ defines the vector of restoring forces and moments.

Indeed, the terms $M^\star(\eta)$, $C^\star(\nu,\eta)$, $D^\star(\nu,\eta)$ and $g^\star(\eta)$ in the CT control law (37) are obtained from the vehicle's dynamics (20). The block diagram of the CT controller structure implemented on Leonard underwater vehicle is illustrated in Figure 4.

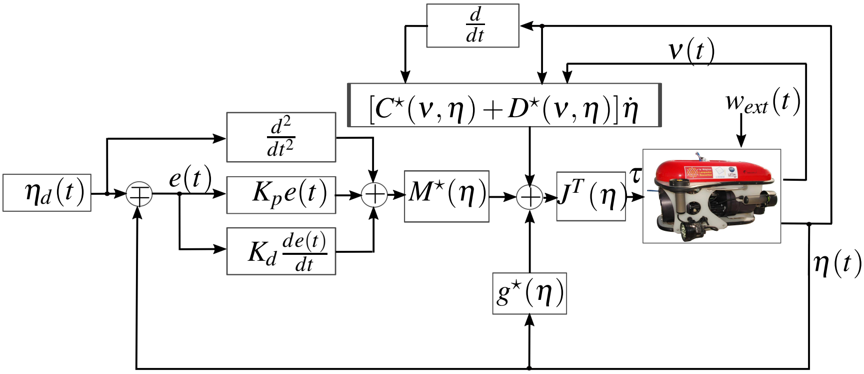

Figure 4. Block diagram of model-based CT control scheme implemented on Leonard underwater vehicle.

3.2.2. Stability Analysis

To facilitate the stability analysis, let us begin by injecting the CT control law (37) into the vehicle's dynamics (20), resulting in:

$$\begin{aligned} M^\star(\eta)\ddot{\eta}+C^\star(\nu,\eta)\dot{\eta}+D^\star(\nu,\eta)\dot{\eta}+g^\star(\eta) &= M^\star(\eta)\ddot{\eta}_d+C^\star(\nu,\eta)\dot{\eta}+D^\star(\nu,\eta)\dot{\eta} \\ &+g^\star(\eta)-M^\star(\eta)\left[K_p e(t)+K_d\frac{de(t)}{dt}\right]+w_{ext}(t) \end{aligned} \tag{38}$$

Then, we can rewrite the above closed-loop dynamics in the state-space form as follows:

$$\frac{d}{dt}\begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} \dot{e} \\ M^\star(\eta)^{-1}w_{ext}(t)-K_p e-K_d\dot{e} \end{bmatrix} \tag{39}$$

Next, we can use the Lyapunov direct method to prove the stability of the resulting closed-loop dynamics by considering the following Lyapunov candidate function:

$$V(e,\dot{e}) = \frac{1}{2}\dot{e}^T\dot{e} + \int_0^e \alpha^T K_p d\alpha \tag{40}$$

The proposed Lyapunov candidate function is positive definite and radially unbounded since $\frac{1}{2}\dot{e}^T M^\star(\eta)\dot{e}$ is positive definite, and the integral term satisfies the following arguments:

$$\int_0^e \alpha^T K_p d\alpha = \int_0^{e_1} \alpha_1^T k_{1p} d\alpha_1 + \int_0^{e_2} \alpha_2^T k_{2p} d\alpha_2 + \ldots + \int_0^{e_6} \alpha_6^T k_{6p} d\alpha_6 \tag{41}$$

$$\int_0^e \alpha^T K_p d\alpha > 0, \quad \forall \ e \neq 0 \in \mathbb{R}^n \tag{42}$$

where $K_p = diag[k_{1p}, k_{2p}, \ldots, k_{6p}]$.

From the above arguments (41) and (42), we can deduce that:

$$\int_0^e \alpha^T K_p d\alpha \to \infty \quad \text{as} \ \ \|e\| \to \infty \tag{43}$$

Next, since $V(e,\dot{e})$ is positive definite and radially unbounded, then we can evaluate its time derivative along the trajectory of the resulting closed-loop dynamics as follows:

$$\dot{V}(e,\dot{e}) = \dot{e}^T\ddot{e} + e^T K_p \dot{e} \tag{44}$$

Substituting (39) into (44) yields:

$$\dot{V}(e,\dot{e}) = \dot{e}^T[M^\star(\eta)^{-1} w_{ext}(t) - K_p e - K_d \dot{e}] + e^T K_p \dot{e} \tag{45}$$

which, we can be rewritten as follows:

$$\dot{V}(e,\dot{e}) = -[\dot{e}^T K_d \dot{e} - \dot{e}^T M^\star(\eta)^{-1} w_{ext}(t)] \tag{46}$$

From (46), and based on Assumption 1, K_d can be designed to compensate for the effect of $w_{ext}(t)$ as follows:

$$K_{id} > \frac{\left\|w_{ext}(t)\right\|}{min_i\left|\lambda_i\{M^\star(\eta)\}\right|\|\dot{e}\|}, \qquad i = \overline{1,6} \tag{47}$$

where $K_d = diag\{k_{1d}, k_{2d}, \ldots, k_{6d}\}$.

Finally, from (46) we can deduce that $\dot{V}(e,\dot{e})$ is negative semidefinite if argument (47) is satisfied. This leads to the conclusion that the origin of the closed-loop dynamics is asymptotically stable based on LaSalle's invariance principle.

3.3. Control Solution 3: Adaptive Model-Based Tracking Control Scheme

Even though having complete or partial knowledge of the system dynamics improves the performance of the control scheme, the process of obtaining an accurate model which represents real system remains a challenging task. Similarly, tracking control of a low-inertia underwater vehicle with a model-based controller may result in a high tracking error due to its challenging modelling process, in addition to the variations of the vehicle's parameters as well as the unpredictable nature of the underwater environments. Besides, the parametric variations, the high sensitivity of low-inertia underwater vehicles, the inherently coupled nonlinearities in their dynamics drastically affect the control schemes performances during marine missions. To deal with these issues, the designed controllers for such vehicles should dynamically adjust themselves to neutralise these effects in real-time. In the field of control systems, any controller with auto-adjustment mechanism is referred to as an adaptive controller [2]. Hence, an adaptive control technique can be considered as a process of designing a control scheme with an auto-adjustment mechanism for a dynamical system under the influence of parametric uncertainties in high precision applications [40]. However, the adaptive control scheme design requires accurate knowledge of the system dynamics structure, which is used to characterise the uncertainty of the system as a set of unknown parametric terms; this may help to facilitate the controller design. The adaptive control scheme can be categorised as a direct adaptive control technique or an indirect adaptive control technique. The direct adaptive control technique deals with direct estimation of the control parameters which are used to modify the system's dynamics. However, the indirect adaptive control technique involves the estimation of the system's dynamics, which is used in the design of the controller. For the case of our underwater vehicle in this section, we focus on the implementation of an adaptive version of PD+ controller (APD+) for trajectory tracking. This controller is designed and implemented on the vehicle subsequently.

3.3.1. Background on the APD+ Control Scheme and Its Application on Leonard Underwater Vehicle

A PD control structure combined with desire compensation terms, obtained from a system dynamics as well as a predefined desired trajectory in tracking control, is known as a PD+ control scheme. Besides, its implementation simplicity in real-time applications, the compensation terms in the control law can be computed offline to reduce computational cost once the desired trajectory is defined [40]. Considering these advantages of the PD+ control scheme, we modify its structure by adding an adaptation to improve its robustness. The improved control scheme is implemented on the highly uncertain dynamical model with a nonlinear coupled behaviour of Leonard underwater vehicle. The design and implementation of this control law as well as its adaptation mechanism for Leonard underwater vehicle are given as follows:

$$\tau = J^T(\eta)\left[M^\star(\eta)\ddot{\eta}_d + C^\star(\nu,\eta)\dot{\eta}_d + D^\star(\nu,\eta)\dot{\eta}_d - \left[\Phi_\vartheta \hat{\vartheta}^T + K_p e(t) + K_d \frac{de(t)}{dt}\right]\right] \tag{48}$$

$$\dot{\hat{\vartheta}} = \Gamma_\vartheta^{-1}\Phi_\vartheta\left[\alpha e(t) + \frac{de(t)}{dt}\right] \tag{49}$$

where $\tau = [\tau_x,\ \tau_y,\ \tau_z,\ \tau_\phi,\ \tau_\theta,\ \tau_\psi]^T$ is the vector of the six control inputs of the vehicle, $J(\eta)$ is a matrix which defines three dimensional spatial-transformation between the earth-fixed frame and vehicle's body-fixed frame, $M^\star(\eta)$ defines the inertia matrix including the added mass effects, $C^\star(\nu,\eta)$ represents the Coriolis and centripetal matrix, $D^\star(\nu,\eta)$ is the hydrodynamic damping matrix including both linear and quadratic effects, $e(t)$ is the vector of the tracking errors, $K_p = diag\{k_{1p}, k_{2p}, ..., k_{6p}\} > 0$ and $K_d = diag\{k_{1d}, k_{2d}, ..., k_{6d}\} > 0$ are the feedback gains, $\alpha > 0$, $\Gamma_\vartheta^{-1} = diag\{\gamma_1, \gamma_2, ..., \gamma_6\} > 0$ is the adaptation gain matrix, Φ_ϑ is the regressor matrix, ϑ is the vector of the unknown constant parameters to be estimated by the controller and $\hat{\vartheta}$ is the estimate of the ϑ.

The dynamics of Leonard underwater vehicle is characterised by its linearity with respect to the dynamic parameters. We exploit this property of the vehicle's dynamics and focus on designing Φ_ϑ and ϑ based on the terms (i.e. $w_{ext}(t)$ and $g^\star(\eta)$) which significantly affect the steady-state of the vehicle as follows [35] [41]:

$$\Phi_{\vartheta} = \left[\Phi_g, \;\; \Phi_{w_{ext}}\right] \quad \text{with } \Phi_g = \begin{bmatrix} \begin{matrix} s\theta \\ -c\theta s\phi \\ -c\theta c\phi \end{matrix} & 0_{3\times 1} \\ 0_{3\times 1} & \begin{matrix} -c\theta s\psi \\ -s\theta \\ 0 \end{matrix} \end{bmatrix} \quad \text{and } \Phi_{w_{ext}} = J(\eta) \tag{50}$$

and

$$\hat{\vartheta}^T = \left[\hat{\vartheta}_g^T, \;\; \hat{\vartheta}_{w_{ext}}^T\right]^T \quad \text{with } \hat{\vartheta}_g^T = [f_B \;\; z_b B]^T \text{ and } \hat{\vartheta}_{w_{ext}}^T = [w_x, \, w_y, \, w_z, \, 0, \, 0, \, 0]^T \tag{51}$$

where Φ_g and $\Phi_{w_{ext}}$ are the regressor matrices of $g^\star(\eta)$ and $w_{ext}(t)$ respectively, while $\hat{\vartheta}_g^T$ and $\hat{\vartheta}_{w_{ext}}^T$ are the estimates of the unknown dynamic parameters of $g^\star(\eta)$ and $w_{ext}(t)$ respectively. Further, $w_{ext}(t)$ is considered as a water current with irrotational components w_x, w_y and w_z in earth-fixed frame.

The structure of the proposed APD+ control scheme implemented on Leonard underwater vehicle is illustrated in Figure 5.

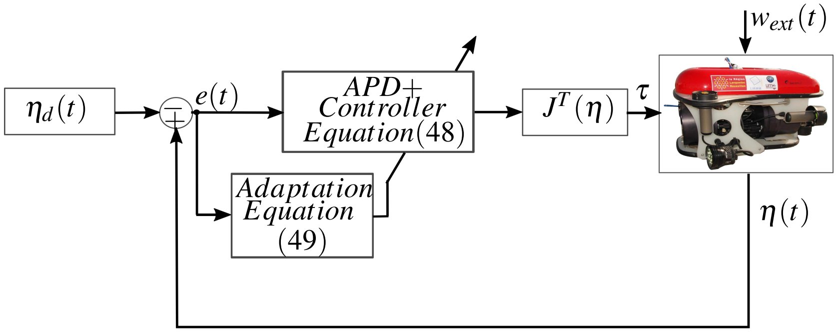

Figure 5. Block diagram of APD+ control scheme implemented on Leonard underwater vehicle.

3.3.2. Stability Analysis

For the ease of the stability analysis, we substitute the controller (48) into the vehicle's dynamics (20) and write the resulting closed-loop dynamics in a state-space form as follows:

$$\frac{d}{dt}\begin{bmatrix} e \\ \dot{e} \\ \tilde{\vartheta} \end{bmatrix} = \begin{bmatrix} \dot{e} \\ M^{\star}(\eta)^{-1}\Big[-C^{\star}(\nu,\eta)\dot{e} - D^{\star}(\nu,\eta)\dot{e} - g^{\star}(\eta) + w_{ext}(t) - \Phi_{\vartheta}\hat{\vartheta}^{T} - K_{p}e - K_{d}\dot{e}\Big] \\ -\Gamma_{\vartheta}^{-1}\Phi_{\vartheta}\Big[\alpha e + \dot{e}\Big] \end{bmatrix} \tag{52}$$

where $\tilde{\vartheta} = \vartheta - \hat{\vartheta}$.

Next, we consider the following Lyapunov candidate function:

$$V(e,\dot{e},\tilde{\vartheta}) = \frac{1}{2}\dot{e}^{T}M^{\star}(\eta)\dot{e} + \frac{1}{2}\tilde{\vartheta}^{T}\Gamma_{\vartheta}\tilde{\vartheta} + \int_{0}^{e}\alpha^{T}K_{p}d\alpha \tag{53}$$

The proposed Lyapunov candidate function in (53) is positive definite and radially unbounded since the first two terms are positive definite, and the integral term satisfies the following arguments:

$$\int_{0}^{e}\alpha^{T}K_{p}d\alpha = \int_{0}^{e_1}\alpha_{1}^{T}k_{1p}d\alpha_{1} + \int_{0}^{e_2}\alpha_{2}^{T}k_{2p}d\alpha_{2} + \ldots + \int_{0}^{e_6}\alpha_{6}^{T}k_{6p}d\alpha_{6} \tag{54}$$

$$\int_{0}^{e}\alpha^{T}K_{p}d\alpha > 0, \quad \forall \;\; e \neq 0 \in \mathbb{R}^{n} \tag{55}$$

where $K_p = diag\{k_{1p}, k_{2p}, \ldots, k_{6p}\}$.

From arguments (54) and (55) above, we can conclude that:

$$\int_{0}^{e}\alpha^{T}K_{p}d\alpha \rightarrow \infty \qquad \text{as} \;\; \|e\| \rightarrow \infty \tag{56}$$

Then, the time derivative of (53) can be written as follows:

$$\dot{V}(e,\dot{e},\tilde{\vartheta}) = \dot{e}^{T}M^{\star}(\eta)\ddot{e} + \frac{1}{2}\dot{e}^{T}\dot{M}^{\star}(\eta)\dot{e} + \tilde{\vartheta}^{T}\Gamma_{\vartheta}\dot{\tilde{\vartheta}} + e^{T}K_{p}\dot{e} \tag{57}$$

By substituting the closed-loop dynamics (52) into the time derivative of the Lyapunov candidate function (57), we deduce:

$$\begin{aligned}\dot{V}(e,\dot{e},\tilde{\vartheta}) = \dot{e}^{T}\Big[-C^{\star}(\nu,\eta)\dot{e} - D^{\star}(\nu,\eta)\dot{e} - g^{\star}(\eta) + w_{ext}(t) - \Phi_{\vartheta}\hat{\vartheta}^{T} - K_{p}e - K_{d}\dot{e}\Big] \\ + \frac{1}{2}\dot{e}^{T}\dot{M}^{\star}(\eta)\dot{e} + \tilde{\vartheta}^{T}\Gamma_{\vartheta}\dot{\tilde{\vartheta}} + e^{T}K_{p}\dot{e}\end{aligned} \tag{58}$$

Since we design the adaptation law of the controller based on the terms which affect the steady-state of the vehicle, that is, $w_{ext}(t)$ and $g^{\star}(\eta)$, then these terms can be rewritten in a regressor form as follows:

$$w_{ext}(t) - g^{\star}(\eta) = \Phi_{\vartheta}\vartheta^{T} \tag{59}$$

Then, substituting (59) into (58) leads to:

$$\dot{V}(e,\dot{e},\tilde{\vartheta}) = -\dot{e}^{T}[D^{\star}(\nu,\eta)+K_d]\dot{e} + \frac{1}{2}\dot{e}^{T}[\dot{M}^{\star}(\eta) - 2C^{\star}(\nu,\eta)]\dot{e} + \dot{e}\Phi_{\vartheta}[\vartheta^{T} - \hat{\vartheta}^{T}] - \tilde{\vartheta}^{T}\Gamma_{\vartheta}\dot{\hat{\vartheta}} \tag{60}$$

Injecting the adaptation law (49) into (60) above, leads to:

$$\dot{V}(e,\dot{e},\tilde{\vartheta}) = -\dot{e}^{T}[D^{\star}(\nu,\eta)+K_d]\dot{e} + \frac{1}{2}\dot{e}^{T}[\dot{M}^{\star}(\eta) - 2C^{\star}(\nu,\eta)]\dot{e} + \dot{e}\Phi_{\vartheta}[\vartheta^{T} - \hat{\vartheta}^{T}] - \tilde{\vartheta}^{T}\Gamma_{\vartheta}\left[\Gamma_{\vartheta}^{-1}\Phi_{\vartheta}[\alpha e + \dot{e}]\right] \tag{61}$$

Then, (61) can be rewritten as follows:

$$\dot{V}(e,\dot{e},\tilde{\vartheta}) = -\dot{e}^{T}[D^{\star}(\nu,\eta)+K_d]\dot{e} + \frac{1}{2}\dot{e}^{T}[\dot{M}^{\star}(\eta) - 2C^{\star}(\nu,\eta)]\dot{e} + \dot{e}\Phi_{\vartheta}\tilde{\vartheta}^{T} - \tilde{\vartheta}^{T}\Phi_{\vartheta}\alpha e - \tilde{\vartheta}^{T}\Phi_{\vartheta}\dot{e} \tag{62}$$

Based on Assumption 2 and the fact that $C^{\star}(\nu,\eta)$ is skew symmetric, then we can rewrite (62) as follows:

$$\dot{V}(e,\dot{e},\tilde{\vartheta}) = -\dot{e}^{T}[D^{\star}(\nu,\eta)+K_d]\dot{e} - \tilde{\vartheta}^{T}\Phi_{\vartheta}\alpha e \tag{63}$$

From (63) above, it is possible to conclude that $\dot{V}(e,\dot{e},\tilde{\vartheta})$ is negative definite; additionally, $D^{\star}(\nu,\eta) > 0$ and $K_d > 0$, and the second term on the right-hand side of $\dot{V}(e,\dot{e},\tilde{\vartheta})$, is negative. Even if, the second term on the right-hand side of $\dot{V}(e,\dot{e},\tilde{\vartheta})$ changes its sign due to the possible high degree of uncertainty on the external disturbance, then α can be designed so that the effect of this second term becomes negligible, while the first term dominates the right-hand side of $\dot{V}(e,\dot{e},\tilde{\vartheta})$. Consequently, $\dot{V}(e,\dot{e},\tilde{\vartheta})$ will remain negative definite despite the influence of these effects. Therefore, we can conclude that the origin of the resulting closed-loop dynamics is asymptotically stable.

4. Simulation Results: A Comparative Study

To compare the effectiveness and robustness of the proposed three controllers designed in the previous section, we implemented them on Leonard underwater vehicle described in section 2. During the implementation process of the control schemes, various scenarios-based numerical simulations have been conducted and the obtained results are discussed in the sequel. Before discussing the obtained results, the proposed simulation scenarios are introduced.

4.1. Proposed Numerical Simulations Scenarios

The following scenarios are tested to evaluate the effectiveness and robustness of all the proposed three control schemes on Leonard (low-inertia) underwater vehicle:

Scenario 1 (nominal case): The main objective of this scenario is to obtain the best control feedback gains, which will result in the best vehicle's desired trajectory tracking. The obtained gains are used in the remaining scenarios without any modification.

Scenario 2 (external disturbance rejection): In this scenario, we consider the presence of water current and the task of transporting an object by the vehicle from a first point and dropping it at another point as external disturbances. The ability of each controller to reject these disturbances and keeps the vehicle on the desired trajectory is evaluated. Indeed, the task of transporting the object and dropping it at a specific desired depth is illustrated in Figure 6.

Scenario 3 (robustness toward vehicle's damping and buoyancy changes): The main objective of this scenario is to evaluate the robustness of each controller towards parametric variations such as the modifications of the vehicle's buoyancy and damping.

4.2. Nominal Scenario (Results and Discussion)

In this simulation test, defined previously as Scenario 1, the vehicle is intended to follow a predefined 3D helical desired trajectory under the influence of internal disturbances such as sensor noise; external disturbances and parametric uncertainties are not considered in order to obtain the best feedback gains to be used in forthcoming scenarios. The obtained results are depicted in Figures 7-11. The three controllers are able to guide the vehicle to follow the desired 3D

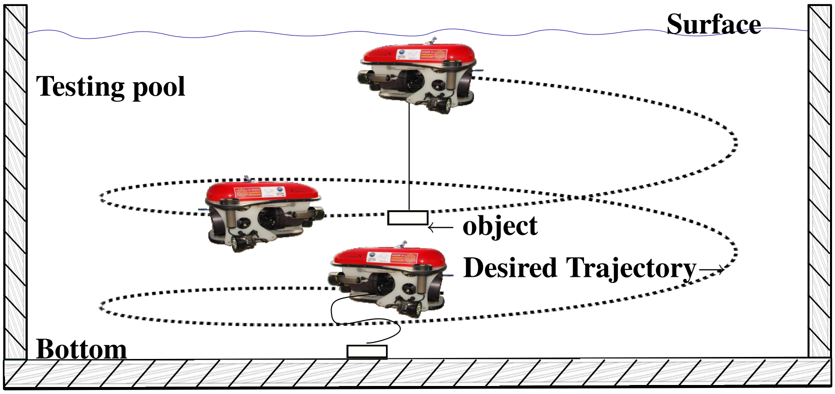

Figure 6. Illustration of Leonard underwater vehicle following a predefined desired helical trajectory while carrying an object from the surface and dropping it at the bottom of the testing pool.

helical trajectory from the initial position ($\approx 0m$) to a depth of approximately $13m$, which is near to the bottom of the testing pool. The vehicle completes this mission in $720s$ and remains stable near to the bottom of the testing pool.

Besides the complex nature of the chosen trajectory, it has a medium radius ($\approx 4m$), which helps us to evaluate the robustness and effectiveness of the proposed controllers to manoeuvre the vehicle in tracking the desired helical trajectory. The six degrees of freedom, namely, surge and roll, sway and pitch as well as heave and yaw evolution versus time are shown in Figure 7 (top plots), Figure 8 (top plots) and Figure 9 (top plots), respectively. Figure 10 shows the tracking results in 3D, which can help to visualise the motion of the vehicle in 3D easily during this test. Moreover, one can observe from Figure 7 (top left plot), Figure 8 (top left plot) and Figure 9 (top left plot) under this scenario, that all the three controllers effectively guide the vehicle to track the desired trajectory in the surge, sway and heave, respectively.

Similarly, regarding the vehicle's attitude tracking, the proposed controllers guide the vehicle to track the desired roll and pitch with slight tracking errors as shown in Figure 7 (top right plot) and Figure 8 (top right plot). However, the roll and pitch tracking errors for the PID are slightly bigger as shown in Figure 7 (top right plot) and Figure 8 (top right plot), respectively. These slightly bigger tracking errors of the proposed PID controller can also be noticed in Figures 7- 8 (middle right plot). Concerning the yaw tracking all the proposed three controllers are able to track the desired yaw as shown in Figure 9 (top right plot). Also, the tracking errors of the proposed controllers in all the six degrees of freedom of the vehicle are shown in Figures 7- 9 (middle plots). To numerically evaluate the tracking performances of the proposed controllers for position and orientation tracking of the vehicle, we use a performance index in 3D known as root mean square error (RMSE) expressed as follows:

$$RMSE(3D\ position/orientation) = \left[\frac{1}{N} \sum_{i=1}^{N} \left[e_{x/\phi}^2(i) + e_{y/\theta}^2(i) + e_{z/\psi}^2(i) \right] \right]^{\frac{1}{2}} \tag{64}$$

where N denotes the number of time-samples, while $e_{x/\phi}$, $e_{y/\theta}$ and $e_{z/\psi}$ are the tracking errors in position and orientation on x, y and z axes, respectively.

Using (64) above, we compute the RMSE for both 3D position and orientation of the controllers; the results of the computations are summarised in Table 2. Next, the control inputs evolution of all the six degrees of freedom of the vehicle for all the controllers are depicted in Figures 7- 9 (bottom plots). Then, from the control signals evolution versus time obtained results, we numerically estimate the energy consumption of each controller using the following index:

$$INT = \int_{t_1}^{t_2} \left\| \tau(t) \right\| dt \tag{65}$$

where INT defines the integral of control signals, $t_1 = 1s$ and $t_2 = 720s$
Since CT and APD+ controllers show superior tracking performances, confirmed by Table 2, we investigate their energy consumption using (65) as follows:

$$\frac{INT_{3D\ position\ APD+}}{INT_{3D\ position\ CT}} = \frac{7130}{7005} = 1.02. \quad \frac{INT_{3D\ orientation\ APD+}}{INT_{3D\ orientation\ CT}} = \frac{209}{214} = 0.98. \tag{66}$$

From (66) it is worth to note, that besides the superior tracking performance of the APD+ controller, its energy consumption is approximately the same as the CT controller in both desired position and orientation trackings. Hence, we can conclude that the APD+ controller demonstrates superior tracking performance than both the CT and PID controllers in this scenario. The uncertain parametric estimations made by the APD+ controller are shown in Figure 11.

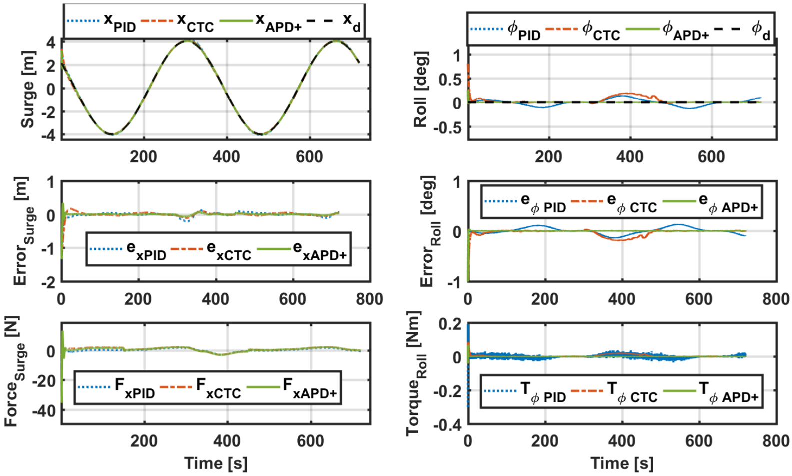

Figure 7. Trackings performances comparison of the APD+, CT and PID controllers implemented on Leonard underwater vehicle in the nominal scenario: (upper plots) surge and roll trackings, (middle plots) surge and roll corresponding tracking errors and (bottom plots) are the evolution of the vehicle's control inputs.

Table 2. Summary of the controllers performance indices

	SCENARIO	PID	CT	APD+
RMSE 3D-position	Nominal Scenario	0.1133	0.0944	0.0618
$[m]$	Combined Scenarios	0.5421	0.1382	0.1462
RMSE 3D-orientation	Nominal Scenario	2.4026	0.9038	0.4618
$[deg]$	Combined Scenarios	2.4108	0.9176	0.4629

4.3. Combined Scenarios (Results and Discussion)

To investigate the robustness of each controller in this test, we propose to combine all the scenarios defined previously in one test, and the vehicle follows the same desired trajectory as in the nominal case. The effect of sensor measurement noise can be noticed in Figures 12- 14 (bottom plots), but more amplified on the roll and pitch control signals of the proposed APD+ controller. When the vehicle reaches a depth of $2.5m$, the influence of a 3-kg object tied at the bottom of the vehicle becomes active as illustrated in Figure 6. We can notice the effect

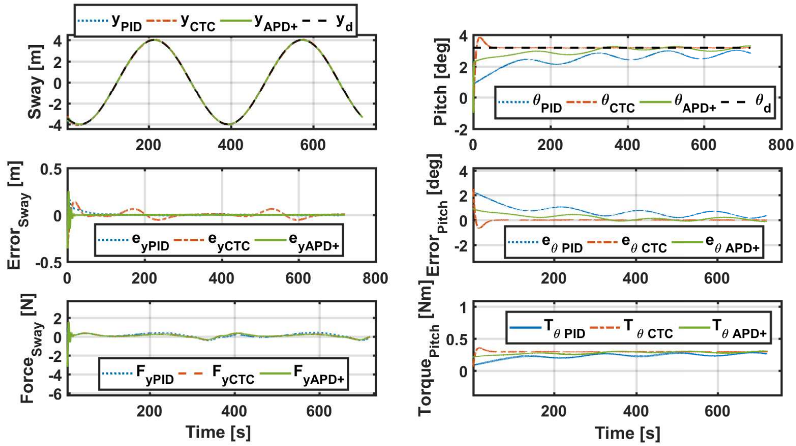

Figure 8. Trackings performances comparison of the APD+, CT and PID controllers implemented on Leonard underwater vehicle in the nominal scenario: (upper plots) sway and pitch trackings, (middle plots) sway and pitch corresponding tracking errors and (bottom plots) are the evolution of the vehicle's sway and pitch control inputs.

of this added mass as a sudden change in the overall mass of the vehicle at $150s$ along the heave axis as shown in Figure 14 (heave plots), which can also be visualised in 3D as shown in Figure 15; however, the vehicle yaw tracking is less affected as illustrated in Figure 14 (yaw plots).

In the case of CT controller, the vehicle deviates slightly from the desired trajectory, while the APD+ controller compensates for the effect within a short time (about $4s$) and keeps the vehicle around the desired trajectory; on the other hand, it takes the PID controller about $40s$ to compensate for the same effect. Also, the vehicle tracking is less affected on the surge, roll and sway in the case of the proposed APD+ controller as compared to the remaining controllers, as shown in Figure 12 and Figure 13 (sway plots). However, the pitch tracking of the proposed APD+ controller is slightly affected as shown in Figure 12 (pitch plots).

When the vehicle reaches $5m$ of depth, its damping and floatability are modified by $+90\%$ and $+200\%$, respectively, to evaluate the robustness of the three controllers towards parametric variations of the vehicle, which is clearly seen in Figure 12 (roll plots), Figure 13 (pitch plots) and Fig 14 (heave plots) at about

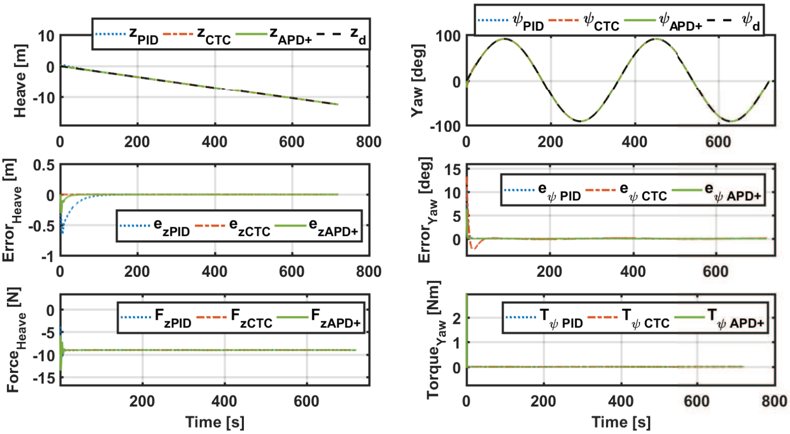

Figure 9. Trackings performances comparison of the APD+, CT and PID controllers implemented on Leonard underwater vehicle in the nominal scenario: (upper plots) heave and yaw trackings, (middle plots) heave and yaw corresponding tracking errors and (bottom plots) are the evolution of the vehicle's heave and yaw control inputs.

$300s$. Concerning the roll and pitch tracking APD+ controller compensates for this effect and keeps the vehicle very close to the desired trajectory. At the same time, it takes the CT controller about $20s$ and $15s$ to compensate for the same effect on roll and pitch, respectively. However, the PID controller oscillates slightly around the desired roll, while tracking the desired pitch with a slightly bigger tracking error.

As the vehicle reaches the depth of about $7.5m$ in $450s$, the 3-kg object tied to the vehicle touches the floor of the testing pool at the same time the vehicle's damping and floatability are rechanged to their nominal values. These effects are clearly observed in Figure 12 (roll plots), Figure 13 (pitch plots) and Figure 14 (heave plots), as well as in Figure 15 (3D plot), while all the three controllers maintain approximately their superior performances in the surge, sway and yaw trackings as shown in Figure 12 (surge plots), Figure 13 (sway plots) and Figure 14 (yaw plots), respectively.

To further evaluate the ability of the controllers to reject external disturbances, when the vehicle goes to $10m$ depth, we apply a water current moving at a speed of $0.35m/s$ to disturb the vehicle. Even though, the controllers reject

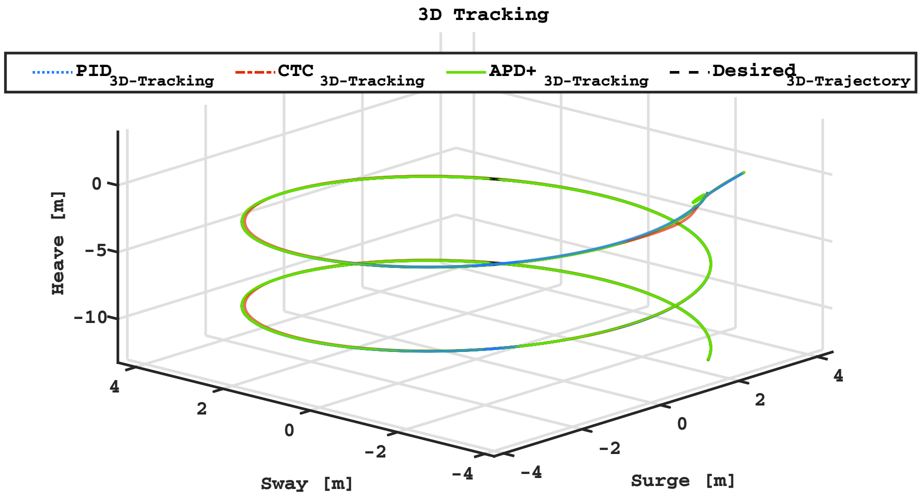

Figure 10. Three-dimensional (3D) helical trajectory trackings performances comparison of the APD+, CT and PID controllers implemented on Leonard underwater vehicle in the nominal scenario.

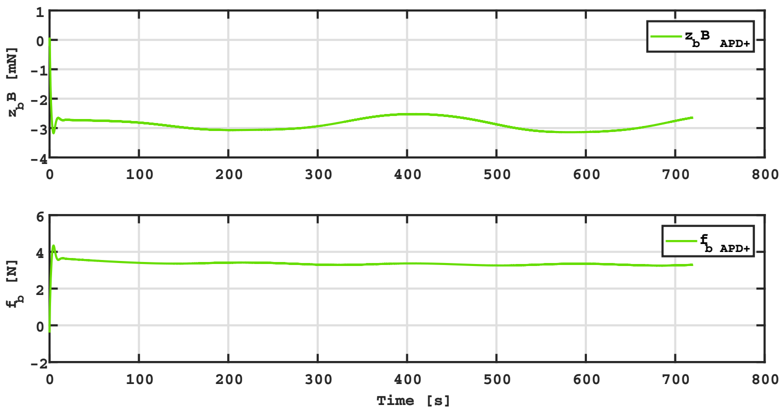

Figure 11. Parametric estimations of the APD+ controller implemented on Leonard underwater vehicle for three-dimensional (3D) helical trajectory tracking in the nominal scenario.

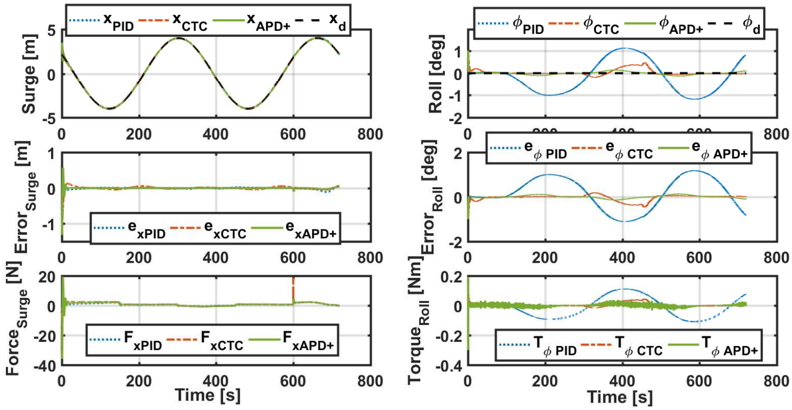

Figure 12. Trackings performances comparison of the APD+, CT and PID controllers implemented on Leonard underwater vehicle in the combined scenario: (upper plots) surge and roll trackings, (middle plots) surge and roll corresponding tracking errors and (bottom plots) the evolution of the vehicle's control inputs.

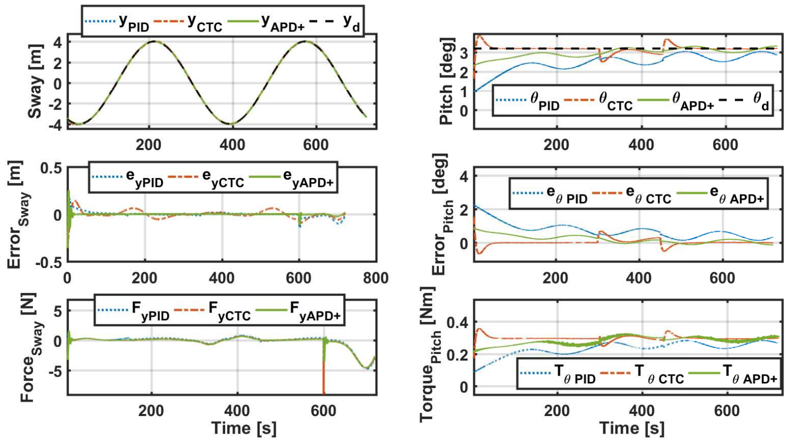

Figure 13. Trackings performances comparison of the APD+, CT and PID controllers implemented on Leonard underwater vehicle in the combined scenario: (upper plots) sway and pitch trackings, (middle plots) sway and pitch corresponding tracking errors and (bottom plots) are the evolution of the vehicle's sway and pitch control inputs.

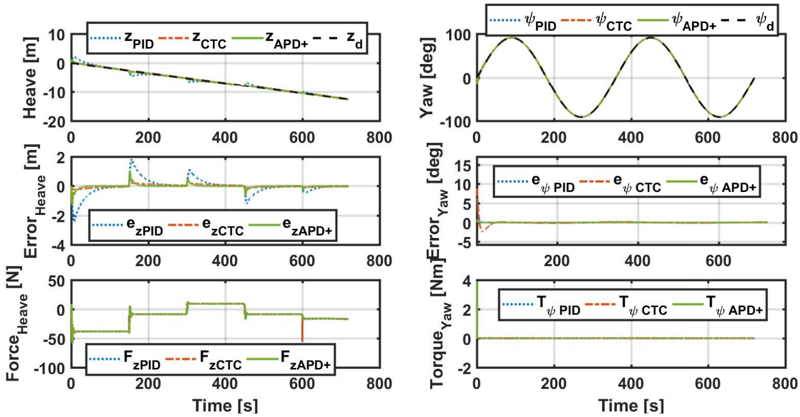

Figure 14. Trackings performances comparison of the APD+, CT and PID controllers implemented on Leonard underwater vehicle in the combined scenario: (upper plots) heave and yaw trackings, (middle plots) heave and yaw corresponding tracking errors and (bottom plots) are the evolution of the vehicle's heave and yaw control inputs.

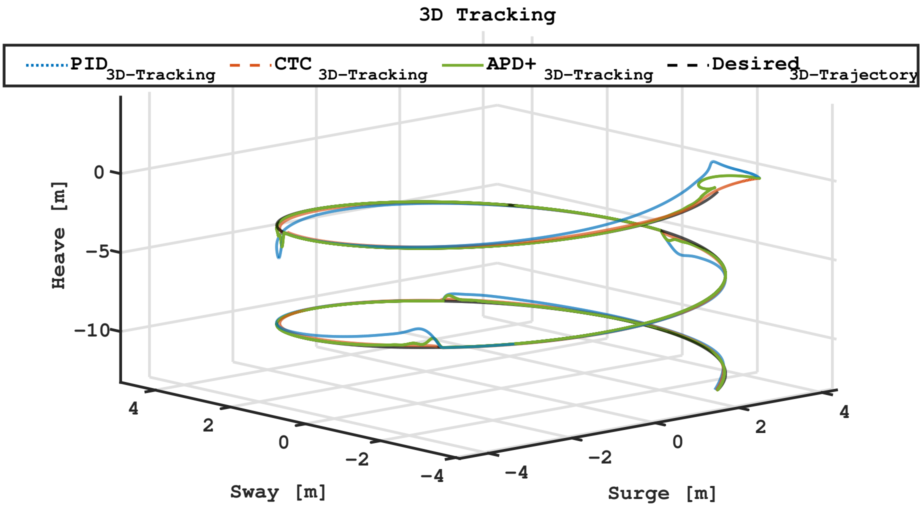

Figure 15. Three-dimensional (3D) helical trajectory trackings performances comparison of the APD+, CT and PID controllers implemented on Leonard underwater vehicle in the combined scenario.

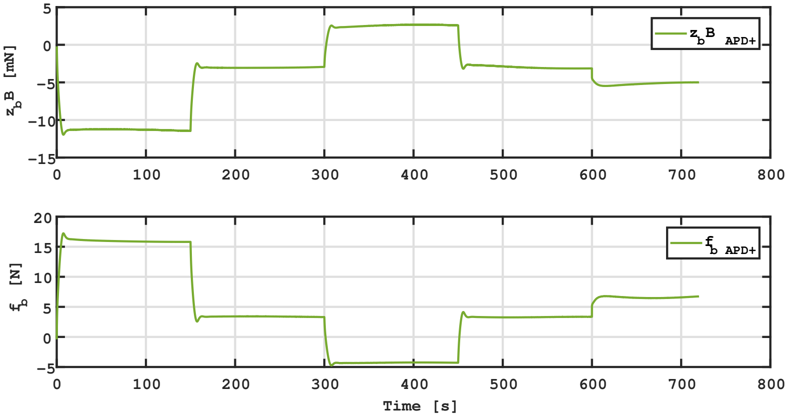

Figure 16. Parametric estimations of the APD+ controller implemented on Leonard underwater vehicle for three-dimensional (3D) helical trajectory tracking in the combined scenario.

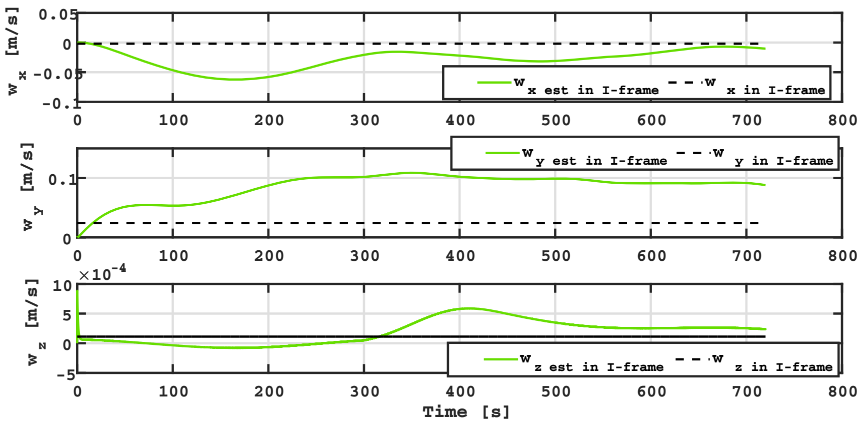

Figure 17. External disturbance estimation of the APD+ controller implemented on Leonard underwater vehicle for three-dimensional (3D) helical trajectory tracking in the combined scenario.

the applied external disturbance, but as a consequence the controllers consume a slightly higher amount of energy especially the CT controller as shown in Figure 12 ($Force_{surge}$ plot), Figure 13 ($Force_{sway}$ plot) and Figure 14 ($Force_{heave}$ plot).

Finally, it is possible to conclude that the APD+ controller demonstrates superior performance as compared to the CT and PID controllers. Even though the CT controller shows similar performance to APD+ controller in terms of 3D position tracking confirmed by numerical computation of RMSE given in Table 2, the RMSE of APD+ controller for 3D orientation tracking from Table 2 is 50% less than the RMSE of CT controller. Moreover, the energy consumption estimation using (65) shows that approximately the same energy is consumed by both APD+ and CT controllers during 3D position trackings. However, concerning 3D orientation tracking, the APD+ controller consumes 3.5% less energy than the CT controller. The uncertain parametric and external disturbance estimations by the APD+ controller are depicted in Figures 16 and 17, respectively. From these figures representing uncertain parametric and external disturbance estimations by the APD+ controller, one can observe the influence of all the effects introduced during this simulation scenario.

CONCLUSION AND FUTURE WORK

In this chapter, the performances of the non-model-based (PID), the model-based (CT) as well as the adaptive model-based (APD+) controllers have been investigated for three-dimensional (3D) helical trajectory tracking of a low-inertia underwater vehicle. The resulting closed-loop dynamics stability analysis of all the three proposed controllers have been conducted based on Lyapunov direct method. The controllers have then been implemented on Leonard underwater vehicle for 3D helical trajectory tracking. Scenarios-based simulation results demonstrate the superior performance of APD+ controller, compared to the two other controllers, for marine applications under the influences of parametric variations, internal vehicle's perturbations and external disturbances. In the near future, we will focus on implementing these control schemes in real-time on low-inertia underwater vehicles. Also, we may integrate observers in real-time to all the controllers for velocity estimation, since the majority of the low-inertia and low-cost underwater vehicles are not equipped with DVL (Doppler velocity logger) sensors.

Acknowledgments

The authors thanks the Petroleum Technology Development Fund (PTDF), Nigeria, for supporting the first author financially through PhD scholarship.

References

[1] Bao J., Li D., Qiao X., and Rauschenbach T., "Integrated navigation for autonomous underwater vehicles in aquaculture: A review," *Information Processing in Agriculture*, 2019.

[2] Maalouf D., Chemori A., and Creuze V., "L1 adaptive depth and pitch control of an underwater vehicle with real-time experiments," *Ocean Engineering*, vol. 98, pp. 66–77, 2015.

[3] Sahoo A., Dwivedy S. K., and Robi P., "Advancements in the field of autonomous underwater vehicle," *Ocean Engineering*, vol. 181, pp. 145–160, 2019.

[4] Chaturvedi S. K., Basu S., and Banerjee S., "Gimbal orientation study and assessment for the autonomous underwater vehicles," *Journal of Ocean Engineering and Science*, 2019.

[5] Antonelli G., *Underwater robots*, vol. 123. Springer, 2018.

[6] Kong F., Guo Y., and Lyu W., "Dynamics modeling and motion control of an new unmanned underwater vehicle," *IEEE Access*, vol. 8, pp. 30119–30126, 2020.

[7] Havenstrøm S. T., Rasheed A., and San O., "Deep reinforcement learning controller for 3d path-following and collision avoidance by autonomous underwater vehicles," 2020.

[8] Guerrero J., Torres J., Creuze V., and Chemori A., "Adaptive disturbance observer for trajectory tracking control of underwater vehicles," *Ocean Engineering*, vol. 200, p. 107080, 2020.

[9] Hammad M. M., Elshenawy A. K., and El Singaby M., "Position control and stabilization of fully actuated auv using pid controller," in *Proceedings of SAI Intelligent Systems Conference*, pp. 517–536, Springer, 2016.

[10] Watson S. A. and Green P. N., "Depth control for micro-autonomous underwater vehicles (μauvs): Simulation and experimentation," *International Journal of Advanced Robotic Systems*, vol. 11, no. 3, p. 31, 2014.

[11] Kuo C.-L., Tsui C.-K., Pai N.-S., Lin C.-H., Chen S.-C., and Li P.-W., "A pid controller for the underwater robot station-keeping," in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, pp. 1242–1246, IEEE, 2016.

[12] Upadhyay V., Gupta S., Dubey A., Rao M., Siddhartha P., Gupta V., George S., R. Bobba, R. Sirikonda, A. Maloo, *et al.*, "Design and motion control of autonomous underwater vehicle, amogh," in *2015 IEEE Underwater Technology (UT)*, pp. 1–9, IEEE, 2015.

[13] Geder J. D., Palmisano J., Ramamurti R., Sandberg W. C., and Ratna B., "Fuzzy logic pid based control design and performance for a pectoral fin propelled unmanned underwater vehicle," in *2008 International Conference on Control, Automation and Systems*, pp. 40–46, IEEE, 2008.

[14] Hu B., Tian H., Qian J., Xie G., Mo L., and Zhang S., "A fuzzy-pid method to improve the depth control of auv," in *2013 IEEE International Conference on Mechatronics and Automation*, pp. 1528–1533, IEEE, 2013.

[15] Chen Q., Chen T., and Zhang Y., "Research of ga-based pid for auv motion control," in *2009 International Conference on Mechatronics and Automation*, pp. 4446–4451, IEEE, 2009.

[16] Guerrero J., Torres J., Creuze V., Chemori A., and Campos E., "Saturation based nonlinear pid control for underwater vehicles: Design, stability analysis and experiments," *Mechatronics*, vol. 61, pp. 96–105, 2019.

[17] Hernández-Alvarado R., García-Valdovinos L. G., Salgado-Jiménez T., Gómez-Espinosa A., and Fonseca-Navarro F., "Neural network-based self-tuning pid control for underwater vehicles," *Sensors*, vol. 16, no. 9, p. 1429, 2016.

[18] Makavita C., Nguyen H., and Ranmuthugala D., "Fuzzy gain scheduling based optimally tuned pid controllers for an unmanned underwater vehicle," *International journal of conceptions on electronics and communication engineering*, vol. 2, no. 1, pp. 7–13, 2014.

[19] Zhang L., Liu L., Zhang S., and Cao S., "Saturation based nonlinear fopd motion control algorithm design for autonomous underwater vehicle," *Applied Sciences*, vol. 9, no. 22, p. 4958, 2019.

[20] Campos E., Monroy J., Abundis H., Chemori A., Creuze V., and Torres J., "A nonlinear controller based on saturation functions with variable parameters to stabilize an auv," *International Journal of Naval Architecture and Ocean Engineering*, vol. 11, no. 1, pp. 211–224, 2019.

[21] Fischer N., Bhasin S., and Dixon W., "Nonlinear control of an autonomous underwater vehicle: A rise-based approach," in *Proceedings of the 2011 American Control Conference*, pp. 3972–3977, IEEE, 2011.

[22] Xu J., Wang M., and Qiao L., "Dynamical sliding mode control for the trajectory tracking of underactuated unmanned underwater vehicles," *Ocean engineering*, vol. 105, pp. 54–63, 2015.

[23] Martin S. C. and Whitcomb L. L., "Nonlinear model-based tracking control of underwater vehicles with three degree-of-freedom fully coupled dynamical plant models: Theory and experimental evaluation," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 2, pp. 404–414, 2017.

[24] Martin S. C. and Whitcomb L. L., "Fully actuated model-based control with six-degree-of-freedom coupled dynamical plant models for underwater vehicles: Theory and experimental evaluation," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1164–1184, 2016.

[25] Xu J. and Wang N., "Optimization of rov control based on genetic algorithm," in *2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO)*, pp. 1–4, IEEE, 2018.

[26] Chen J., Zhu H., Zhang L., and Sun Y., "Research on fuzzy control of path tracking for underwater vehicle based on genetic algorithm optimization," *Ocean Engineering*, vol. 156, pp. 217–223, 2018.

[27] Al Makdah A. A. R., Daher N., Asmar D., and Shammas E., "Three-dimensional trajectory tracking of a hybrid autonomous underwater vehicle in the presence of underwater current," *Ocean Engineering*, vol. 185, pp. 115–132, 2019.

[28] Liu X., Zhang M., and Wang S., "Adaptive region tracking control with prescribed transient performance for autonomous underwater vehicle with thruster fault," *Ocean Engineering*, vol. 196, p. 106804, 2020.

[29] Guerrero J., Torres J., Creuze V., and Chemori A., "Trajectory tracking for autonomous underwater vehicle: An adaptive approach," *Ocean Engineering*, vol. 172, pp. 511–522, 2019.

[30] Liao Y., Du T., and Jiang Q., "Model-free adaptive control method with variable forgetting factor for unmanned surface vehicle control," *Applied Ocean Research*, vol. 93, p. 101945, 2019.

[31] Zhang Z. and Wu Y., "Adaptive fuzzy tracking control of autonomous underwater vehicles with output constraints," *IEEE Transactions on Fuzzy Systems*, 2020.

[32] Lu Y., Zhang G., Qiao L., and Zhang W., "Adaptive output-feedback formation control for underactuated surface vessels," *International Journal of Control*, vol. 93, no. 3, pp. 400–409, 2020.

[33] Mohan S. and Kim J., "Indirect adaptive control of an autonomous underwater vehicle-manipulator system for underwater manipulation tasks," *Ocean Engineering*, vol. 54, pp. 233–243, 2012.

[34] Campos E., Chemori A., Creuze V., Torres J., and Lozano R., "Saturation based nonlinear depth and yaw control of underwater vehicles with stability analysis and real-time experiments," *Mechatronics*, vol. 45, pp. 49–59, 2017.

[35] Fossen T. I., "Guidance, navigation, and control of ships, rigs and underwater vehicles," *Trondheim, Norway: Marine Cybernetics*, 2002.

[36] Saied H., Chemori A., El Rafei M., Francis C., and Pierrot F., "From non-model-based to model-based control of pkms: a comparative study," in *Mechanism, Machine, Robotics and Mechatronics Sciences*, pp. 153–169, Springer, 2019.

[37] Londhe P., Mohan S., Patre B., and Waghmare L., "Robust task-space control of an autonomous underwater vehicle-manipulator system by pid-like fuzzy control scheme with disturbance estimator," *Ocean Engineering*, vol. 139, pp. 1–13, 2017.

[38] Kim M., Joe H., Pyo J., Kim J., Kim H., and Yu S.-c., "Variable-structure pid controller with anti-windup for autonomous underwater vehicle," in *2013 OCEANS-San Diego*, pp. 1–5, IEEE, 2013.

[39] Wu H., Su W., and Liu Z., "Pid controllers: Design and tuning methods," in *2014 9th IEEE Conference on Industrial Electronics and Applications*, pp. 808–813, IEEE, 2014.

[40] Kelly R., Davila V. S., and Perez J. A. L., *Control of robot manipulators in joint space*. Springer Science & Business Media, 2006.

[41] Antonelli G., Chiaverini S., Sarkar N., and West M., "Adaptive control of an autonomous underwater vehicle: experimental results on odin," *IEEE Transactions on Control Systems Technology*, vol. 9, no. 5, pp. 756–765, 2001.

In: Underwater Vehicles
Editor: George M. Roman
ISBN: 978-1-53618-876-9

Chapter 3

CONTROLLERS TO AVOID COLLISION WITH 3D OBSTACLES USING SENSORS

Jonghoek Kim*
Electronic and Electrical Convergence Department,
Hongik University, Sejong-City, South Korea

Abstract

Collision avoidance in 3D environments is important to the problem of planning safe trajectories for an autonomous vehicle. Existing literature on collision avoidance assumed that obstacle shapes are known a priori and modeled obstacles as spheres or bounding boxes. However, in 3D environments, an obstacle shape is unknown to the autonomous vehicle, and the vehicle detects an obstacle boundary using 3D sensors, such as 3D sonar. In this chapter, we introduce control laws for collision avoidance, considering scenarios where a vehicle detects arbitrarily shaped and non-convex obstacles using sensors. Moreover, our control laws are designed considering motion constraints, such as the maximum turn rate and the maximum speed rate of the vehicle. The effectiveness of our control laws is verified using MATLAB simulations.

Keywords: collision avoidance, 3D environment, maximum turn rate, maximum acceleration, arbitrarily shaped obstacle

*Corresponding Author's Email: jonghoek@hongik.ac.kr.

1. INTRODUCTION

In recent years, significant advancements have been made in the capabilities of Unmanned Vehicles. Collision avoidance is important to the problem of planning safe trajectories for these vehicles.

Many papers tackled collision avoidance in 2D environments [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. The Velocity Obstacle (VO) approach has been adopted to avoid moving obstacles [3, 4, 7]. In the VO approach, it is assumed that an obstacle maintains its velocity at all future times. The VO approach accounts for collision checks at all future times (under the linear velocity assumption) and is very fast to compute and is suitable for robotic applications, where the algorithm is implemented on embedded systems that have limited computational resources and hard real-time requirements [2]. [11] considered collision avoidance laws for objects with arbitrary shapes. However, [11] is restricted to an arbitrary obstacle in two dimensions.

Several authors addressed 3D collision avoidance [12, 13, 15, 22, 23]. Existing literature on collision avoidance assumed that obstacle shapes are known a priori and modeled obstacles as spheres or bounding boxes [12, 13, 14, 15, 16]. However, in 3D environments, a vehicle detects an obstacle boundary using 3D sensors, such as 3D sonar. Thus, the vehicle can only access 3D points on an obstacle boundary, not the entire obstacle [17, 18, 19, 20, 21]. Our control laws are developed to avoid collision with each of these points. As far as we know, the collision avoidance controllers presented in this chapter are novel in considering the fact that an obstacle shape is unknown a priori and that the vehicle detects an obstacle boundary using sensors. Since we do not use shape approximations, our approach is suitable for avoiding an arbitrary shaped and non-convex obstacles.

A vehicle cannot turn or accelerate with infinite acceleration due to hardware limits. Considering the dynamic constraints of an UAV, [15, 22, 23] presented how to plan a global path for the UAV so that it reaches a goal while avoiding collision. However, [15, 22, 23] did not consider a scenario where a vehicle detects an obstacle boundary using on-board sensors. Note that a moving obstacle can change its velocity while it moves. However, the control laws in [22, 23] did not use the velocity information of a moving obstacle, thus may not be suitable for avoiding a fast moving obstacle.

[15] is based on the dynamic rapidly-exploring random tree(RRT) algorithm, which expands the vehicle path iteratively until a path to the goal is found.

This approach assumes that obstacle environments are known to the vehicle a priori. However, this may not be feasible in practice. Also, there may be a case where the goal is too far from the vehicle position in 3D environments. In this case, it may not be feasible to generate a 3D path to the goal in real time. Thus, [15] may not be suitable for a case where the vehicle detects an obstacle boundary abruptly and the vehicle must re-plan to avoid the obstacle.

This chapter assumes that an obstacle maintains its velocity for a certain amount of time (U time steps) in the future. Under this assumption, we provide control laws to achieve collision avoidance within U time steps in the future. Since we update control laws at each time step, we achieve collision avoidance at each time step. In the case where the vehicle does not detect an obstacle boundary, the vehicle just moves toward its goal.

In our control laws, we generate a safe velocity of the vehicle so that a collision is avoided using the generated velocity. Our control laws use the velocity information of a moving obstacle, thus are suitable for avoiding a fast moving obstacle. Our control laws are designed considering motion constraints, such as the maximum turn rate and the maximum speed rate of the vehicle. Moreover, our control laws are designed to generate a safe velocity as fast as possible while minimizing the velocity change. As far as we know, the proposed collision avoidance controllers are novel in considering the fact that an obstacle shape is unknown a priori and that the vehicle detects an obstacle boundary using sensors. The effectiveness of our control laws is verified using MATLAB simulations.

The chapter is organized as follows: Section 2 introduces the preliminary information of this chapter. Section 3 discusses definitions and assumptions related to this chapter. Section 4 introduces our control laws for collision avoidance. Section 5 introduces MATLAB simulations to verify our method. Section 5.2 provides conclusions.

2. PRELIMINARY INFORMATION

2.1. Model

This chapter uses two reference frames: an inertial reference frame $\{I\}$ and a body-fixed reference frame $\{B\}$ [24]. We introduce several definitions which are used in rigid-body dynamics [24].

The origin of $\{I\}$ is an appropriate location with the axes pointing North, East, and Down respectively. The $\{B\}$ frame is fixed to the vehicle and acts as the moving frame. The origin of $\{B\}$ frame is at the center of gravity of the vehicle.

ϕ, θ, ψ denote *euler roll angle*, *euler pitch angle*, and *euler yaw angle* in the inertial coordinate system. ϕ, θ, ψ are used to place a 3-D body in any orientation.

Let $c(\eta)$ denote $\cos(\eta)$. Also, let $s(\eta)$ denote $\sin(\eta)$. Let $t(\eta)$ denote $\tan(\eta)$.

The counterclockwise rotation of ψ about the z-axis in the inertial coordinate system is represented by the following rotation matrix $\mathbf{R}(\psi)$.

$$\mathbf{R}(\psi) = \begin{pmatrix} c(\psi) & -s(\psi) & 0 \\ s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{1}$$

The counterclockwise rotation of θ about the y-axis in the inertial coordinate system is represented by the following rotation matrix $\mathbf{R}(\theta)$.

$$\mathbf{R}(\theta) = \begin{pmatrix} c(\theta) & 0 & s(\theta) \\ 0 & 1 & 0 \\ -s(\theta) & 0 & c(\theta) \end{pmatrix}. \tag{2}$$

The counterclockwise rotation of ϕ about the x-axis in the inertial coordinate system is represented by the following rotation matrix $\mathbf{R}(\phi)$.

$$\mathbf{R}(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c(\phi) & -s(\phi) \\ 0 & s(\phi) & c(\phi) \end{pmatrix}. \tag{3}$$

A single rotation matrix can be formed by multiplying the yaw, pitch, and roll rotation matrices to obtain

$$\mathbf{R}(\psi, \theta, \phi) = \mathbf{R}(\psi)\mathbf{R}(\theta)\mathbf{R}(\phi). \tag{4}$$

3. Definitions and Assumptions

Let $\mathbf{q} \in R^3$ denote the position of a vehicle. Let $\mathbf{v} = \dot{\mathbf{q}}$ denote the velocity of the vehicle. Let $\mathbf{h} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$ denote the *heading vector* of the vehicle.

$\eta(k)$ is used to indicate η at time step k. For instance, a vehicle at time step k is located at $\mathbf{q}(k) \in R^3$.

The following equation shows the motion model of the vehicle in discrete-time systems.

$$\mathbf{q}(k+1) = \mathbf{q}(k) + \mathbf{v}(k) * dt \tag{5}$$

Here, dt denotes the sampling interval of our discrete-time system.

Considering the hardware information of the vehicle, there are bounds for $\mathbf{v}(k)$ as follows. The maximum speed of the vehicle is s_{max}. This implies that $\|\mathbf{v}(k)\| \leq s_{max}$. The maximum speed rate of the vehicle is a_{max}, which implies that

$$-a_{max} \leq \frac{(\|\mathbf{v}(k+1)\| - \|\mathbf{v}(k)\|)}{dt} \leq a_{max}. \tag{6}$$

The maximum turn rate of the vehicle is α. This implies that

$$A(\mathbf{v}(k), \mathbf{v}(k+1)) < \alpha * dt \tag{7}$$

for every time step k. Here, $A(\mathbf{v}_1, \mathbf{v}_2) = \arccos(\frac{\mathbf{v}_1 \cdot \mathbf{v}_2}{\|\mathbf{v}_1\|\|\mathbf{v}_2\|})$ denotes the angle formed by two vectors $\mathbf{v}_1$ and $\mathbf{v}_2$.

Recall that the vehicle can access 3D points on an obstacle boundary using sensors. Each point is called the *obstacle point*, say $\mathbf{O}$. We assume that a vehicle can estimate the position and velocity of an obstacle point. This assumption is commonly used in collision avoidance algorithms based on VO approaches [7, 3, 4].[1]

We say that the vehicle and an obstacle point *collide* in the case where the relative distance between the vehicle and the obstacle point is less than r. To avoid collision with an obstacle, it is assumed that the sensing range of a vehicle is bigger than r. Let *obstacle sphere* denote the sphere centered at an obstacle point $\mathbf{O}$ with radius r.

r is used to compensate for the inaccuracy of sensor measurements. In other words, r is determined by measurement error of 3D sensors on the vehicle. Many sensors (LIDAR, RADAR or any time-of-flight or vision sensors) are

[1]The velocity of an obstacle point is that of an obstacle which has the obstacle point on its boundary. We can track an obstacle point with features to estimate the velocity of the associated obstacle. [17, 21, 25] used 3D sensor, such as lidar, to estimate the position and velocity of an obstacle. The approach in [17, 21, 25] can be used to estimate the velocity of an obstacle. However, estimating the velocity of an obstacle is not within the scope of this chapter.

bound to measurement error. As we consider a 3D sensor with large measurement error, we need to set r as a large value. However, considering a 3D sensor with small measurement error, we can set r as a small value.

4. Control Laws for Collision Avoidance

4.1. Collision Prediction

We introduce how to predict collision between the vehicle $\mathbf{q}$ and an obstacle point, say $\mathbf{O}$, within U time steps in the future.

The prediction window size U is related to the maneuverability of the vehicle. In the case where α is small, the vehicle cannot turn abruptly. This case, the vehicle must begin collision avoidance maneuver from an instant when the vehicle is far from an obstacle. Thus, we need to set U as a large value for safe collision avoidance. However, in the case where α is large, the vehicle can turn abruptly. This case, we can set U as a small value for safe collision avoidance.

Let k denote the current time step. Let $\mathbf{A}(k)$ denote $\mathbf{A}$ at time step k. We consider the case where $\|\mathbf{q}(k)-\mathbf{O}(k)\| > r$. We assume that the vehicle moves with a constant velocity $\mathbf{v}$ within U time steps in the future. Also, we assume that $\mathbf{O}$ moves with a constant velocity $\mathbf{v}_O$ within U time steps. Since $\|\mathbf{q}(k) - \mathbf{O}(k)\| > r$, $\mathbf{q}(k)$ is outside the obstacle sphere centered at $\mathbf{O}_k$.

Since both $\mathbf{q}$ and $\mathbf{O}$ move with constant speeds, a collision does not occur between two time steps k and $k+U$ in the case where

$$\|\mathbf{q}(k) - \mathbf{O}(k) + (\mathbf{v} - \mathbf{v}_O)u * dt\| > r. \tag{8}$$

for all u between 0 and U. As we vary u in (8) from 0 to U, $\mathbf{q}(k)+(\mathbf{v}-\mathbf{v}_O)u*dt$ forms the line $L(\mathbf{q}(k), \mathbf{q}(k) + (\mathbf{v}-\mathbf{v}_O)U * dt)$. Here, $L(\mathbf{A}, \mathbf{B})$ indicates the line segment connecting two points $\mathbf{A}$ and $\mathbf{B}$.

Using (8), we derive the following lemma, which achieves collision avoidance between two time steps k and $k+U$.

Lemma 1. *An obstacle point $\boldsymbol{O}$ moves with a constant velocity $\boldsymbol{v}_O$ between two time steps k and $k+U$. Also, the vehicle $\boldsymbol{q}$ moves with a constant velocity $\boldsymbol{v}$ between two time steps k and $k+U$. Suppose that $\boldsymbol{q}(k)$ is outside the obstacle sphere centered at $\boldsymbol{O}(k)$. The vehicle avoids colliding with the obstacle point between two time steps k and $k+U$ in the case where $L(\boldsymbol{q}(k), \boldsymbol{q}(k) + (\boldsymbol{v} - \boldsymbol{v}_O)U * dt)$ does not meet the obstacle sphere.*

We apply Lemma 1 to provide a condition for collision avoidance within U time steps in the future. We assume that $\mathbf{v}_O$ does not change within U time steps. Under this assumption, $\mathbf{q}$ can check whether it will collide within U time steps. $\mathbf{q}$ checks if $\mathbf{v}$, which does not consider collision avoidance, satisfies the following *basic collision avoidance condition* within U time steps: $L(\boldsymbol{q}(k), \boldsymbol{q}(k) + (\boldsymbol{v} - \boldsymbol{v}_O)U * dt)$ *does not meet the obstacle sphere centered at* $\boldsymbol{O}(k)$.

In the case where $L(\mathbf{q}(k), \mathbf{q}(k)+(\mathbf{v}-\mathbf{v}_O)U*dt)$ does not meet the obstacle sphere, the vehicle uses $\mathbf{v}$ as its velocity command within one time step, since collision is avoided using $\mathbf{v}$. In the case where $L(\mathbf{q}(k), \mathbf{q}(k) + (\mathbf{v} - \mathbf{v}_O)U * dt)$ meets the obstacle sphere, the vehicle searches for a feasible velocity to avoid collision, using the method in the next subsection.

Next, Lemma 2 presents the collision avoidance condition analytically, using the basic collision avoidance condition.

Lemma 2. *Let* $u^c = \frac{(\boldsymbol{v}(k)-\boldsymbol{v}_O(k))\cdot(\boldsymbol{O}(k)-\boldsymbol{q}(k))}{dt\|\boldsymbol{v}(k)-\boldsymbol{v}_O(k)\|^2}$. *If* u^c *exists between 0 and* U *while satisfying that* $\boldsymbol{v}(k) \neq \boldsymbol{v}_O(k)$, *then the vehicle avoids collision within* U *steps as long as the distance between* $\boldsymbol{q}(k) + u^c dt(\boldsymbol{v}(k) - \boldsymbol{v}_O(k))$ *and* $\boldsymbol{O}(k)$ *is bigger than* r. *Otherwise, the vehicle avoids collision within* U *steps as long as* $\|\boldsymbol{q}(k) + (\boldsymbol{v} - \boldsymbol{v}_O)U * dt - \boldsymbol{O}(k)\| > r$.

Proof. Let d_{min} denote the minimum distance between $\mathbf{O}(k)$ and a point on $L(\mathbf{q}(k), \mathbf{q}(k) + (\mathbf{v} - \mathbf{v}_O)U * dt)$. We present how to derive d_{min} analytically. Let $d(u)$ be defined as

$$d(u) = \|\mathbf{O}(k) - \mathbf{q}(k) - (\mathbf{v} - \mathbf{v}_O)u * dt\|, \tag{9}$$

where u is between 0 and U.

We next search for u minimizing $d(u)$. To search for u minimizing $d(u)$, we find u satisfying that $\frac{\partial d(u)}{\partial u} = 0$. Let u^c denote u satisfying this. We get

$$u^c = \frac{(\mathbf{v}(k) - \mathbf{v}_O(k)) \cdot (\mathbf{O}(k) - \mathbf{q}(k))}{\|\mathbf{v}(k) - \mathbf{v}_O(k)\|^2 * dt}. \tag{10}$$

If u^c exists between 0 and U while satisfying that $\mathbf{v}(k) \neq \mathbf{v}_O(k)$, then d_{min} is $d(u^c)$. Otherwise, the point on $L(\mathbf{q}(k), \mathbf{q}(k) + (\mathbf{v} - \mathbf{v}_O)U * dt)$, which is the closest to $\mathbf{O}(k)$, is one end point of this line. Thus, d_{min} is $min(\|\mathbf{O}(k) - \mathbf{q}(k)\|, \|\mathbf{q}(k) + (\mathbf{v} - \mathbf{v}_O)U * dt - \mathbf{O}(k)\|)$.

Recall that d_{min} denotes the minimum distance between $\mathbf{O}(k)$ and a point on $L(\mathbf{q}(k), \mathbf{q}(k) + (\mathbf{v} - \mathbf{v}_O)U * dt)$. In the case where d_{min} is bigger than r, the vehicle avoids collision within U steps, using the basic collision avoidance condition. This lemma is proved using d_{min} derived in the previous paragraph. □

4.2. Search for a Safe Velocity Vector Avoiding Collision

In the case where the collision avoidance condition is not satisfied for at least one obstacle point, the vehicle searches for a safe velocity to avoid collision. Once a safe velocity is found, then the vehicle uses the found velocity as its velocity within one step. This maneuver is called the *collision avoidance maneuver*.

To avoid an abrupt collision case, the vehicle does not move toward the goal directly, as the vehicle detects any obstacle using its sensors. Also, we set the minimum time interval I for the collision avoidance maneuver. Once a collision avoidance maneuver is triggered at time step k, the vehicle does not move toward the goal directly from k to $k + I$ time steps.

We introduce two collision avoidance maneuvers: *constant-speed safe maneuver* and *variable-speed safe maneuver*. The idea of these maneuvers is to search for a velocity satisfying the collision avoidance condition, followed by using the velocity to move the vehicle within one time step. The constant-speed safe maneuver is to search for a feasible velocity to avoid collision, while not changing its speed $\|\mathbf{v}\|$. The variable-speed safe maneuver is to search for a feasible velocity to avoid collision, while changing its speed $\|\mathbf{v}\|$.

4.2.1. Constant-Speed Safe Maneuver

We consider constant-speed safe maneuver in Section 4.2.1. Let $\mathbf{u} = (1, 0, 0)^T$ denote the heading vector of the vehicle in the body-fixed frame. Using (4), we have

$$\mathbf{v}(k) = \mathbf{R}(\psi(k), \theta(k), \phi(k)) * \|\mathbf{v}(k)\| * \mathbf{u}. \tag{11}$$

We first search for the orientation of the body, say $\psi(k), \theta(k), \phi(k)$, associated to $\mathbf{v}(k)$. Suppose that only the pitch and the yaw of the body are controlled to achieve the desired heading. This implies that we set $\phi(k)$ in (11) as zero. Then, using (11), we derive

$$\mathbf{h}(k) = (\mathbf{h}(k, 1), \mathbf{h}(k, 2), \mathbf{h}(k, 3))^T, \tag{12}$$

where $\mathbf{h}(k) = \frac{\mathbf{v}(k)}{\|\mathbf{v}(k)\|}$ is the heading vector at time step k. Also, $\mathbf{h}(k,1) = c(\psi(k)) * c(\theta(k))$, $\mathbf{h}(k,2) = s(\psi(k)) * c(\theta(k))$, and $\mathbf{h}(k,3) = -s(\theta(k))$. Here, $\mathbf{h}(k,j)$ denote the jth element of $\mathbf{h}(k)$.

We solve (12) to get

$$\theta(k) = atan2(-\mathbf{h}(k,3), \sqrt{\mathbf{h}(k,1)^2 + \mathbf{h}(k,2)^2}). \tag{13}$$

We further solve (12) to get $\psi(k)$. In the case where $c(\theta(k)) > 0$, we get

$$\psi(k) = atan2(\mathbf{h}(k,2), \mathbf{h}(k,1)). \tag{14}$$

In the case where $c(\theta(k)) < 0$, we get

$$\psi(k) = atan2(-\mathbf{h}(k,2), -\mathbf{h}(k,1)). \tag{15}$$

We use numerical methods to search for a safe velocity vector at time step $k+1$ to avoid collision. We search for a safe velocity in the *velocity searching space* considering the maximum turn rate α.

The vehicle's velocity vector in the body-fixed frame is $(\mathbf{v}(k), 0, 0)$, which is depicted as a bold arrow on Figure 1. Since Section 4.2.1 considers constant-speed safe maneuver, the velocity vector at time step $k+1$ exists on the sphere depicted on this figure. The velocity searching space in the body-fixed frame is depicted as a spherical dome on Figure 1. The velocity vector at time step $k+1$ must exist on this dome-shaped velocity searching space to satisfy the maximum turn rate α. See that the height of the dome is $\|\mathbf{v}\| * (1 - c(\alpha * dt))$.

Algorithm 1 provides numerical methods to search for a safe velocity vector at time step $k+1$ in the dome-shaped velocity searching space. N and M in this algorithm are positive constants, indicating the fineness of our search. As these constants increase, our search becomes more fine while increasing the computational load.

In Algorithm 1, $\mathbf{V}_c^B = R(\phi_0) * R(\psi_0) * \|\mathbf{v}(k)\| * \mathbf{u}$ indicates a velocity vector, which exists in the velocity searching space, in the body-fixed frame. In Algorithm 1, $R(\psi(k), \theta(k), 0)$ is multiplied to $\mathbf{V}_c^B$ so that we derive $\mathbf{V}_c^I$, a velocity vector in the inertial reference frame.

Algorithm 1 is designed to generate a safe velocity as fast as possible while minimizing the velocity change. This algorithm searches for a safe velocity vector at time step $k+1$ using the following FOR loop: for $\psi_0 = 0$, $\psi_0 = \psi_0 + \frac{\alpha * dt}{N}$, while $\psi_0 < \alpha * dt$. This FOR loop implies that we check a vector forming a small angle with $\mathbf{v}(k)$ before checking a vector forming a large angle with $\mathbf{v}(k)$. Once a safe velocity is found, then the algorithm is done, and the vehicle moves within one time step using the found safe velocity.

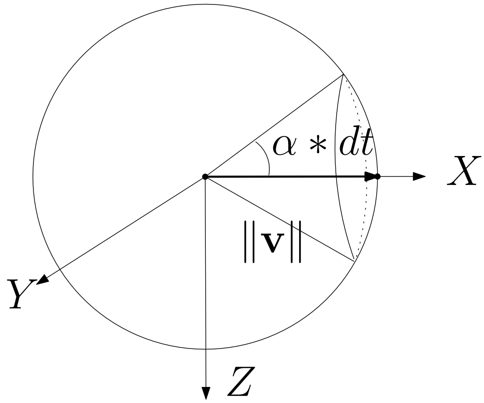

Figure 1. The velocity searching space in the body-fixed frame. The velocity searching space is depicted as a spherical dome.

Algorithm 1 Search for a Safe Velocity Vector at Time Step $k+1$ (constant speed)

the current time step is k;
for $\psi_0 = 0, \psi_0 = \psi_0 + \frac{\alpha * dt}{N}$, while $\psi_0 < \alpha * dt$ **do**
  for $\phi_0 = 0, \phi_0 = \phi_0 + \frac{2*\pi}{M}$, while $\phi_0 < 2 * \pi$ **do**
    $\mathbf{V}_c^B = R(\phi_0) * R(\psi_0) * \|\mathbf{v}(k)\| * \mathbf{u}$;
    $\mathbf{V}_c^I = R(\psi(k), \theta(k), 0) * \mathbf{V}_c^B$;
    if $\mathbf{V}_c^I$ satisfies collision avoidance condition within U time steps; **then**
      set $\mathbf{V}_c^I$ as the velocity vector at time step $k+1$;
      get out of all loops;
    end if
  end for
end for

4.2.2. Variable-Speed Safe Maneuver

We consider variable-speed safe maneuver in Section 4.2.2. Recall that the speed of the vehicle is bounded by s_{max} and that $\|\dot{\mathbf{v}}\| \leq a_{max}$. Thus, the speed of the vehicle at time step $k+1$ is bounded as follows.

$$A \leq \|\mathbf{v}(k+1)\| \leq A + B, \tag{16}$$

where $A = \|\mathbf{v}(k)\| - a_{max} * dt$, and $B = min(\|\mathbf{v}(k)\| + a_{max} * dt, s_{max}) - (\|\mathbf{v}(k)\| - a_{max} * dt)$ for convenience. B indicates the feasible range of the vehicle's speed at time step $k + 1$. Algorithm 2 provides numerical methods to search for a safe velocity vector at time step $k + 1$ using (16). See that the FOR loop for v_0 is at the most outer loop in this algorithm. This algorithm searches for a safe speed using the following FOR loop: for $v_0 = A + B$, $v_0 = v_0 - \frac{B}{N}$, while $v_0 > A$. This FOR loop implies that we check a velocity vector with high speed before checking a velocity vector with low speed.

Algorithm 2 Search for a Safe Velocity Vector at Time Step $k + 1$ (Variable Speed)

```
the current time step is k;
for v_0 = A + B, v_0 = v_0 - B/N, while v_0 > A do
  for ψ_0 = 0, ψ_0 = ψ_0 + (α*dt)/N, while ψ_0 < α * dt do
    for φ_0 = 0, φ_0 = φ_0 + (2*π)/M, while φ_0 < 2 * π do
      V_c^B = R(φ_0) * R(ψ_0) * v_0 * u;
      V_c^I = R(ψ(k), θ(k), 0) * V_c^B;
      if V_c^I satisfies collision avoidance condition within U time steps; then
        set V_c^I as the velocity vector at time step k + 1;
        get out of all loops;
      end if
    end for
  end for
end for
```

Algorithm 2 is suitable for a vehicle which can change its speed. Since many underwater vehicles can change its speed by changing propeller rate, Algorithm 2 is better than Algorithm 1 considering collision avoidance effectiveness. Simulations are performed to verify the effectiveness of Algorithm 2.

4.3. A Safe Velocity Vector Is Not Found

There may be a case where a safe velocity vector is not found using Algorithms 1 and 2. This implies that we cannot find a safe velocity vector which assures collision avoidance within U time steps in the future.

In this case, we set $U_s = max(U/2, 1)$. Using this new U_s, we check the collision avoidance condition within U_s time steps as follows: $L(\boldsymbol{q}(k), \boldsymbol{q}(k) + (\boldsymbol{v} - \boldsymbol{v}_O)U_s * dt)$ *does not meet the obstacle sphere centered at* $\boldsymbol{O}(k)$.

In the case where the above condition is not met, then we use Algorithms 1 and 2 to search for a safe velocity vector based on U_s, not on U. In the case where the above condition is met, then we use **v** to control the vehicle within one time step.

There may be a case where a safe velocity vector is not found using U_s. In this case, we decrease U_s iteratively using $U_s = max(1, U_s/2)$, while searching for a safe velocity vector using the decreased U_s. Considering time efficiency, we may stop decreasing U_s before it reaches 1.

We acknowledge that there may be a case where a collision is unavoidable. For instance, consider the case where the maximum turn rate and the maximum speed rate of the vehicle are too small. Due to the strict motion constraints, the vehicle cannot avoid collision with an obstacle which appears in front of the vehicle abruptly.

5. MATLAB SIMULATIONS

The effectiveness of our variable-speed collision avoidance control laws in Algorithm 2 is verified using MATLAB simulations. $dt = 1$ seconds, and $I = 10$ seconds. The goal of the vehicle is $(0, 500, 50)$. The task of the vehicle is to reach the goal while avoiding collision.

We say that a collision occurs in the case where the relative distance between the vehicle and an obstacle point is less than $r = 10$ meters. Here, r is set considering the uncertainty of range measurements. The initial speed of the vehicle is $6.3m/s$. The maximum speed of the vehicle is $12.75m/s$. Initially, we set $U = 300$. The maximum speed rate is $0.12m/s^2$. The maximum turn rate is $1.2rad/s$.

We consider a vehicle equipped with a forward-looking sonar sensor for collision avoidance. The maximum sensing range is 100 meters.

In the simulations, we consider limited FOV of sonar sensors. The sensor has 240 degrees horizontal scan and 240 degrees vertical scan. 24 rays are evenly generated in the horizontal direction as well as in the vertical direction. In total, 24*24 rays are generated to detect an obstacle at each time step.

To make our simulation more realistic, we simulated sensor rays with low detection rate 0.68. This implies that we randomly select 68 percents of all 24*24 sensor rays and use the selected rays to detect an obstacle boundary. As a selected ray intersects an obstacle, an obstacle point is generated at the intersection. In other words, an obstacle point is generated as a selected ray intersects an obstacle boundary. Obstacle points are then used as inputs of collision avoidance control laws.

In the following simulations, we consider arbitrarily shaped obstacles, each of which has distinct size. Also, we generated a non-convex tunnel in front of the goal position so that the vehicle must move through the tunnel before reaching the goal. The obstacle points detected by the vehicle's sensors are depicted with black points. The blue points indicate the vertices of each box-shaped obstacle. Also, the trajectory of the vehicle is depicted with red circles. The goal point is depicted with a green asterisk.

5.1. The Initial Position of the Vehicle Is $(0, 50, 90)$

This subsection considers the case where the vehicle starts from $(0, 50, 90)$. Figure 2 shows the top view of the 3D simulation result. Also, Figure 3 shows the 3D view of the simulation result. These figures show that the vehicle reaches the goal point while avoiding collision. See that obstacle points detected by on-board sensors are generated on obstacle boundaries and that the vehicle moves over the top of the first obstacle that it encounters. The height of the first obstacle is 100 meters. Also, the vehicle moves through the tunnel just before reaching the goal.

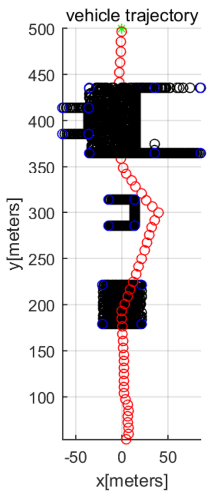

Figure 2. The vehicle starts from $(0, 50, 90)$. (top view).

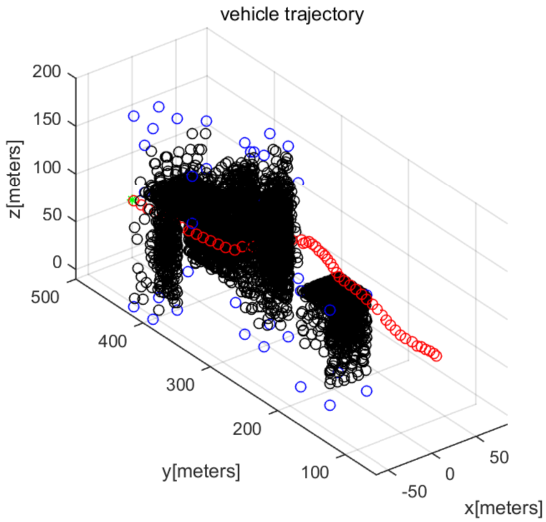

Figure 3. The vehicle starts from $(0, 50, 90)$. (3D view).

Figure 4 shows the yaw, pitch, and speed of the vehicle with respect to time, considering the scenario in Figure 2. At 6 seconds, the vehicle detects an obstacle boundary for the first time and begins to change its pitch to get over the detected obstacle. At 22 seconds, the vehicle detects the second obstacle (located at (0,300) in Figure 2) and changes its yaw to avoid the obstacle. See that the vehicle changes its attitude and speed in order to achieve collision avoidance while approaching the goal.

5.2. The Initial Position of the Vehicle Is $(0, 50, 50)$

This subsection considers the case where the vehicle starts from $(0, 50, 50)$. We first consider the case where all obstacles do not move. Figure 5 presents the top view of the 3D simulation result, and Figure 6 presents the 3D view of the simulation result. In these figures, obstacle points detected by on-board sensors are generated on obstacle boundaries. Since the initial z coordinate of the vehicle is not so large, the vehicle does not move over the top of the first obstacle that it encounters. Recall that the height of the first obstacle is 100 meters. Also, the vehicle moves through the tunnel just before reaching the goal.

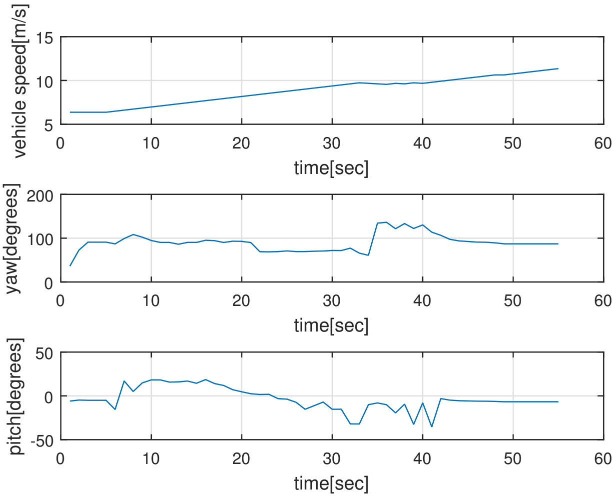

Figure 4. The yaw, pitch, and speed of the vehicle with respect to time, considering the scenario in Figure 2.

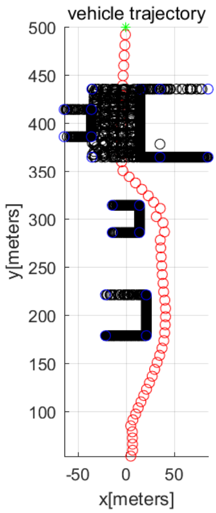

Figure 5. All obstacles do not move (top view).

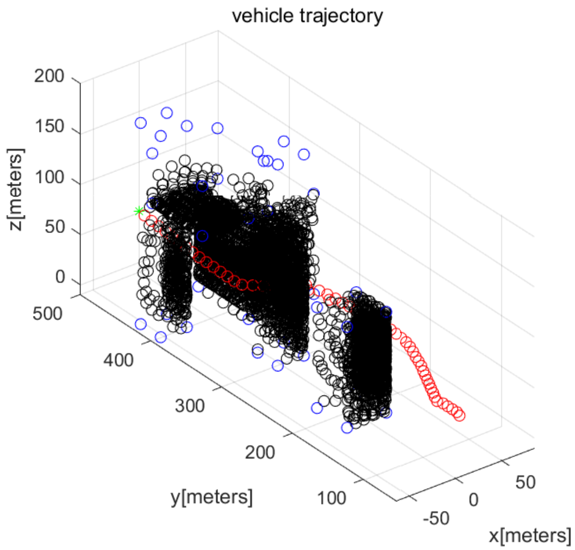

Figure 6. All obstacles do not move (3D view).

Figure 7 presents the yaw, pitch, and speed of the vehicle with respect to time, considering the scenario in Figure 5. At 6 seconds, the vehicle detects an obstacle boundary for the first time and begins to change its yaw to avoid colliding with the detected obstacle. At 42 seconds, the vehicle changes its yaw to get into the tunnel safely. Note that the vehicle changes its attitude and speed to avoid colliding with obstacles while approaching the goal.

We next consider the case where one obstacle (the first obstacle that the vehicle encounters) in Figure 5 move with velocity (0,-1,0) in m/s. The changing position of an obstacle is plotted every 3 seconds. Figure 8 presents the top view of the 3D simulation result, and Figure 9 presents the 3D view of the simulation result. Two blue lines on top of the first obstacle that the vehicle encounters indicate the trajectory of vertices of the moving obstacle. These figures verify that the vehicle reaches the goal point while achieving collision avoidance. To let the reader observe the vehicle's motion more clearly, we uploaded the movie of our vehicle on the following website: *https://youtu.be/1ItxMnKcjWY*.

Figure 10 presents the yaw, pitch, and speed of the vehicle with respect to time, considering the scenario in Figure 8. At 6 seconds, the vehicle detects

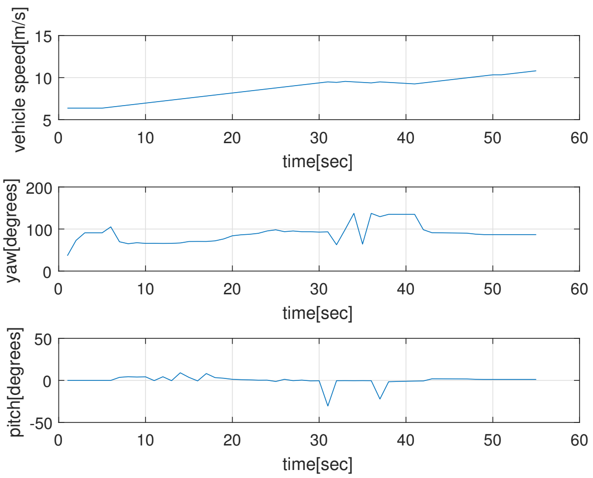

Figure 7. The yaw, pitch, and speed of the vehicle with respect to time, considering the scenario in Figure 5.

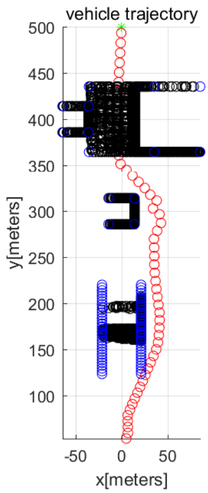

Figure 8. One obstacle moves (top view).

an obstacle boundary for the first time and changes its yaw abruptly to avoid colliding with the moving obstacle. See that the vehicle changes its attitude and

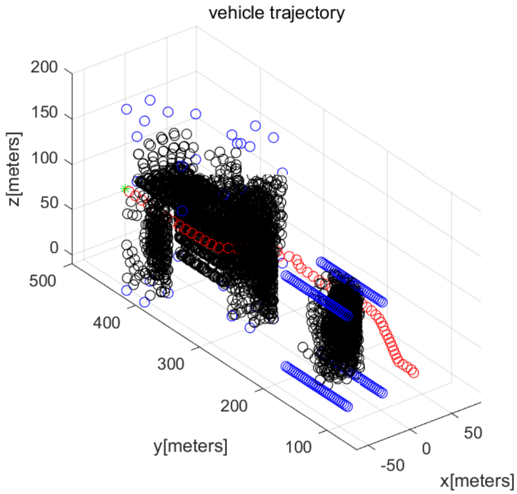

Figure 9. One obstacle moves (3D view).

speed to avoid colliding with obstacles.

Lastly, we observe the effect of changing dt. We set $dt = 5$ seconds instead of 1 second. We consider the case where one obstacle (the first obstacle that the vehicle encounters) in Figure 5 move with velocity (0,-1,0) in m/s. The changing position of an obstacle is plotted every 15 seconds. Figure 11 shows the top view of the 3D simulation result, and Figure 12 shows the 3D view of the simulation result. Since we set $dt = 5$ seconds, the trajectory of the vehicle gets less smoothed than the case where $dt = 1$ second. Also, obstacle points are distributed more sparsely than the case where $dt = 1$ second.

Figure 13 shows the yaw, pitch, and speed of the vehicle with respect to time, considering the scenario in Figure 11. At 6 seconds, the vehicle detects an obstacle boundary for the first time and change its pitch and yaw for collision avoidance. Figure 13 clearly shows that the vehicle's attitude and speed change at every $dt = 5$ seconds.

CONCLUSION

In this chapter, we introduce control laws for collision avoidance, considering 3D scenarios where a vehicle detects arbitrarily shaped obstacles using on-board

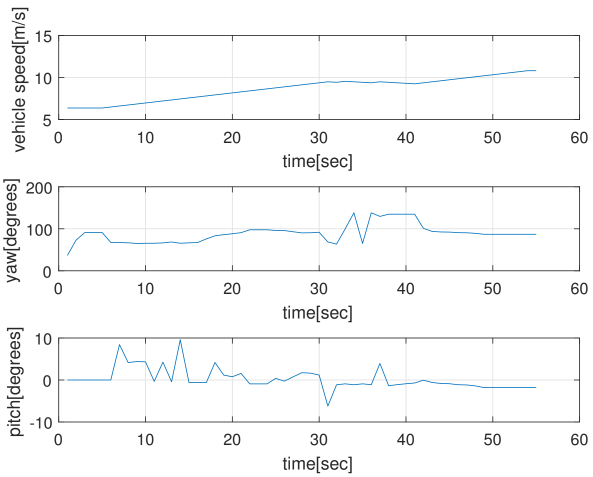

Figure 10. The yaw, pitch, and speed of the vehicle with respect to time, considering the scenario in Figure 8.

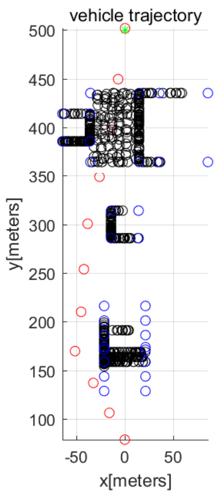

Figure 11. One obstacle moves (top view). We set $dt = 5$ seconds.

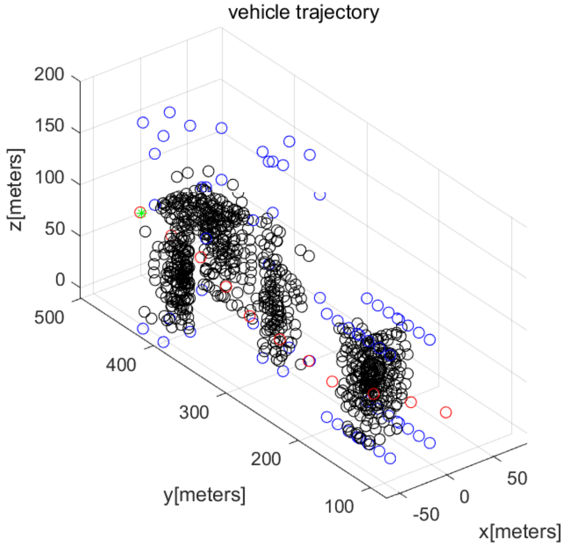

Figure 12. One obstacle moves (3D view). We set $dt = 5$ seconds.

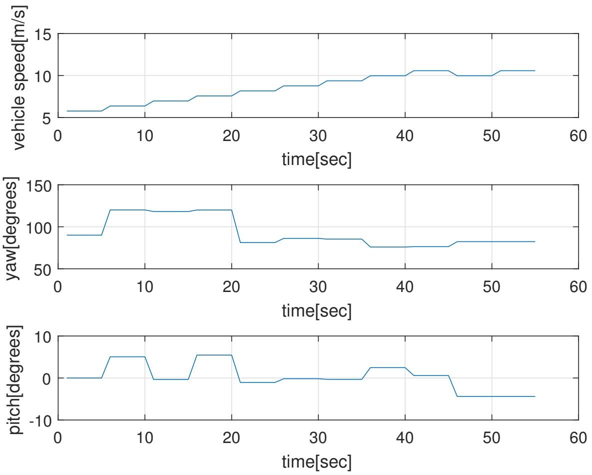

Figure 13. The yaw, pitch, and speed of the vehicle with respect to time, considering the scenario in Figure 11. We set $dt = 5$ seconds.

sensors. Our control laws are designed considering the maximum turn rate and the maximum speed rate of the vehicle. The effectiveness of our control laws is verified using MATLAB simulations. As our future works, we will verify the effectiveness of our control laws using experiments with underwater robots.

This chapter considers an obstacle which is not a deforming object. For an underwater application, shape shifting aquatic life forms can be considered as such deforming obstacles. Considering a deforming object, each obstacle point can move with different velocities. As our future work, we will consider collision avoidance with a deforming object.

REFERENCES

[1] Kuchar James K. and Yang Lee C., A review of conflict detection and resolution modeling methods. *IEEE Transactions on Intelligent Transportation Systems*, 1:179 – 189, 2000.

[2] Kuwata Yoshiaki, Wolf Michael T., Zarzhitsky Dimitri, and Huntsberger Terrance L., Safe maritime autonomous navigation with colregs, using velocity obstacles. *IEEE Journal of Ocean Engineering*, 39(1):110 – 119, 2014.

[3] Lalish E., *Distributed Reactive Collision Avoidance*. University of Washington, 2009.

[4] van den Berg Jur, Guy Stephen J., and Ming Lin Dinesh Manocha, Reciprocal n-body collision avoidance. *Robotics Research Springer Tracts in Advanced Robotics*, 70:3 – 19, 2011.

[5] Guy Stephen J., Chhugani Jatin, Kim Changkyu, Satish Nadathur, Lin Ming C., Manocha Dinesh, and Dubey Pradeep, Clearpath: Highly parallel collision avoidance for multi-agent simulation. In *ACM Siggraph/eurographics symposium on computer animation*, pages 177 – 187. ACM, 2009.

[6] Kosecka J., Tomlin C., Pappas G., and Sastry S., Generation of conflict resolution maneuvers for air traffic management. In *Proc. International Conference of Intelligent Robotic Systems*, pages 1598 – 1603, USA, 1997.

[7] Chakravarthy A. and Ghose D., Obstacle avoidance in a dynamic environment: a collision cone approach. *IEEE Transactions on Systems, Man and Cybernetics*, 28:562 – 574, 1998.

[8] Chang Dong Eui, Shadden Shawn C., Marsden Jerrold E., and Olfati-Saber Reza, Collision avoidance for multiple agent systems. In *Proc. of IEEE International Conference on Decision and Control*, pages 539–543, USA, 2003.

[9] Lalish E. and Morgansen K.A., Distributed reactive collision avoidance. *Autonomous Robot*, 32:207 – 226, 2012.

[10] Eriksen Bjorn-Olav Holtung, Horizontal collision avoidance for autonomous underwater vehicles. Master's thesis, Norwegian University of Science and Technology, 2015.

[11] Sunkara V. and Chakravarthy A., Collision avoidance laws for objects with arbitrary shapes. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 5158–5164, Dec 2016.

[12] Yang Xilin, Alvarez Luis Mejias, and Bruggemann Troy, A 3d collision avoidance strategy for uavs in a non-cooperative environment. *Journal of Intelligent and Robotic Systems*, 70:315 – 327, 2013.

[13] Chakravarthy Animesh and Ghose Debasish, Generalization of the collision cone approach for motion safety in 3-d environments. *Autonomous Robots*, 32:243 – 266, 2012.

[14] Choi Yi-King, Wang W., Liu Y., and Kim M., Continuous collision detection for two moving elliptic disks. *IEEE Transactions on Robotics*, 22(2):213–224, April 2006.

[15] Lu Liang, Zong Chengxing, Lei Xinyu, Chen Bozhi, and Zhao Ping, Fixed-wing uav path planning in a dynamic environment via dynamic rrt algorithm. *Mechanism and Machine Science. Lecture Notes in Electrical Engineering*, 408:271 – 282, 2017.

[16] Ju Ming-Yi, Liu Jing-Sin, Shiang Shen-Po, Chien Yuh-Ren, Hwang Kao-Shing, and Lee Wan-Chi, A novel collision detection method based on enclosed ellipsoid. In *Proceedings 2001 ICRA. IEEE International Confer-*

ence on Robotics and Automation (Cat. No.01CH37164), volume 3, pages 2897–2902 vol.3, May 2001.

[17] Dayangac Enes, Baumann Florian, Aulinas Josep, and Zobel Matthias, Target position and speed estimation using lidar. In *proc. of International Conference Image Analysis and Recognition*, pages 470 – 477, USA, 2016.

[18] Darms Michael S., Rybski Paul E., Baker Christopher, and Urmson Chris, Obstacle detection and tracking for the urban challenge. *IEEE Transactions on Intelligent Transportation Systems*, 10:475 – 485, 2009.

[19] Pomerleau Francois, Colas Francis, Siegwart Roland, and Magnenat Stephane, Comparing ICP variants on real-world data sets. *Autonomous Robots*, 34:133 – 148, 2013.

[20] Nobili Simona, Camurri Marco, Barasuol Victor, Focchi Michele, Caldwell Darwin G., Semini Claudio, and Fallon Maurice, Heterogeneous sensor fusion for accurate state estimation of dynamic legged robots. In *Proc. of Robotics: Science and Systems XIII*, pages 1 – 9, USA, 2017.

[21] Granstrom K. and Orguner U., A phd filter for tracking multiple extended targets using random matrices. *IEEE Transactions on Signal Processing*, 60(11):5657–5671, Nov 2012.

[22] Xinggang Wu, Cong Guo, and Yibo Li, Variable probability based bidirectional RRT algorithm for uav path planning. In *Proc. of Control and Decision Conference (2014 CCDC), The 26th Chinese*, pages 2217 – 2222, China, 2014.

[23] Roelofsen Steven, Martinoli Alcherio, and Gillet Denis, 3d collision avoidance algorithm for unmanned aerial vehicles with limited field of view constraints. In *proc. of IEEE International Conference on Decision and Control (CDC)*, pages 2555 – 2560, USA, 2016.

[24] Fossen T. I., *Guidance and control of ocean vehicles*. John Wiley and Sons, 1994.

[25] Opromolla Roberto, Fasano Giancarmine, Rufino Giancarlo, and Grassi Michele, Uncooperative pose estimation with a lidar-based system. *Acta Astronautica*, 110:287 – 297, 2015.

INDEX

S

T

U

V

W

George M. Roman
Editor

Underwater Vehicles

Design and Applications

nova
science publishers

www.novapublishers.com

ISBN 978-1-53618-876-9

9 781536 188769