UNIVERSITY OF RIJEKA
FACULTY OF ENGINEERING

Luca Braidotti

THE ASSESSMENT OF SHIP DAMAGE CONSEQUENCES BY A FAST SIMULATION OF COMPARTMENTS FLOODING

DOCTORAL DISSERTATION

Rijeka, 2021

UNIVERSITY OF RIJEKA
FACULTY OF ENGINEERING

Luca Braidotti

THE ASSESSMENT OF SHIP DAMAGE CONSEQUENCES BY A FAST SIMULATION OF COMPARTMENTS FLOODING

DOCTORAL DISSERTATION

Supervisor: Prof. dr. sc. Jasna Prpić-Oršić
Cosupervisor: Izv. prof. dr. sc. Marko Valčić

Rijeka, 2021

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET

Luca Braidotti

PROCJENA POSLJEDICA OŠTEĆENJA BRODA BRZOM SIMULACIJOM NAPLAVLJIVANJA ODJELJAKA

DOKTORSKA DISERTACIJA

Mentorica: Prof. dr. sc. Jasna Prpić-Oršić
Komentor: Izv. prof. dr. sc. Marko Valčić

Rijeka, 2021.

Mentorica rada: Prof. dr. sc. Jasna Prpić-Oršić
Komentor rada: Izv. prof. dr. sc. Marko Valčić

Doktorska disertacija obranjen je dana _____ u/na ____
_____, pred povjerenstvom u sastavu:

1. _____
2. _____
3. _____

Acknowledgements

This dissertation has been fully supported by the Croatian Science Foundation under the project IP-2018-01-3739.

Abstract

Recently, progressive flooding simulations have been applied onboard to support decisions during emergencies based on the outcomes of flooding sensors. To this end, both simulation accuracy and low computational effort are essential. A new mathematical formulation based on governing equations' linearization can improve both these conflicting requirements, being more easily applied when calculation time is limited or a large number of flooding simulation has to be carried out. Besides, databases of flooding simulations together with machine learning algorithms can be exploited to assess the consequences of a damage scenario (final fate, flooded compartments time-to-flood) based only on the time evolution of the ship floating position (heel, trim and sinkage). The proposed method can be the base for a new generation of onboard decision support systems, which does not require the installation of flooding sensors, enabling cheap retrofit on most of the existing fleet.

Keywords: damaged ship, progressive flooding simulation, emergency decision support, machine learning.

Sažetak

Na temelju signala senzora naplavljivanja u posljednje se vrijeme simulira tijek naplavljivanja kod hitnih slučajeva na brodu, a sve sa ciljem stvaranja podrške pri donošenju odluka odgovornih članova posade. Da bi simulacija bila učinkovita, ista mora biti dovoljno točna, ali i dovoljno brza bez značajnijih računalnih zahtjeva. Nova matematička formulacija temeljena na linearizaciji jednadžbi modela naplavljivanja može poboljšati oba ova sukobljena zahtjeva. Na taj se način predložena metoda znatno lakše može primijeniti kada je vrijeme izračuna ograničeno ili kada se mora provesti velik broj simulacija naplavljivanja. Osim toga, baze podataka koje sadrže rezultate simulacija naplavljivanja mogu se uz algoritme strojnog učenja adekvatno koristiti za procjenu posljedica scenarija štete (konačni ishod, vrijeme naplavljivanja odjeljaka) samo na temelju vremenskog zapisa položaja i orijentacije plutajućeg broda (poprečni nagib broda, trim i uron). Predložena metoda može biti osnova za novu generaciju sustava za donošenje odluka na brodu, koji ne zahtijevaju ugradnju senzora naplavljivanja, što pak omogućuje jeftinu nadogradnju postojećih brodova.

Ključne riječi: oštećeni brod, brza simulacija naplavljivanja, podrška odlučivanju u nuždi, strojno učenje.

Contents

Chapter 1

Introduction

Nowadays, the continuous growth in the size of passenger vessels is raising new safety concerns, since even more, human lives are at stake. Among the possible threats that can affect ship safety during its operative life, flooding is one of the most dangerous. It means, a vessel might not survive hull damage, requiring its evacuation and abandonment, which is a time-consuming task when involving thousands of persons [101]. In case of emergency, all the responsibility is in charge of the shipmaster, who often has to rapidly take hard decisions based on lacking or incomplete information. Human error, which is widely recognised as the major cause of maritime accidents [34], can then affect not only the accidents occurrence but also the control and mitigation of the flooding consequences. A bad decision in the initial phases of progressive flooding might compromise the crew and passenger safety, leading, in the worst case, to the loss of the vessel and/or many casualties.

1.1 Motivation

In the last decade, the need to provide onboard decision support during a flooding emergency has been highlighted by several accidents that occurred to passenger vessels. In particular, the sink of Costa Concordia in 2012 highlighted the vulnerability of large cruise ships and the difficulties associated with the prediction of flooding dynamics after an accident [30]. Large passenger ships have a very complex non-watertight subdivision within the watertight compartments as well as above the bulkhead deck. Such an internal subdivision may reduce the free surface moment due to floodwater but lead to possible asymmetric flooding scenarios and a difficult prediction of flooding sequences [6]. The vulnerability of a passenger vessel is also worsened by the short freeboard up to the bulkhead deck and short metacentric radius, which might lead the progressive flooding of intact watertight compartments and capsize, respectively.

Therefore, after a collision or grounding nor the ship final fate (new equilibrium position, capsizing, foundering) neither the time-to-flood (i.e., the time to reach the final stage of flooding) can be assessed without the aid of a proper tool. In particular, it is really handy for a master to evaluate the ship dynamics in case of flooding going beyond the static assessment

provided by a standard loading computer at the final stage of flooding. To this end, use can be made of enhanced onboard Decision Support Systems (DSS) that, after the identification of damage location and extent, are capable to simulate with fast simplified algorithms the flooding dynamics. Currently, damage detection can be performed only using flooding sensors, which are mandatory only for passenger ships built after 1 July 2010 [48]. Hence, on the older ships, serious safety issues can arise due to knowledge lacking during a flooding emergency. Even the number and location of damaged compartments might not be certain, forcing the master to make decisions based on heuristics or wasting time in collecting more reliable information.

A possible solution is to install a modern DSS on the whole passenger fleet. However, on the large majority of the current fleet, such a retrofit is hindered by its high cost since requires to install a complete flooding detection system (wiring, flooding sensors, new hardware, etc.). This is why cruise ship lines are seeking alternative solutions to improve the safety of their older ships with lower investments. In this context, a flooding-sensor-agnostic DSS might find application. Besides, such a system might be installed also on merchant ships, which are still not required to carry any flooding detection system by the current international rules. The present dissertation is moving in this context, proposing a methodology to improve the onboard decision support, with special attention to safety enhancement of older passenger vessels.

1.2 Literature Review

Before defining how the present dissertation would impact the background introduced above, the existing literature shall be analysed referring to two main topics: onboard decision support during a flooding emergency and flooding simulation techniques.

1.2.1 Onboard decision support

Recently, growing attention has been given to onboard decision support, ranging from intact ship condition [55, 4] to the operation and restoring of damaged vessel [26, 5]. Here, focus will be made only on the decision support on the bridge during a flooding emergency. In such a context, considering passenger vessels, several solutions are already available on the market. First, the international regulations enlist the mandatory documentation to be carried onboard [47]. However, paperwork documents, such as damage stability booklets or damage control booklets, are not considered suitable to provide emergency decision support by many seafarers. In fact, their consultation is very time-consuming and they encompass only the damage scenarios checked during ship design in the standard loading conditions which are not likely met during the ship's operative life. To partially overcome these issues, document digitalization is underway while several tools have been developed to support damage control [69, 43, 20, 58] or ship functionality restoring [22] at the final stage of flooding.

Besides, almost all the passenger ships are equipped with loading computers capable to

assess the damage stability employing the lost buoyancy method, although they are mandatory only since 1 January 2014 [50]. However, loading computers, as well as DSS for damage control, do not consider the transient and progressive flooding, which could lead to ship capsize before reaching the stable final floating position assessed statically. Furthermore, no time-to-flood estimation can be provided and usually the user is required to manually input the damaged rooms, which could be unknown on aged vessels not provided with flooding sensors. Finally, many seafarers claimed the scarce user-friendliness of most loading computers, whose graphical interface is not usually updated since the codes are subject to class societies approval [44].

These issues have been tackled through a direct onboard application of progressive flooding simulation codes, leading to the so-called "emergency computers". A conventional emergency computer exploits flooding sensors to assess the damage position and dimension [74, 89]. Then, according to such an input, it simulates the progressive flooding process by means of fast algorithms. The results are usually elaborated and presented with user-friendly interfaces [85, 73], often exploiting the virtual reality to enhance their decision support capability [109]. Emergency tools considering the time domain have been recently claimed to be the best option to provide decision support during a flooding emergency [87].

However, all these DSSs rely on flooding sensors capable to measure the floodwater level in all the main compartments of the ship or at least in the most critical [59]. As mentioned, the flooding detection system is installed only on a small number of recent passenger ships. Hence, on the large majority of the existing fleet, emergency DSSs based on flooding simulation can be installed only provided that a costly retrofit of the flooding detection system is done. To enable more economical solutions for older vessels, alternative damage recognition procedures shall be developed relying on a more basic set of sensors. A viable solution could be performing damage detection from the time records of the floating position of the damaged ship. Thus, the required sensors are limited to inclinometers (usually present in most of the ships) and a level radar to measure the ship lists and draught, respectively. Such a concept has been used in a system comparing the current record of the floating position with offline simulations retrieved from a very large database [102]. The most similar simulation was identified and provided as a forecast of the next flooding phases. Although functional, such a scheme requires interaction with all database items. Therefore, for complex geometries, it could lead to unreasonable response times and, thus, to large crew reaction time, which is not desirable during a flooding emergency.

1.2.2 Progressive flooding simulation techniques

Flooding simulation is the core of a modern emergency DSS. Many flooding simulation programmes are present in the literature. They can be grouped into three main families having an increasing precision and, consequently, computational effort too:

- the quasi-static (or quasi-dynamic) methods,
- the dynamic methods,

- the Computational Fluid Dynamics (CFD) methods.

The quasi-static methods are based on the assumption of horizontal and flat free surface inside the flooded compartments at each time instant during the flooding process. Furthermore, hydraulic laws are used to assess the flowrates through damages and internal openings. Moving from methods based on empirical formulations proposed since the '80s [100], several quasi-static methods were developed at the beginning of the Millennium, devoted to reproducing the progressive flooding of ships with complex subdivision or asymmetric flooding scenarios, e.g. [99, 105, 95]. Later on, the pressure correction technique was introduced to capture the effect of air compression while assessing the flow rates of water and air through openings and braches [79]. The method was validated against model [91, 80] and full-scale tests [83] wih good results. Afterwards, it was improved by applying a variable time step [81] and taking into account the roll motion. Another quasi-static method exploits a weighted predictor-corrector integration scheme to solve the governing equations and a non-linear system of equations to assess the pressure inside filled compartments [28]. Despite considering air pressure can improve simulation accuracy, its practical adoption is not straightforward, as will be discussed later on. Hence, it has been neglected in several recent simulations codes [84, 76]. In the framework of quasi-static methods, the steady Bernoulli equation is usually adopted to model the flowrate through the damages and internal openings. Recently an alternative technique has been proposed based on the dynamic orifice equation [64, 65]. The objective is to overtake some instabilities associated with the classical Bernoulli equation. However, a work that clearly shows the advantages of this formulation comparing its outcomes to the Bernoulli-based ones is lacking. For such a reason, most of the current flooding simulation methods are still based on steady Bernoulli equation, which is supported by a quite large literature and experimental validation [103].

The dynamic methods are mainly devoted to capturing the effect of waves and sloshing of floodwater on a damaged ship. They are also capable to reproduce the dynamic transient phases, i.e. the fast initial phase of flooding. Two methods are typically employed to assess the dynamic behaviour of a damaged ship in waves: the lumped mass method or the shallow water equations. The lumped-mass methods model the floodwater motions by concentrating its mass in a point while assuming a moving non-horizontal free surface [72, 71, 36]. The shallow water methods are based on the application of wave theory to model sloshing inside a flooded room [32, 23, 93]. These techniques can be used within a dynamic code accounting for floodwater spreading in internal spaces to simulate the transient flooding and/or progressive flooding of a damaged ship in harsh weather. For instance, lumped-mass method with moving free surface has been fully coupled with the ship motions, the inflooding jet, the floodwater motions and the flow through the openings [66]. Although the good accuracy in transient phase prediction, the main issue of dynamic codes deals with the computational time. It is usually comparable or higher than simulated physical time. Simplified dynamic methods have been also developed to reduce the computational effort being capable to reproduce a model-scale

experiment almost three times faster than real-time [2, 1]. These performances, although being remarkable for a dynamic method, are still not acceptable for decision support purposes on a cruise vessel, since the evacuation, if required, could be triggered too late.

Eventually, the CFD methods are even more time consuming than dynamic ones [92]. Hence, they have been applied only to study very specific problems where computational time was not considered a critical aspect. For instance, CFD methods have been employed to simulate the progressive flooding on a box-shaped barge [79, 37, 21, 16], to reproduce with high accuracy the transient flooding [40], to study the coupled dynamics during all stages of flooding on a very simple geometry [25] or the effect of air compression on a damaged river vessel [112].

In light of the previous review of flooding simulation methods, only the quasi-static ones can be applied when the computational time shall be minimised, as happens in direct onboard application or large dataset building up. Considering cruise ships having dense non-watertight subdivision, these simulation methods are satisfactory for both operative environment and design application [86]. However, among them, only a few methods have been applied onboard or specifically developed for this purpose. The fastest quasi-static method available in literature was designed during FLOODSTAND project [88]. It is based on the reverse solution of flooding chains and adopts several strong simplifications: air compression is neglected as well as all the dynamic effects. Moreover, a decreasing floodwater level is assumed along each flooding chain. Thus, flow reversals at breach or internal openings cannot be modelled. Nevertheless, the obtained results, although having lower accuracy compared to other quasi-static methods, were found sufficiently accurate for onboard application since they are capable to reproduce qualitatively the progressive flooding process [84].

1.3 Objectives

The purpose of this dissertation is to improve the accuracy of quasi-steady progressive flooding simulations while reducing the required computational effort. The first objective is achieved through the algebraic solution of the linearised equations governing the progressive flooding of the ship; the latter, by adopting an adaptive integration time step based on floodwater levels derivatives. The new simulation methodology enables faster onboard simulation as well as an easier generation of large databases of progressive flooding simulations in the design environment.

Besides, the present dissertation explores the application of machine learning algorithms to predict the main damage consequences (final fate, flooded compartments, time-to-flood) based only on the time evolution of the ship floating position. In the proposed process, water levels inside flooded rooms are not considered, opening the path toward a flooding-sensor-agnostic emergency DSS. In this context, different supervised algorithms and generation techniques for training database have been tested and compared in order to assess the best options fitting the studied classification and regression problems.

1.4 Organisation of the Thesis

The present dissertation is divided into two main parts: the development of a new quasi-static simulation technique and the application of machine learning algorithms to predict the damage consequences.

The new simulation technique provides a good compromise between accuracy and computational load compared to the simplified chain of flooding method [84]. In Chapter 2, the theoretical basis of the proposed progressive flooding simulation technique is presented, focusing on the flooding physics and modelling strategies. Then, in Chapter 3, the core of the new simulation method is described, providing three different sub-methods to predict the progressive flooding of a damaged vessel. Their advantages and limitations are introduced and discussed in more detail in Chapter 4, exploiting a full-scale example. Besides, the assessment of the integration time step is also presented, focusing on the control strategies to avoid the errors coming from its excessive size. In Chapter 4 the new progressive flooding simulation method is validated against model-scale experimental records. Then, as mentioned, it has been applied to a large passenger ship in order to prove its capability to tackle complex geometries.

In the second part of the dissertation, the new simulation technique is employed to generate large databases of flooding simulations to explore the adoption of machine learning algorithms in flooding consequences prediction. In Chapter 5, after a brief introduction to machine learning, the proposed strategy to assess onboard damage consequences without flooding sensors aid is presented. Then, the tested supervised algorithms (Decision trees, K-nearest neighbours, support vector machines) and different database generation methods (three Monte Carlo methods and one parametric) are introduced. The proposed process is tested in Chapter 6. A simple barge geometry has been designed to evaluate the accuracy of the tested algorithms and their computational performances. Then the best method is employed to explore the effect coming to the adoption of different training databases all validated against a large Monte Carlo dataset built according to SOLAS probability distributions.

In Chapter 7 the conclusions related to both the parts of the dissertation are provided, including the perspectives of the present dissertation. Focus will be made on the steps deemed necessary to develop and install onboard a flooding-sensor-agnostic emergency DSS.

Chapter 2

Theoretical bases of progressive flooding

In the present chapter, the theoretical bases of the proposed progressive flooding simulation strategy are presented. First the whole flooding process is introduced focusing on the particular case of a large passenger ship. Then the main assumptions and the equations governing the progressive flooding of a damaged ship are presented and discussed. Finally, the internal openings modelling strategy is addressed, being a very critical aspect for large passenger vessels.

2.1 Flooding process

As a consequence of a collision or grounding the watertight surface of the hull can be compromised, leading to flooding the ship's internal spaces. The flooding process of a damaged ship is a complex phenomenon that can be strongly affected by several aspects, e.g. the design of the ship's internal subdivision, the leakage or collapsing of internal openings and structures, the development of air pockets, etc. Hence, a general introduction to the flooding process is usefull to understand the problems that can be encountered in the simulation of the damaged ship behaviour.

The floodwater flow is normally non-stationary, hence, the flooding process is characterised by different pace during its time evolution. The flooding of a damaged ship is conventionally divided into three main phases: the dynamic transient, the progressive flooding and the final steady state [100]. In the first phase, the water rushes through the damage openings into the damaged rooms. Then, during the progressive flooding, the floodwater spreads inside the other internal rooms flowing through internal openings. As a consequence of flooding, three final fates are possible: the ship can sink, capsize or reach a new steady-state, i.e. a new equilibrium floating position where all the water levels inside the flooded rooms equalise the sea level. The time taken by the ship to reach its final fate is named time-to-flood. The Figure 2.1 shows the evolution of heel angle in three different flooding scenarios characterised by different fate, highlighting the three phases of the flooding process. In the following, dynamic transient and progressive flooding phases are discussed in more detail.

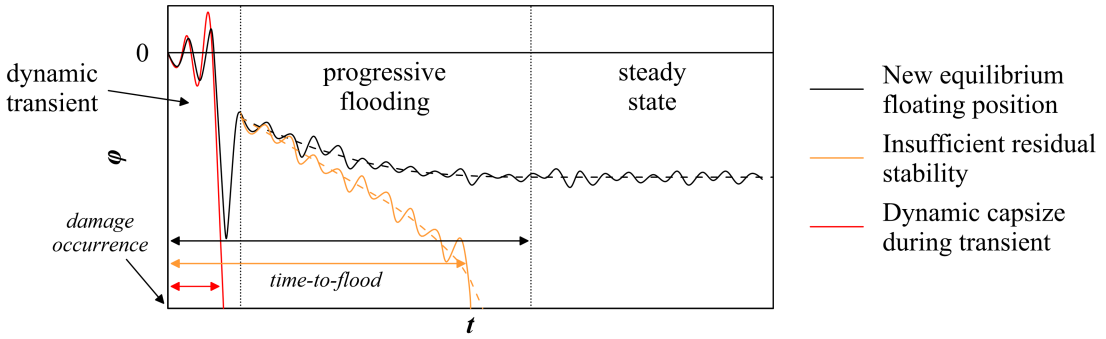

Figure 2.1 Qualitative sketch of heel angle time evloution during multiple flooding processes having different final fate

2.1.1 Dynamic transient

In the initial flooding phase, the floodwater rushes into the damaged rooms with a very fast pace. If the breaches are sufficiently large and deeply submerged, the room-filling can be considered almost instantaneous. In this phase dynamic phenomena usually arise, being non-negligible and might lead to sudden the ship capsizing after a few seconds or a few minutes [76]. In fact, considering a side damage and a non-symmetric subdivision, large heeling moments can be applied leading the vessel to rapidly heel towards the damaged side. Due to inertial effects, the heel angle can exceed the static value related to the complete flooding of the damaged rooms. This phenomenon is usually mitigated through cross flooding ducts having a proper cross-section to let floodwater spreading rapidly towards the undamaged side. Such an equalization reduces the heeling moments bringing back the ship close to the upright position. Cross flooding ducts are usually fitted to connect side tanks in a passenger ship, where the large list can also significantly slow down the evaluation process.

During the transient phase, flooding can be heavily affected by several factors. First, the penetration of the damage might compromise also the internal watertight boundaries of the ship, e.g. bulkheads. The damage penetration differs due to the accident: grounding [19] or collision [51]. Moreover, the damage is not necessarily instantaneous but can be opened in a finite time as the ship scrapes with the rocks/seabed or the striking ship draws back after a collision.

In a simulation, these aspects are not easy to be tackled, hence, it is common practice to neglect them [80]. Moreover, quasi-static simulation methods are not usually capable to properly reproduce the dynamic transient. To this end, dynamic methods can be applied with good results [66], being also capable to identify fast capsize scenarios. However, aiming at onboard application to provide decision support, the prediction of fast capsize cannot provide any useful information to the crew since the time is not sufficient to save the ship and people onboard by means of any active countermeasure.

2.1.2 Progressive flooding

During the progressive flooding, the floodwater spreads from the damaged rooms to the other internal spaces connected by open or non-watertight openings. Usually, due to the small size of the openings, this phase can be considered more quasi-steady than the transient [80]. As the floodwater spreads onboard, the floating position changes increasing the sinkage while the lists can evolve more randomly due to the subdivision geometry (non-watertight boundaries, internal openings, cross-flooding ducts, downflooding devices). Usually, also the ship stability is reduced during the progressive flooding until the vessel capsizes or sinks or, hopefully, reaches a new steady state. The duration of this phase ranges from a few minutes to several hours, hence, immediately after the damage it is hard to predict the time-to-flood without a flooding decision system based on a progressive flooding simulation engine [87]. As mentioned, the time-to-flood is essential on passenger ships, since it roughly corresponds to the available time for evacuation in a fatal scenario. This is why the present work mainly focuses on the progressive flooding simulation and the new ways to exploit these tools for onboard decision support.

The progressive flooding phase presents some main issues, especially on large passener ships, which are caracterised by a dens and complex non-watertigh subdivision. First, the geometry and closure status of the internal openings (open/closed) has a very strong effect on the progressive flooding [82]. The closure status could also lead to different fates in the same damage scenario. Moreover, a closed non-watertight door under a critical effective waterhead, i.e. the difference in waterhead between the rooms connected by the opening, might start leaking floodwater and eventually might collapse. The collapse order assessment is not straightforward [6], especially when multiple doors are connected to corridors, but can anyway strongly affect the process. Hence, in progressive flooding simulations, special attention shall be due to internal opening modelling. The problem has been addressed by IMO which provided guidelines regarding internal spaces modelling for flooding simulations [45]. Then an experimental campaign on this topic carried out during FLOODSTAND project improved the modelling methodology for leakage and collapse of several opening types [90]. Finally, some external factors can affect the progressive flooding process and also the time-to-flood. The presence of the striking ship for collisions and the sea bed/rocks for grounding might also change or distort the behaviour of the damaged ship during the progressive flooding. As for the transient, these interactions are usually neglected in all the flooding simulation methods. Besides, the effect of the waves on progressive flooding should be discussed. Several studies addressed the flooding of large passenger ships in waves [104, 70]. Due to the dense non-watertight subdivision, a large passenger ship's progressive flooding and survivability in waves were not proved to be heavily affected by waves [86] as happens for RoRo and RoRo-Pax vessels, which can rapidly capsize due to water accumulation on vehicle deck [94]. Large passenger ships mainly capsize during the progressive flooding due to insufficient residual stability. This is why, for this kind of vessel quasi-static methods can be applied with good results [30], especially for direct onboard

application.

2.2 Assumptions in progressive flooding simulation

The main assumptions of the proposed simulation technique are in line with most of the recent tools applicable to large passenger ships having low computational load [84, 29, 76]. First, in this work, the internal structures constituting the ship subdivision are assumed to be intact. Hence, the damage is modelled only as an opening on the hull surface and does not affect the internal boundaries, both watertight and non-watertight. Besides, the method assumptions deal with two main additonal aspects: the process dynamics and the air compression, which has been recently investigated by several studies [79, 29].

2.2.1 Process dynamics and reference systems

The progressive flooding process is considered quasi-static. This means that all the dynamic phenomena (e.g. sloshing, ship motions in waves, wave pumping, etc.) other than floodwater pouring are neglected. In this way, the computational effort required by the simulation method is minimised improving the direct onboard applicability. The floating position of the ship is considered fixed over a single integration time step and is evaluated taking into account the initial loading condition and the actual floodwater distribution. The sea free surface and the waterplanes of flooded compartments are assumed as flat and all the waterplanes are parallel to the sea free surface.

With these assumptions, two reference systems can be defined as shown in Figure 2.2). The first is the standard ship-fixed right-handed reference system $OXYZ$ having the origin at the intersection between baseline and after perpendicular. The second is an auxiliary earth-fixed right-handed reference system $O'xyz$ corresponding to the sea free surface. In such a system, the flooded compartments' waterplanes can be identified by a unique level indicated with z.

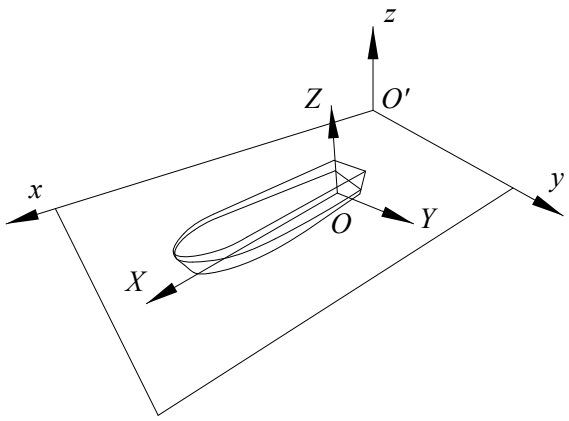

Figure 2.2 Ship-fixed and auxiliary earth-fixed reference systems [8]

This auxiliary system can be defined at each time step according to the fixed floating position of the vessel, i.e its temporary equilibrium position.

The hull and all the internal subdivision (rooms and openings) are be modelled in 3D through non-structured triangular meshes to apply a self-developed hydrostatic code based on pressure integration technique [12]. The flooding chains are not considered [84], removing the simplified hypothesis of decreasing level along the chain itself and, thus, permitting flow reversal.

2.2.2 Discussion on air compression

In the proposed simulation technique, all the internal spaces are assumed to be fully-vented. Air compression is significant only in rooms connected by small size vent openings, e.g. tanks equipped with ventilation pipes (relevant effects have been observed when the sum of all ventilation openings' areas is lower than the 10% of the area of the openings that pours floodwater into the room) [80]. Usually, ship 3D models utilized for stability assessment do not include such details, in order to streamline the flooding model configuration.

In the main compartments, stable air pockets can appear when all the openings are flooded. The air pockets can affect the simulation process having a non-negligible effect on the ship floating position and stability. However, the internal watertight boundaries are not always completely airtight [83], leading to a temporary presence of air pockets and a difficult assessment of effective vent area.

In a real environment, ship model uncertainties (e.g. permeabilities, areas of the openings, their constant discharge coefficients and leakage/collapse waterhead, etc.) might anyway affect simulation reliability [77, 7], vanishing the accuracy enhancement coming from air compression modelling. Furthermore, the introduction of air compression complicates the governing equations leading to higher computational loads. Therefore, especially when the simulation technique is designed for the onboard application, it is reasonable to consider all compartments as fully vented. Here, only air pockets are modelled employing a simplified method to consider their effect on ship floating position which can affect the ship stability and the progressive flooding simulation.

2.3 Progressive flooding governing equations

The progressive flooding dynamic is governed by two main equations: the conservation of mass and the conservation of momentum, that have to be instantaneously satisfied for each flooded room and for each connection among them.

2.3.1 Conservation of mass

Considering the i-th flooded room, the conservation of mass states that the sum of incoming and outgoing fluxes of floodwater is equal to the internal volume derivative:

$$\dot{V}_i = \sum_{j=1}^{O_i} Q_{ji} \tag{2.1}$$

where $\dot{V}_i$ is the time derivative of floodwater volume inside the room and Q_{ji} are the flow rates through the O_i openings connecting the selected room to other ones, denoted by index j. Assuming a constant waterplane area and a constant permeability, the conservation of mass can be written as:

$$\dot{z}_i \mu_i S_i = \sum_{j=1}^{O_i} Q_{ji} \tag{2.2}$$

where S_i, μ_i and $\dot{z}_i$ are the waterplane surface of the i-th room, its permeability and its floodwater level derivative respectively (Figure 2.3), all referred to the earth-fixed reference system. It is worth to notice that this assumption is exact only for wall-sided spaces that are extremely common in the subdivision layout of large passenger ships, but might introduce computational errors in the spaces located in the fore and aft body considering slender hull forms.

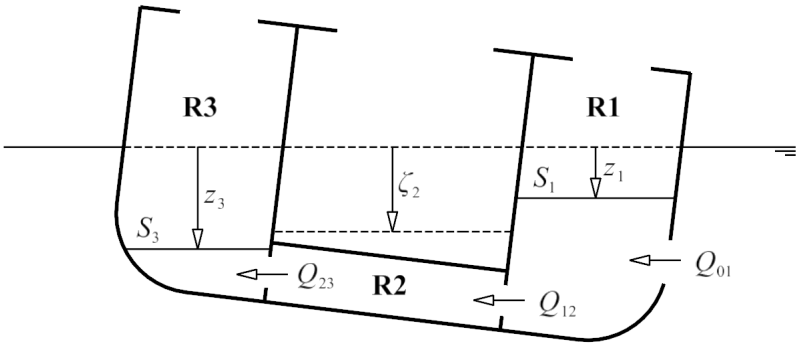

Figure 2.3 Sketch of a three rooms geometry [9]

2.3.2 Conservation of momentum

The conservation of momentum describes the flowrates among different flooded rooms. Hence, it shall be applied to each opening. Assuming a deeply submerged opening connecting two rooms and a stationary flow through the opening, the Bernoulli equation can be utilized considering a streamline starting from the centre of the room's free surface A and ending to the opening centre B:

$$\frac{p_{a_B} - p_{a_A}}{\rho} + \frac{1}{2}\left(u_B^2 - u_A^2\right) + g\left(z_B - z_A\right) + \frac{1}{2} k_L u_B^2 = 0 \tag{2.3}$$

where p_a is the air pressure, u the flow velocity and k_L the opening's pressure loss coefficient. Neglecting air compression, the pressures p_{a_A} and p_{a_B} are equal. Moreover, due to the larger value of free surface area compared to the opening cross-section, the flux velocity u_A can be considered negligible compared the the velocity u_B measured at the opening. With these assumptions, the flow rate Q_{ji} through the opening can be evaluated as:

$$Q_{ji} = C_{d_{ji}} A_{ji} \operatorname{sgn}\left(\hat{z}_j - \hat{z}_i\right) \sqrt{2g|\hat{z}_j - \hat{z}_i|} \tag{2.4}$$

where the level $\hat{z}$ is evaluated as:

$$\hat{z}_j = \max\left(z_j, z_{min_{ji}}\right) \tag{2.5}$$

where $z_{min_{ji}}$ is the distance of the lowest tip of the opening connecting i-th and j-th rooms from sea free surface (i.e. the lowest level tip level in the earth-fixed reference system), A_{ji} its cross-section and $C_{d_{ji}} = 1/\sqrt{1+k_L}$ its non-dimensional discharge coefficients depending on the orifice shape and edges. Despite some recent studies are questioning this assumption, it is still common practice in quasi-static simulation methods to assume constant values of the discharge coefficient ranging from 0.6 to 0.7 [77].

2.3.3 Simplified air pockets modelling

When a room is assumed airtight, air cannot flow through the room boundaries being forced to flow out only by main openings. In airtight rooms, the floodwater level rising can block all the openings by submerging their highest tip $z_{max_{ji}}$ measured in the earth fixed reference system. At that time, the remaining volume of air is trapped inside the room creating an air pocket, which has a constant air mass during the following phases of progressive flooding process. As the waterhead continues rising, the fixed air mass is compressed affecting the actual floodwater level inside the room. Hence, air pockets have several macroscopic effects on the progressive flooding process. They reduce the stability of the flooded ship due to the additional free surface moments. Moreover, air pockets usually slow down the progressive flooding since the water level inside the airtight compartment is lower than the predicted level under the hypothesis of fully-ventilation. The lower floodwater mass can reduce the sinkage and lists, thus reducing the effective waterheads on the openings too. Aiming to the onboard application, the slower change in floating position is a conservative error, while the stability reduction might be critical. This is why, despite the already mentioned problems connected to ventilation, simplified modelling of air pockets has been developed based on the correction of floodwater level.

Considering air an ideal gas and an isothermal process, the compression follows Boyle's law:

$$p_a V_a = k \tag{2.6}$$

where p_a and V_a are the pressure and the volume of air respectively and k is constant during the isothermal compression. The air pressure into the pocket can be considered as an equivalent waterhead defined as:

$$z_a = \frac{p_a}{\rho g} \tag{2.7}$$

Hence, the Equation 2.6 can be rewritten as:

$$z_{a_i} = \frac{z_0 V_{a_{0,i}}}{V_{a_i}} \tag{2.8}$$

where z_0 is the atmospheric pressure equivalent waterhead, $V_{a_{0_i}}$ is the volume of air trapped inside the i-th room as the highest opening is submerged (Figure 2.4).

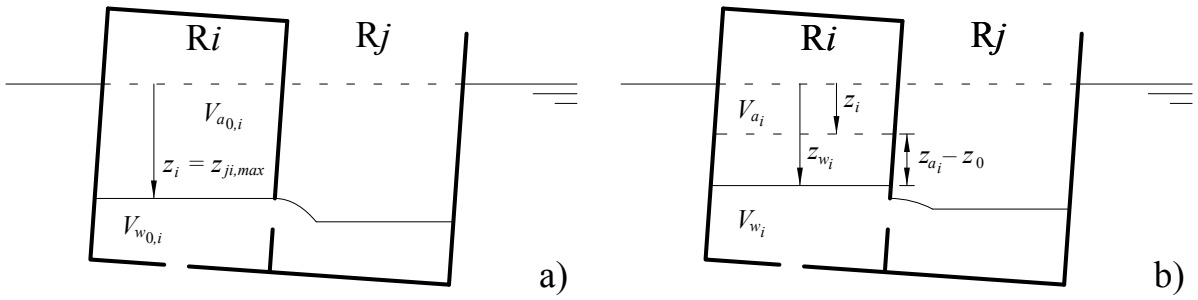

Figure 2.4 Sketch of an air pocket formation (a) and waterheads within the air pocket during compression (b)

Within an airtight room having a total waterhead z_i, the actual floodwater level z_{w_i} can be calculated as:

$$z_{w_i} = z_i - z_{a_i} + z_0 \tag{2.9}$$

Combining Equations 2.8 and 2.9, the actual level of floodwater, taking into account the air pocket effect, is obtained solving the following equation:

$$z_{w_i} = z - \frac{z_0 V_{a,i_0}}{V_{e_i} - V_{w_i}(z_{w_i})} + z_0 \tag{2.10}$$

where V_{e_i} and $V_{w_i}(z_{w_i})$ are the capacity and the volume of liquid corresponding to level z_{w_i} respectively. The Equation 2.10 is non-linear but can be solved iteratively in z_{w_i} at each time step for all the air pockets detected during the progressive flooding process. This approach enables a proper assessment of ship displacement, mass centre and stability of the ship, without introducing any complication in the formulation of the problem.

2.4 Openings modelling

In progressive flooding simulations, special attention shall be due to the openings modelling. The flowrate evaluated according to Equation 2.4 is valid for deeply submerged openings only. Otherwise, especially large openings, the free outflow implies a different flowrate requiring another formulation of the conservation of momentum [28]. Moreover, aiming at the onboard application on large passenger ships, the behaviour of closed non-watertight doors shall be also modelled since can have a strong effect of progressive flooding process [106, 82, 6]. Here, both the problems are addressed introducing a correction on the effective area of the opening.

2.4.1 Area correction for free outflow

Considering a large opening, different submersion statuses can be identified as shown in Figure 2.5. As mentioned, the Equation 2.4 provides the flowrate for the deaply submerged opening case, i.e. an opening which satisfy both the conditions $z_{max_{ji}} < z_i$ and $z_{max_{ji}} < z_j$ (Figure 2.5a) and thus $A_{ds_{ji}} = A_{ji}$. In the case shown in Figure 2.5d all the area is interested by free outflow, whereas in the remaining cases the free outflow area $A_{fo_{ji}}$ is calculated as:

$$A_{fo_{ji}} = A_{ji} - A_{ds_{ji}} \tag{2.11}$$

Assuming a flat rectangular opening, the free outflow flowrate can be evaluated as:

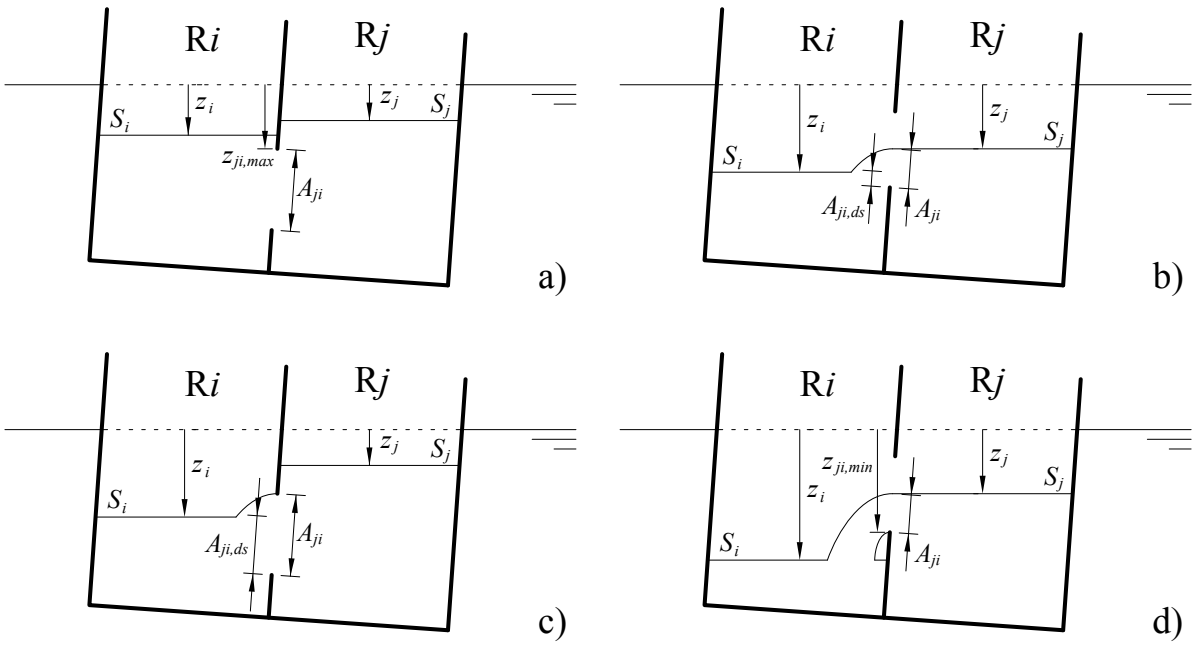

Figure 2.5 Sketch of different submersion status of a large opening.

$$Q_{fo_{ji}} = \int_{z_b}^{z_t} C_d \frac{A_{fo_{ji}}}{(z_t - z_b)} \sqrt{2g(z_j - z)}\, \mathrm{d}z \tag{2.12}$$

where $z_b = \max(z_i, z_{min_{ji}})$ and $z_t = \min(z_j, z_{max_{ji}})$. Integrating the equation 2.12, the flowrate is:

$$Q_{fo_{ji}} = C_d \sqrt{2g} \frac{A_{fo_{ji}}}{(z_t - z_b)} \frac{2}{3} \left[\left(z_j - z_b\right)^{\frac{3}{2}} - \left(z_j - z_t\right)^{\frac{3}{2}} \right] \tag{2.13}$$

The same flowrate can be evaluated using the deaply submerged formulation with a reduced effective area $A_{e_{ji}}$:

$$Q_{fo_{ji}} = C_d A_{e_{ji}} \sqrt{2g(z_j - z_b)} \tag{2.14}$$

Equating the Equation 2.13 and the Equation 2.14, a free outflow area correction factor can be defined as:

$$\gamma_{ji} = \frac{A_{e_{ji}}}{A_{fo_{ji}}} = \frac{2 \left[\left(z_j - z_b\right)^{\frac{3}{2}} - \left(z_j - z_t\right)^{\frac{3}{2}} \right]}{3\sqrt{(z_j - z_b)}(z_t - z_b)} \tag{2.15}$$

If the equation $z_j \leq z_{max_{ji}}$ is satisfied, the general evaluation of the correction factor can be further simplified as:

$$\gamma_{ji} = \frac{2}{3} \tag{2.16}$$

Hence, the deeply submerged formulation in Equation 2.4 can be applied in all cases shown in Figure 2.5, provided that the opening area is assessed according to:

$$A_{ji} = A_{ds_{ji}} + \gamma_{ji} A_{fo_{ji}} \tag{2.17}$$

It is worth to notice that this formulation is not applicable for horizontal openings in the earth fixed reference system. In such a case, the unitary correction factor shall be used.

2.4.2 Leakage and collapse

In progressive flooding simulations, the openings' leakage and collapse can have a very strong effect on the ship dynamics [82]. All the openings, especially the non-watertight ones, are not able to withstand high effective pressure heads $z_{e_{ji}} = \hat{z}_j - \hat{z}_i$. As the effective pressure head rises, closed doors start leak floodwater and then might collapse. Hence, this behaviour must be modelled especially for large passenger ships, where the complex internal subdivision presents a large number of non-watertight doors.

After some initial guidelines provided by IMO [45], this topic has been studied during the FLOODSTAND project, including a quite large experimental campaign on several different door's types [90]. As a result of the project, also leakage modelling guidelines have been

issued [53]. The proposed approach defines two critical effective waterheads: the leakage one z_l and the collapse one z_c. When the effective waterhead on a closed opening overtakes z_l, the effective area A_{ji} considering leakage is modelled as:

$$A_{ji} = A_{o_{ji}} \left(\alpha_{ji} + \beta_{ji} \left| z_{e_{ji}} \right| \right) \tag{2.18}$$

where $A_{o_{ji}}$ is the open area of the opening's submerged fraction whereas α and β are two coefficients related to the opening type and water flow direction. In fact, especially hinged doors usually better withstand the forces generated by the effective waterhead if they are opposite to his opening direction (out), and thus can more easily leak and collapse if they are concordant (into). When the waterhead overtakes z_c the door collapses and the effective area A_{ji} is assumed equal to $A_{o_{ji}}$.

In the present work, the applied coefficients and critical waterheads are reported in Table 2.1 for different doors types. The classification of doors has been defined according to IMO standards [45] which provide also the initial reference values. For light watertight doors, fire doors and cold room doors, the values as been updated according to [90, 53] whereas, for the elevators doors, values are taken from [60].

Table 2.1 Assumed leakage and collapse particulars for closed opening modelling

Type	**Code**	α_{into}	α_{out}	β_{into}	β_{out}	$z_{l,into}$	$z_{l,out}$	$z_{c,into}$	$z_{c,out}$
Watertight Door	A1	0.000	0.000	0.000	0.000	$+\infty$	$+\infty$	$+\infty$	$+\infty$
Light Watertight Door	A2	0.000	0.000	0.000	0.000	$+\infty$	$+\infty$	8.000	8.000
Semi Watertight Door	A3	0.200	0.200	0.000	0.000	1.000	1.000	3.000	3.000
Weathertight Door	B1	0.050	0.050	0.000	0.000	0.300	0.300	4.000	4.000
A-Class Sliding Fire Door	B2-S	0.025	0.025	0.000	0.000	0.000	0.000	1.000	1.000
A-Class Hinged Fire Door	B2-H	0.000	0.000	0.020	0.030	0.000	0.000	2.500	2.500
A-Class Double Leaf Fire Door	B2-DL	0.025	0.025	0.000	0.000	0.000	0.000	2.000	2.000
Cold Room Sliding Door	B2-C	0.000	0.000	0.010	0.010	0.000	0.000	3.500	3.500
Cold Room Hinged Door	B2-CH	0.000	0.000	0.020	0.030	0.000	0.000	2.500	2.500
B-Class Joiner Door	B3	0.000	0.030	0.030	0.000	0.000	0.000	1.500	1.500
Low Restriction of Flooding	C	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Elevator	E	0.090	0.090	0.000	0.000	0.000	0.000	1.000	1.000

Chapter 3

Progressive flooding simulation method

In the present chapter, the proposed approach is described to simulate the progressive flooding of a damaged ship in the time domain. Different mathematical formulations are presented, all based on the linearization of the governing equations of the phenomenon to permit its algebraic solution [8]. The linearised estimation of the floodwater level time-evolution inside the rooms enables the application of large integration time steps capable to reduce the computational effort. However, the application of a linearised technique requires special attention in dealing with completely filled rooms [9]. Here, two different strategies to face the problem are presented and qualitatively compared.

As already mentioned, aiming to the onboard application, it is essential to reach a satisfactory balance between accuracy and computational time. The linearised methods can offer an advantage in this sense compared to other techniques [80, 76]. Greater benefits can be achieved by applying an adaptive time step, as it has been demonstrated in [81]. However, the errors associated with too large time steps shall be taken under control in order to preserve simulation accuracy.

In the present chapter, after a description of the overall simulation strategy, the proposed mathematical formulations of the linearised problem are provided. Then, the adjustment strategy for the adaptive time step is introduced, focusing also on its limitation to reduce computational errors.

3.1 Main simulation algorithm

The proposed simulation technique is based on a single loop over a fixed floating position of the damaged ship. The main steps of the process are shown in Figure 3.1 and are briefly described hereinafter. In the first stage, the quasi-steady floating position of the ship is evaluated through a 3-DOF equilibrium [12], considering the loading condition of the ship at damage occurrence (lightship, dry cargo, liquids in tanks) and the total amount of floodwater present onboard. If the ship has not a sufficient residual buoyancy or stability, it sinks or capsizes respectively. In such cases, the simulation process is stopped.

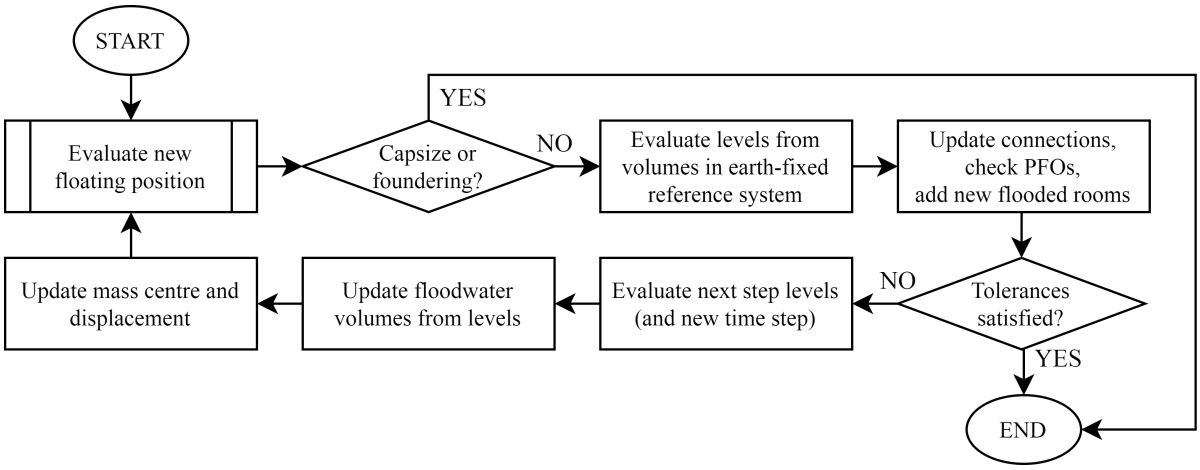

Figure 3.1 Main loop of the progressive flooding simulation [9]

Otherwise, the water levels within flooded rooms are evaluated in the earth-fixed reference system. The levels are obtained applying 1-dof equilibrium procedure on the room 3D mesh to assure the conservation of floodwater volume in each room.

Then, according to floodwater levels, the status of the openings of the ship model is checked. In detail, the unprotected openings connecting the ship's internal spaces to the external environment and the internal openings connecting a flooded room to an empty one are considered. In the following, these openings will be referenced as Progressive Flooding Openings (PFO). If a PFO is submerged, a new room is flooded and thus shall be taken into account during the next stages of progressive flooding simulation. Moreover, due to the complexity of the internal subdivision of a large passenger ship, an initially submerged PFO might completely emerge during the progressive flooding due to a change in heel angle, trim angle and/or sinkage. In such a case, a non-empty flooded room can be isolated from the others and therefore shall not be considered any more as a flooded room. Hence, in the subsequent phases of simulation, it is treated as an ordinary tank being taken into account only for the assessment of the quasi-steady equilibrium position of the ship.

Provided that no new flooded rooms have been detected, the quasi-steady floating position of the ship is analysed to identify whether the steady-state has been reached or not. The quasi-steady floating position is considered the final equilibrium position if four conditions are simultaneously satisfied. The first three are related to the floating position rate of change: the estimated velocity of the heel angle ϕ, trim angle θ and non-dimensional mean draught T_M/T_{M_0} shall be lower than the thresholds provided in Table 3.1. The last condition deals with the water level into the flooded rooms that shall be close to the sea free surface. Since the levels are defined in the earth-fixed reference system, they shall be null with a small tolerance to meet this condition. In order to make the threshold independent from vessel size, the levels are normalized with the mean draught T_{M_0} at damage occurrence as shown in Table 3.1. If the four conditions are met, the simulation is stopped and the ship survives the damage scenario. If a new iteration is required, the new floodwater levels are assessed according to the linearised

methodology. It is worth to notice that at this stage the integration time step is also adjusted according to flooding pace. Then, the floodwater volumes corresponding to the new levels are computed and utilized to update the damaged ship displacement and centre of mass. Formerly, a new iteration of the main loop can start with the assessment of the new quasi-steady floating position of the vessel.

Table 3.1 Applied thresholds for the stopping algorithm when the steady-state is reached.

Description	Bounded Value	Threshold
Heel Angle	$\lvert\phi_k - \phi_{k-1}\rvert/\mathrm{d}t$	0.00050 deg/s
Trim Angle	$\lvert\theta_k - \theta_{k-1}\rvert/\mathrm{d}t$	0.00005 deg/s
Mean Draught	$\lvert T_{M_k} - T_{M_{k-1}}\rvert/(T_{M_0}\mathrm{d}t)$	0.00001 1/s
Level	$\lvert z_i\rvert/T_{M_0}$	0.0001

3.2 Estimation of floodwater levels

In this section, the linearised methodologies are presented to assess the new levels within the flooded rooms of the vessel. The most straightforward approach is to define an Ordinary Differential Equations (ODE) system. The corresponding level estimation method is named Linearised ODE (LODE). However, this formulation is inadequate to simulate the behaviour of filled rooms, especially for flooding scenarios involving a large passenger vessel [6]. To overcome these issues a viable approach is represented by filled rooms grouping [8] or, better, considering a different mathematical formulation of the problem modelled through Differential Algebraic Equation (DAE) system [9]. The resulting level estimation methods are named LODE with Grouping (LODEG) and Linearised DAE (LDAE) respectively.

3.2.1 Linearization of ordinary differential equations

At a defined time instant t^* during the progresive flooding process, n flooded rooms are considered partially flooded. As mentioned, the amount of floodwater can be univocally defined by the levels mesuared in the earth-fixed reference system $\mathbf{z}^* = \mathbf{z}(t^*) = (z_1^*, ..., z_n^*)$. Combining the governing Equations (2.2) and (2.3) an ODE system can be defined that describes the evolution in time domain of levels $\mathbf{z} = \mathbf{z}(t - t^*) = (z_1, ..., z_n)$ during the progressive flooding in the form:

$$\dot{\mathbf{z}} = \tilde{f}(\mathbf{z}) \tag{3.1}$$

where

$$\tilde{f}_i(\mathbf{z}) = \sum_{\substack{j=0 \\ j\neq i}}^{n} \frac{K_{ji}}{\mu_i S_i(t^*)} \operatorname{sgn}(z_j - z_i) \sqrt{|z_j - z_i|} \tag{3.2}$$

and

$$K_{ji} = C_{d\,ji} A_{ji} \sqrt{2g} \tag{3.3}$$

In the sum above, a level $z_0 = 0$ also appears, i.e. the sea level that is always null in the earth fixed reference system. The ODE system can be linearised in $\mathbf{z}^*$ as:

$$\dot{\mathbf{z}}' = \mathbf{J}_{\tilde{f}}(\mathbf{z}^*)\mathbf{z}' + \tilde{f}(\mathbf{z}^*) \tag{3.4a}$$

$$\mathbf{z}' = \mathbf{z} - \mathbf{z}^* \tag{3.4b}$$

where $\mathbf{J}_{\tilde{f}}$ is the Jacobian matrix of the $\tilde{f}$ evaluated in $\mathbf{z}^*$. The Jacobian matrix can be decomposed in two $n \times n$ matrices as:

$$\mathbf{J}_{\tilde{f}}(\mathbf{z}^*) = \mathbf{S} \times \mathbf{Z}(\mathbf{z}^*) \tag{3.5}$$

having

$$\mathbf{S} = \text{diag}\left(\frac{1}{\mu_1 S_1(t^*)}, ..., \frac{1}{\mu_n S_n(t^*)}\right) \tag{3.6a}$$

$$Z_{ij}(\mathbf{z}^*) = \begin{cases} \text{if} \quad i = j & -\sum_{\substack{k=0 \\ k \neq i}}^{n} \frac{K_{ki}}{\sqrt{|z_k^* - z_i^*|}} \\ \text{if} \quad i \neq j & \frac{K_{ji}}{\sqrt{|z_j^* - z_i^*|}} \end{cases} \tag{3.6b}$$

Both matrices $\mathbf{S}$ and $\mathbf{Z}(\mathbf{z}^*)$ are symmetric. Moreover, $\mathbf{S}$ is positive-defined. Therefore, the Jacobian matrix is diagonalizable and, by applying the single value decomposition, can be rewritten as:

$$\mathbf{J}_{\tilde{f}}(\mathbf{z}^*) = \mathbf{V} \times \mathbf{D} \times \mathbf{V}^{-1} \tag{3.7}$$

where $\mathbf{D}$ is the $n \times n$ diagonal matrix of the eigenvalues of $\mathbf{J}_{\tilde{f}}$, $\mathbf{V}$ the $n \times n$ matrix whose columns are its eigenvectors and $\mathbf{V^{-1}}$ the inverse of $\mathbf{V}$.

Multiplying both sides of Equation (3.4a) by $\mathbf{V^{-1}}$, the linearised system can be simplified as:

$$\dot{\mathbf{r}} = \mathbf{D}\mathbf{r} + \mathbf{v} \tag{3.8a}$$

$$\mathbf{r} = \mathbf{V}^{-1}\mathbf{z}' \tag{3.8b}$$

$$\mathbf{v} = \mathbf{V}^{-1}\tilde{f}(\mathbf{z}^*) \tag{3.8c}$$

Since $\mathbf{D}$ is a diagonal matrix, the equations in System (3.8a) are decoupled. Hence, the

initial value problem has the following algebraic solution:

$$r_i = \frac{v_i \left(e^{D_{ii}(t-t^*)} - 1 \right)}{D_{ii}} \tag{3.9}$$

Therefore, combining the Equations (3.4b), (3.8b) and (3.9), the time-domain evolution of the level of the i-th flooded room can be estimated according to:

$$z_i = z_i^* + \sum_{j=1}^{n} \frac{V_{ij} v_j \left(e^{D_{jj}(t-t^*)} - 1 \right)}{D_{jj}} \tag{3.10}$$

This equation can be utilized to assess the new levels after an integration time step $\mathrm{d}t = t - t^*$.

It is worth to notice that the linearised solution has an exponential trend. Hence, since the eigenvalues of the Jacobian matrix are always negative, all the levels tend to the null level, i.e. the sea level by definition. This behaviour is physically consistent with the progressive flooding problem.

3.2.2 Grooping of filled rooms

The LODE formulation has been derived with the assumption of partially filled rooms. However, especially for complex non-watertight subdivision having multiple decks, during the progressive flooding, several rooms can be filled by water. In such a case, the left terms in the Equation 3.1 related to the filled rooms are null (the level derivative and the waterplane are both nulls). Hence, the system is no more ordinary and a straightforward application of the Equation 3.10 assuming the free surface area S_i equal to the room's top area introduce a computational error. In fact, with this assumption, the conservation of mass is no more respected for the filled rooms where an amount of floodwater is still added and thus "disappears" from the ship model. Aiming to onboard application, this kind of error cannot be considered conservative, since implies an overestimation of the time-to-flood.

A viable solution to overcome this problem is the filled rooms' grouping [8]. Applying this approach, as a k-th room is filled, it is removed from the equation system and it is grouped with the i-th room connected by the opening having the largest value of K_{ki}. Hence, in the subsequent steps of the simulation, the rooms previously connected to the filled one are associated with the group, which is modelled by means of a single level z_i, a single waterplane area S_i and a mean permeability μ_i computed as the weighted average of the grouped rooms' ones.

This method, called LODEG, reduces the dimension of the system in presence of filled rooms. This can lead also to slightly reduce the computational effort required to compute the linearized solution. However, the grouping assumptions still involve a theoretical error. In fact, the pressure loss of the opening connecting the two grouped rooms is implicitly neglected. Nevertheless, the LODEG error can be considered conservative, since drives to an underestimation of the time-to-flood. Considering the onboard perspective, an underestimation is less critical

provided that does not lead to rush ship evacuation inducing panic among crew or passengers.

The different behaviour of LODE and LODEG methods can be easily understood considering an example. Simulations have been carried out on the simple geometry reported in Figure 3.2. The box-shaped barge is 75 m long, has a breadth of 20 m and a fixed draught of 12 m. The 5 main rooms has a watertight tanktop located at 5 m from baseline to be progressively filled by water. The rooms are connected through a set of openings specified in Table 3.2. Two flooding scenarios called 1 and 2 can be simulated by opening the R1-R2-A or R1-R2-B respectively.

In Figure 3.3 the simulations are compared on the two flooding scenarios applying both the LODE and LODEG methods. It is worth to notice that applying the LODE formulation the shape of the level curve is continuous after a room's filling, i.e. when the level reaches the tanktop level. As mentioned, this behaviour is caused by the inconsistent addition of floodwater inside filled rooms.

On the other hand, applying the LODEG formulation, each room, after filling, is grouped with the previous one in the flooding chain, i.e. the sea in both the flooding scenarios. Hence, the assumed waterhead inside filled rooms is the sea level, which is null. The different Δz lead to a divergence of the results obtained with the two formulations after the room R1 filling. In flooding scenario 1 the levels within R2 remain comparable only due to the fast increasing of waterhead up to the sea level inside R1 experienced applying the LODE formulation. In terms of time-to-flood the effect of the mathematical formulation is significant: in flooding scenario 1 the time-to-flood differs of about 287 s whereas in scenario 2 of about 187 s.

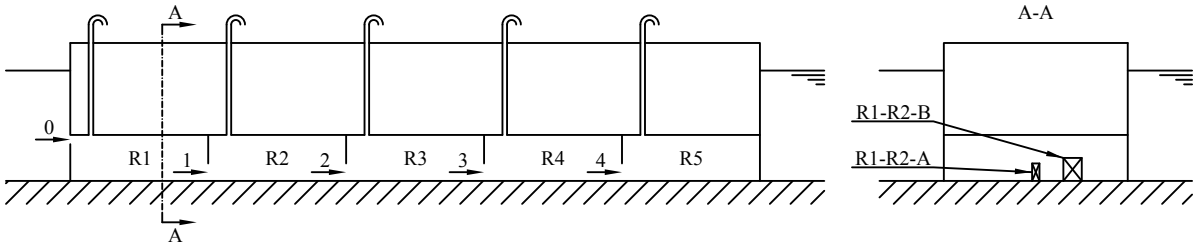

Figure 3.2 Simple test arrangement for qualitative comparison of methods with flooding sequence [9]

Table 3.2 Main characteristics of the test case openings. $C = (X_C, Y_C, Z_C)$ is the centre of the opening in ship-fixed reference system

id	h **(m)**	w **(m)**	X_C **(m)**	Y_C **(m)**	Z_C **(m)**
SEA-R1	1.000	20.000	0.000	0.000	4.500
R1-R2-A	1.900	0.800	15.000	0.000	4.500
R1-R2-B	2.500	2.000	15.000	0.000	4.500
R2-R3	1.900	0.800	30.000	0.000	4.500
R3-R4	1.900	0.800	45.000	0.000	4.500
R4-R5	1.900	0.800	60.000	0.000	4.500

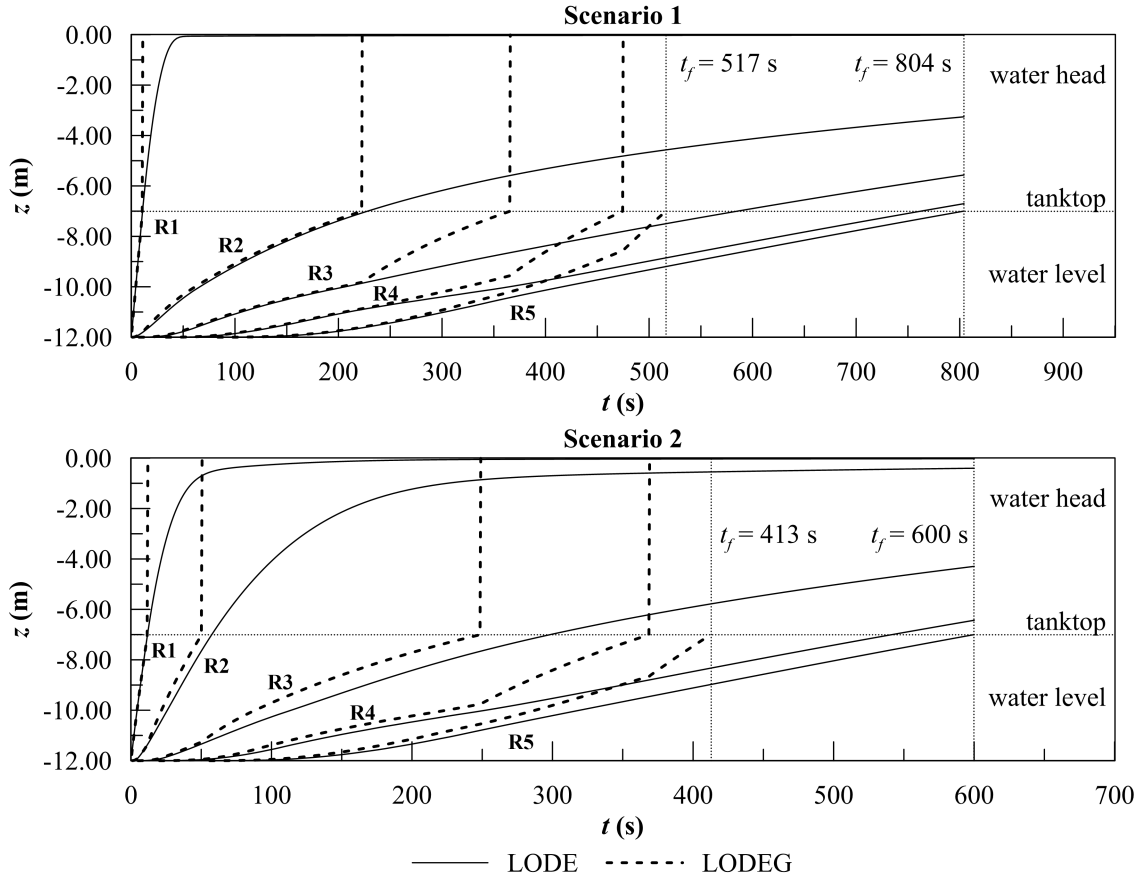

Figure 3.3 Comparison of the LODE and LODEG method on a simple test arrangement

3.2.3 Linearisation of differential algebraic equations

The LODEG mathematical formulation, despite validated with success against model experiments [8], can be applied as long as the pressure loss on the openings connecting the grouped rooms are negligible. This condition is satisfied only if the opening K_{ki} has a higher order of magnitude compared to the ones related to the other openings connecting the filled room to the other internal spaces or the external environment. Considering a typical large passenger ship, this hypothesis is not likely respected. In fact, the openings connecting the internal spaces are usually light watertight doors, fire doors or light joiner doors having standard dimensions. This can limit the application of the LODEG formulation in a real full scale environment.

This is why, a more reliable technique to consider completely filled rooms is here proposed. The mathematical formulation of the problem has been changed, definying a non-linear semi-explicit differential-algebraic equations system, instead the ODE one. Considering n partially flooded rooms and $m-n$ filled rooms, the DAE system is defined as:

$$\begin{cases} \dot{\mathbf{z}} &= f(\mathbf{z}, \boldsymbol{\zeta}) \\ \mathbf{0} &= g(\mathbf{z}, \boldsymbol{\zeta}) \end{cases} \tag{3.11}$$

where the $\mathbf{z}$ are the floodwater levels in the earth fixed reference system inside the n partially

filled rooms while $\boldsymbol{\zeta} = \boldsymbol{\zeta}(t-t^*) = (\zeta_{n+1}, ..., \zeta_m)$ are the waterheads representing the hydrostatic pressure within the $m-n$ completely filled rooms. The functions f and g are again obtained combining the governing Equations (2.2) and (2.3) and can be written as:

$$f_i(\mathbf{z}, \boldsymbol{\zeta}) = \sum_{\substack{j=0 \\ j \neq i}}^{n} \frac{K_{ji}}{\mu_i S_i} \operatorname{sgn}(z_j - z_i) \sqrt{|z_j - z_i|} + \sum_{j=n+1}^{m} \frac{K_{ji}}{\mu_i S_i} \operatorname{sgn}(\zeta_j - z_i) \sqrt{|\zeta_j - z_i|} \tag{3.12a}$$

$$g_i(\mathbf{z}, \boldsymbol{\zeta}) = \sum_{j=0}^{n} K_{ji} \operatorname{sgn}(z_j - \zeta_i) \sqrt{|z_j - \zeta_i|} + \sum_{\substack{j=n+1 \\ j \neq i}}^{m} K_{ji} \operatorname{sgn}(\zeta_j - \zeta_i) \sqrt{|\zeta_j - \zeta_i|} \tag{3.12b}$$

As mentioned, the DAE system cannot be solved analytically either in linearised form. Standard integration techniques can be applied, e.g. 4th order Runge-Kutta, ODE-45 or Runge-Kutta-Fehlberg. However, these methods require short integration time steps compared to the linearised techniques. Hence, to reduce the computational effort, the System 3.11 is solved separating the differential and algebraic parts.

The differential part is linearised in $(\mathbf{z}^*, \boldsymbol{\zeta}^*)$ and solved with the technique presented in Section 3.2.1 obtaining an explicit solution of the problem for the partially filled rooms.

Once the levels $\mathbf{z}(\mathrm{d}t)$ in partially filled rooms are defined at the next time step $\mathrm{d}t$, the algebraic part of the System (3.11) can be solved in $\boldsymbol{\zeta}$. The resulting non-linear system of equation is differentiable having a Jacobian matrix $\mathbf{J}_g(\mathbf{z}(\mathrm{d}t), \boldsymbol{\zeta})$ defined as:

$$J_{g_{ij}} = \begin{cases} -\sum_{k=0}^{n} \frac{K_{ki}}{\sqrt{4|z_k(\mathrm{d}t) - \zeta_i|}} - \sum_{\substack{k=n+1 \\ k \neq i}}^{m} \frac{K_{ki}}{\sqrt{4|\zeta_k - \zeta_i|}} & \text{if} \quad i = j \\ \frac{K_{ji}}{\sqrt{4|\zeta_j - \zeta_i|}} & \text{if} \quad i \neq j \end{cases} \tag{3.13}$$

Therefore, the system can be solved as a least-square problem applying the Levenberg–Marquardt algorithm and assuming as initial condition $\boldsymbol{\zeta}^* = \boldsymbol{\zeta}(t^*)$. The waterheads which satisfy the minimum condition are assumed as system solution $\boldsymbol{\zeta}(\mathrm{d}t)$. Thus, all the unknown variables are computed without requiring the grouping of the filled rooms.

This so-called LDAE formulation is theoretically capable to assess the effective waterhead inside the filled rooms even in case of openings having equal pressure loss (equal shape, area, discharge coefficient, etc.). For instance, Figure 3.4 compares the application of LODEG and LDAE methods on the simple test geometry introduced in Section 3.2.2. The example shows how the LODEG method provides results comparable to LDAE until the openings connecting filled rooms have non-comparable size. According to Equation (2.3), the pressure loss is proportional to the square of the flow velocity through the opening and thus can be negligible for the larger ones. For this reason, in scenario 1, the results are almost equal until the complete filling of R2.

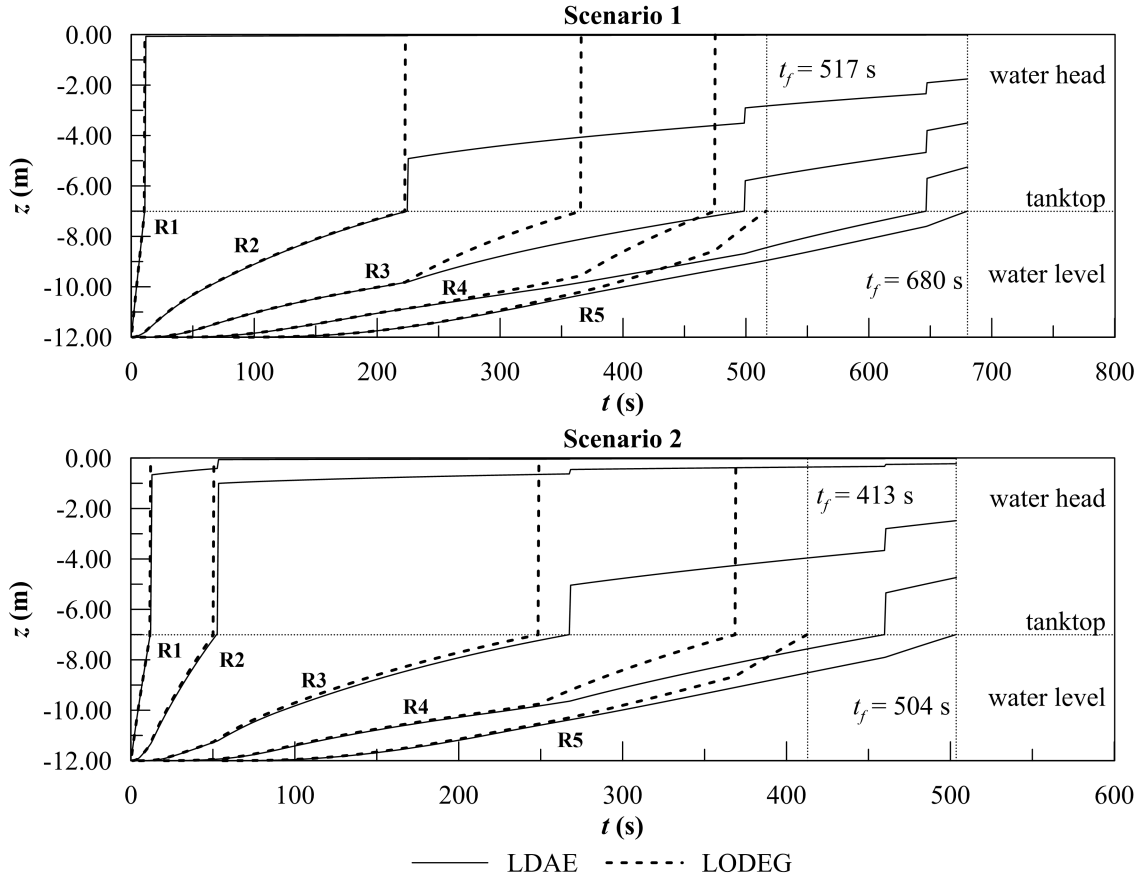

Figure 3.4 Comparison of the LDAE and LODEG method on a simple test arrangement

Afterwards, the progressive flooding pace presents relevant differences: due to the room grouping, the LODEG method underestimates the time-to-flood of about 25% compared to LDAE one. In scenario 2, the opening R1-R2-B, being considerably larger than R2-R3 and smaller than the breach, reduces the water head overestimate introduced by LODEG method after R2 filling, extending the comparable results up to R3 filling. Then, again the grouping-related issues can be noted, leading to an overall underestimation of time-to-flood of about 15% when the LODEG method is applied. Hence, the LDAE method results can be considered physically more consistent than both the LODE and LODEG ones.

3.3 Adaptive integration time step

As mentioned in Section 2.1, the flooding process is characterised by a variable pace. During the transient, a very fast shift from the intact floating position occurs. Even though the dynamic phenomena other than floodwater pouring are neglected, the simulation of the initial flooding phases requires a very small integration time step [76]. During the progressive flooding, the process slows down until a new equilibrium position is reached, following in first approximation, an exponential trend. In this phase, larger time steps can be applied without affecting the

simulation accuracy [81]. Considering a large passenger vessel, during the progressive flooding the pace can also sudden accelerate due to the collapse of non-watertight openings (fire dorrs, joiner doors, etc.) or lolling before ship capsize. All these considerations push the application of non-constant integration time step, capable to assure a good simulation accuracy while reducing the computational load. Hereinafter, a time step adjustment based on the floodwater level derivatives is proposed. Moreover, an explicit strategy to control the errors associated with very large time step is presented.

3.3.1 Adaptation procedure

The water level derivatives $\dot{\mathbf{z}}$ have been already proved to be a good indicator for integration time step adjustment [81]. However, up to now, they have only be utilized to switch between multiple discrete values of the integration time step. Here, a continuous formulation is proposed, applicable disregarding the vessel size [9]. The integration time step is adapted according to the following formulation:

$$\mathrm{d}t = k_{\mathrm{d}t} \frac{T_M^*}{\max \dot{\mathbf{z}}^*} \tag{3.14}$$

where $\dot{\mathbf{z}}^*$ and T_M^* are the time derivatives of the floodwater levels and the ship mean draft, respectively, evaluated at the initial time instant t^*, while $k_{\mathrm{d}t}$ is a non-dimensional parameter.

Applying any linearised formulation, the floodwater level derivatives can be easily assessed according to Equations (3.1) and (3.2). To assure the dimensional consistency of the adjustment procedure, a length shall be applied as a scaling factor. Among several tested options, the mean draft has been selected. In fact, mean draught is the best option to account for scaling effects of the progressive flooding process, having a direct relationship with the maximum effective waterhead applicable at the hull breaches. Eventually, the $k_{\mathrm{d}t}$ parameter in Equation (3.14) controls the accuracy level of the simulation and can be experimentally calibrated to reach the appropriate balance between accuracy and computational effort.

3.3.2 Maximum time step

The application of large integration time steps has a positive effect on the computational time required by the progressive flooding simulation. However, the application of too large time steps can affect the simulation accuracy. Several phenomena occurring at a specific instant can induce distorsions or errors in a progressive flooding simulation. Among them, the most relevant are:

- the submersion of a PFO, which implies the addition of a new flooded room to the equations' system;
- the filling of a flooded room that triggers the switch from the differential to algebraic part of the equations' system when LDAE method is applied;

- the collapse of a closed non-watertight opening due to overtaking of the critical effective waterhead;
- an excessive increment in floodwater volumes leading to a large variation of the ship floating position, which is non-consistent with the hypothesis of fixed floating position over a single time step.

For instance, Figure 3.5 shows the effect of the application of an unbounded variable time step on Scenario 1 introduced in Section 3.2.2. In this case, the late recognition of rooms' filling due to large time step drives to a total delay of 11 s compared to a solution where the time step is limited. The limitation of the time step allows to better simulate the occurrence of all the critical events, whose occurrence has, thus, to be predicted. The most simple approach is to assess the event occurrence a-posteriori and then recalculate the last time step applying a properly reduced integration time step. However, to reduce the computational effort of the simulation, such an iterative procedure should be avoided in favour of an explicit methodology, not implying recalculation. Here another solution is proposed, based on an a-priori estimation of maximum time step $\mathrm{d}t_{max}$ from the already computed floodwater levels during latest time steps.

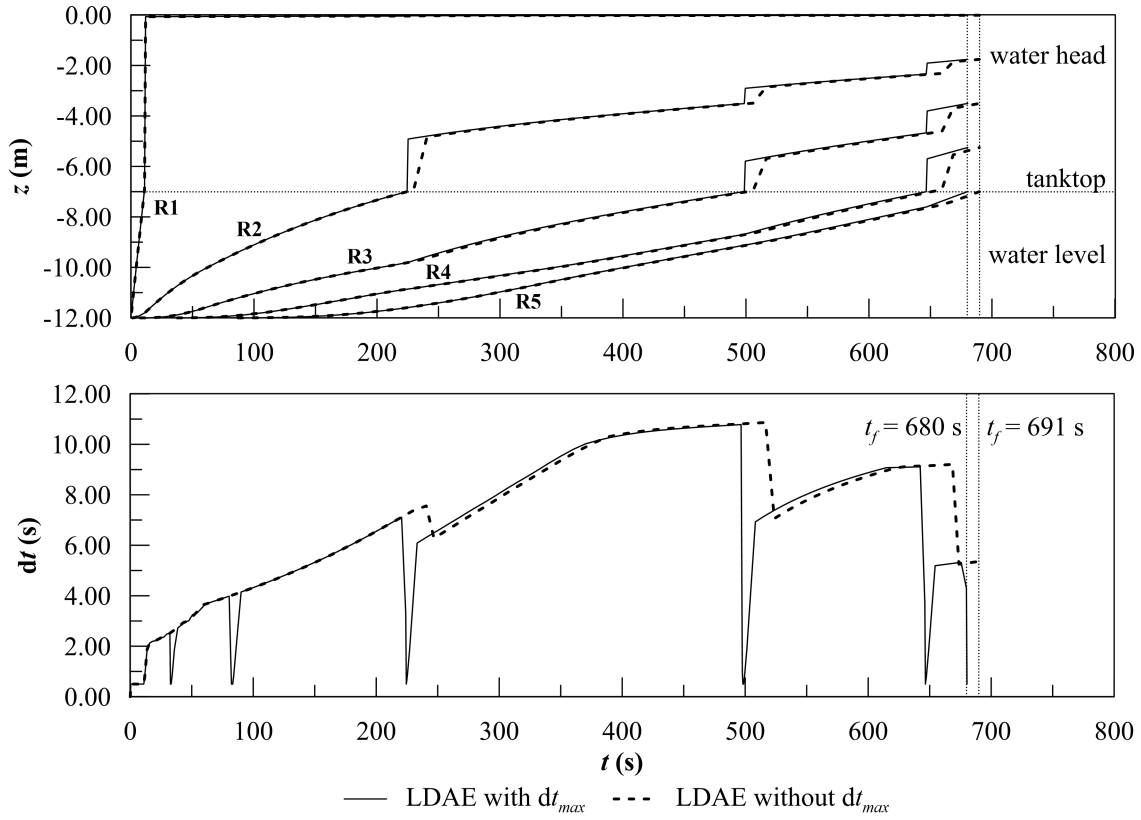

Figure 3.5 Simulation distortion due to late switch from differential to algebraic formulation in Scenario 1 on test case arrangement (Section 3.2.2) with adaptive time step (k = 0.01)

At a certain instant, a specific critical event occurs if a critical level z_c is reached in an observed room; for instance, z_c can be the level of the lowest edge of a PFO in the earth-fixed reference system to detect a new flooded room or the highest point of a room mesh to detect room-filling. Considering a specific room, the floodwater level rises during the progressive flooding and reaches z_c at a time instant t_c. Applying any linearised technique, the raising of floodwater level is determined according to Equation 3.10, which is not invertible. Hence, the time t_c cannot be assessed without applying an iterative process, that should be again avoided to reduce the computational time.

Alternatively, a first estimation of t_c can be obtained by exponential fitting of z according to the two latest values observed inside the room. Under this assumption, the estimated level $\tilde{z}(t)$ is evaluated as:

$$\tilde{z}(t) = z^{**} \left(\frac{z^*}{z^{**}} \right)^{\frac{t}{\mathrm{d}t^*}} \tag{3.15}$$

where $\mathrm{d}t^*$ and z^* are the last time step and level, respectively and z^{**} is the penultimate level. The described condition is sketched in Figure 3.6. Since the Equation (3.15) is invertible, the maximum time step associated to the selected room $\mathrm{d}t_c$ can be evaluated analytically as:

$$\mathrm{d}t_c = t_c - t^* = \left[\frac{\ln\left(\frac{z_c}{z^{**}}\right)}{\ln\left(\frac{z^*}{z^{**}}\right)} - 1 \right] \mathrm{d}t^* \tag{3.16}$$

The described strategy is directly applicable to the cases of lower PFO submersion or room-filling. For the case of door collapse, z shall be replaced by $|z_j - z_i|$ and z_c is the collapse effective waterhead. Moreover, the maximum time step can be determined also to avoid too large change in the ship floating position. In such a case, the variations of heel angle, trim angle and non-dimensional mean draught shall be considered instead of the level z. Besides, the z_c has to be arbitrarily stated as the maximum allowable variation of the floating position per main loop iteration. For instance, the values provided by [9] are reported in Table 3.3.

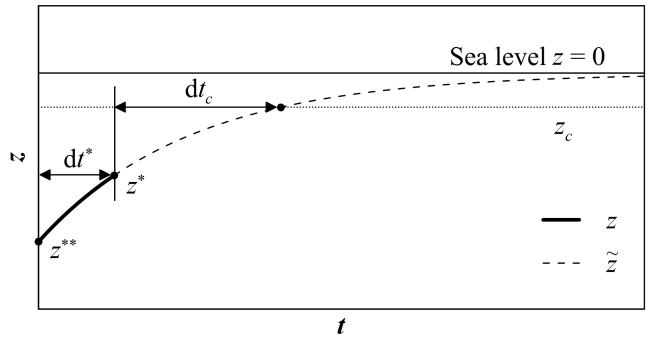

Figure 3.6 Maximum time step control strategy for a room having a critical level z_c

Table 3.3 Maximum accepted variation of floating position per main loop iteration

Description	**Bounded Value**	**Threshold**
Heel Angle	$\lvert\phi_k - \phi_{k-1}\rvert$	0.100 deg
Trim Angle	$\lvert\theta_k - \theta_{k-1}\rvert$	0.050 deg
Mean Draught	$\lvert T_{M_k} - T_{M_{k-1}}\rvert / (T_{M_0})$	0.005

At last, the maximum time step dt_{max} applied to assess the next levels is determined as the minimum among the ones obtained for PFOs submersion, rooms' filling, non-watertight doors collapsing and large variation of the floating position.

Chapter 4

Validation of the progressive flooding simulation method

In the present chapter the simulation technique presented in Chapter 3 is calibrated and validated against experimental data. Moreover, a comparison with other simulation methodologies is provided. Then, the three proposed formulations are compared on a full scale geometry to highlight their different behaviour in the challenging environment of a large passenger ship.

4.1 Validation in model-scale

The validation of the flooding simulation method adopting the LDAE formulation is here presented. First the adaptive time step procedure is calibrated against experimental data in model scale, then, the simulation results are compared with experimental records and with the results coming from other techniques available in the literature [103]. The selected test geometry is a box-shaped barge that has been used during an experimental campaign carried out at Helsinki University of Technology model basin [78]. The tests were devoted to validating the pressure correction technique [80], however they have been, here, chosen in accordance being recommended by ITTC guidelines for the validation of flooding simulation tools [52].

4.1.1 Test arrangement

The box-shaped barge utilized for validation purposes has the main particulars summarised in Table 4.1. The barge is equipped with a portable element 0.7 m long that can be used to test different internal layouts. The subdivision adopted in the present study is provided in Figure 4.1. In such a layout, the portable element is divided in two 0.35 m long compartments having by a transversal bulkhead. Moreover, a tank-top and a main deck are fitted in each compartment at 0.150 m and 0.475 m respectively. Eventually, two wing tanks have been fitted in the fore compartment between the tank-top and the main deck. The wing tanks are symmetric and 0.160 m wide.

Table 4.1 Main particulars of box-shaped barge

Description	Symb.	Value		Description	Symb.	Value	
Length overall	L_{OA}	4.000	m	Breadth	B	0.800	m
Draught	T	0.500	m	Height	D	0.800	m
Displacing Volume	∇	1.450	m^3	Metacentric Height	GM	0.110	m

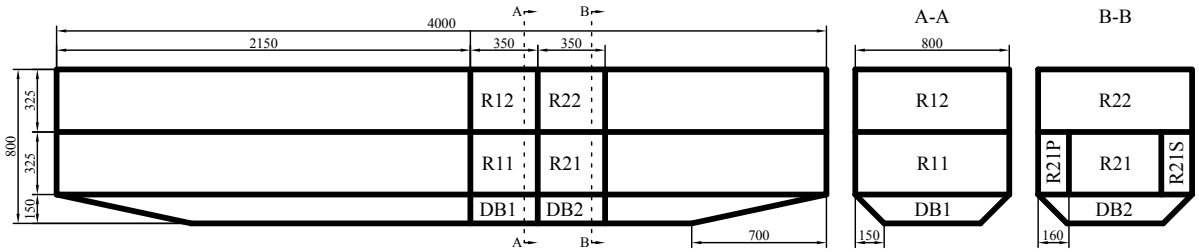

Figure 4.1 Subdivision layout of the box-shaped barge (dimensions in mm*) [8]*

The hull and the rooms have been modelled by means of non-structured triangular meshes, having maximum panel area equal to 0.001 m^2. All the internal and external boundaries (hull shell, bulkheads, decks) have a 0.01 thickness which has been considered in barge model. Hence, unitary permeabilities have been applied to all the internal spaces. The adopted 3D model of the box-shaped barge is shown in Figure 4.2.

The internal spaces are interconnected by closable openings in order to test multiple flooding scenarios. The shape, location and dimension of the openings is provided in Table 4.2 along with their discharge coefficients, which have been experimentally determined. The internal openings as well as the breaches are again modelled by means of non-structured meshes, but the maximum panel area is reduced to 0.0001 m^2

It shall be noted that the double bottom in the aft. compartment (DB1) is airtight. Hence, stable air pockets might there appear, having a strong influence on the progressive flooding process. This is why the simplified method for air pocket modelling has been introduced in Section 2.3.3. Moreover, the wing tanks are equipped with vent pipes having limited cross section. Hence, as the openings R21-R21S or R21-R21P are blocked by floodwater, air compression occours in the wing tanks.

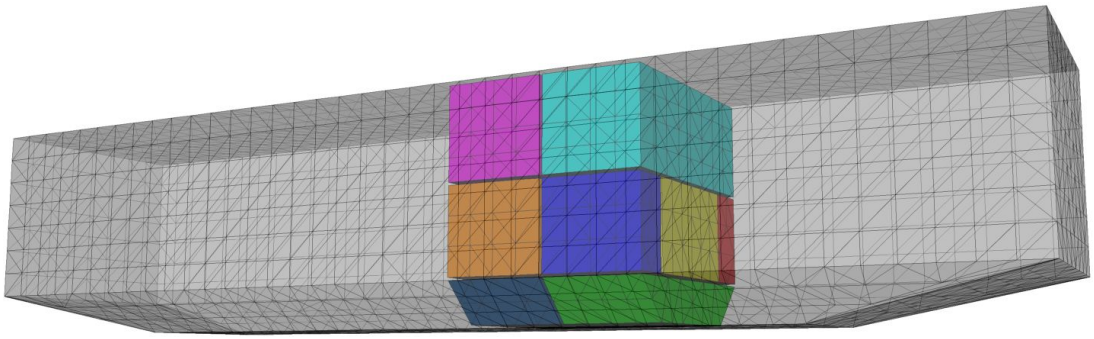

Figure 4.2 Mesh of the hull and internal rooms of box-shaped barge [8]

Table 4.2 Main characteristics of the barge internal openings. $C = (X_C, Y_C, Z_C)$ is the centre of the opening in ship-fixed reference system

id	Type	Size (m)	X_C (m)	Y_C (m)	Z_C (m)	C_d (-)
SEA-DB1	Rectangular	0.060×0.040	2.325	0.000	0.005	0.78
SEA-DB2	Rectangular	0.025×0.025	2.675	0.000	0.005	0.83
SEA-R21S	Rectangular	0.060×0.040	2.675	0.395	0.315	0.78
DB1-DB2	Circular	$D = 0.020$	2.500	0.000	0.075	0.80
DB2-R21	Rectangular	0.060×0.040	2.800	-0.170	0.155	0.78
R21-R21S	Rectangular	0.020×0.200	2.675	0.240	0.260	0.75
R21-R21P	Rectangular	0.020×0.200	2.675	-0.240	0.260	0.75
R21-R11	Circular	$D = 0.020$	2.500	0.000	0.315	0.80
R21-R22	Rectangular	0.100×0.100	2.750	-0.160	0.475	0.72
R11-R12	Rectangular	0.100×0.100	2.400	-0.160	0.475	0.72
R12-R22	Rectangular	0.080×0.200	2.500	0.000	0.580	0.72

However, it has been observed that this phenomenon has only a limited impact on the progressive flooding process, thus the test cases has been successfully used to validate progressive flooding simulation codes neglecting air compression [103, 76].

4.1.2 Flooding scenarios

In the present study, three different flooding scenario (A, B, C) has been simulated on the box-shaped barge. The recorded flooding sequences are sketched in Figure 4.3 and are briefly described hereinafter. For more details about test arrangement and flooding scenarios, reference can be made to the tests' report [78].

The first damage scenario (A) is a side damage case leading to non-symmetric progressive flooding of the barge. The damage (opening SEA-R21S) involves the fore compartment, leading first to the flooding of the starboard wing tank (R21S). In scenario A, the openings SEA-DB1, SEA-DB2, DB2-R21 and R12-R22 are closed, thus, the double bottom is intact. The progressive flooding of the barge lasts about 350 s in scenario A.

The second damage scenario (B) is a bottom damage case driving to down-flooding and flow reversal. In this case, the opening SEA-DB2 is open whereas the openings SEA-DB1 and SEA-R21S are kept close. Besides, all the internal connections are open, leading to the flooding of all the rooms within the barge portable element. In detail, since the floodwater level rises faster in compartment 2 than compartment 1, the room R12 is first flooded by opening R12-R22 leading also, through opening R11-R12, to the down-flooding of R11 accelerating its level rising. As R11 is filled, the flow through opening R11-R12 is reverted. The progressive flooding of the barge lasts about 440 s in scenario B.

The last scenario (C) is again a bottom damage, but insted down-flooding it leads to a quite long and complex flooding chain. The breach is the opening SEA-DB1, while the SEA-DB2 and SEA-R21S are kept close.

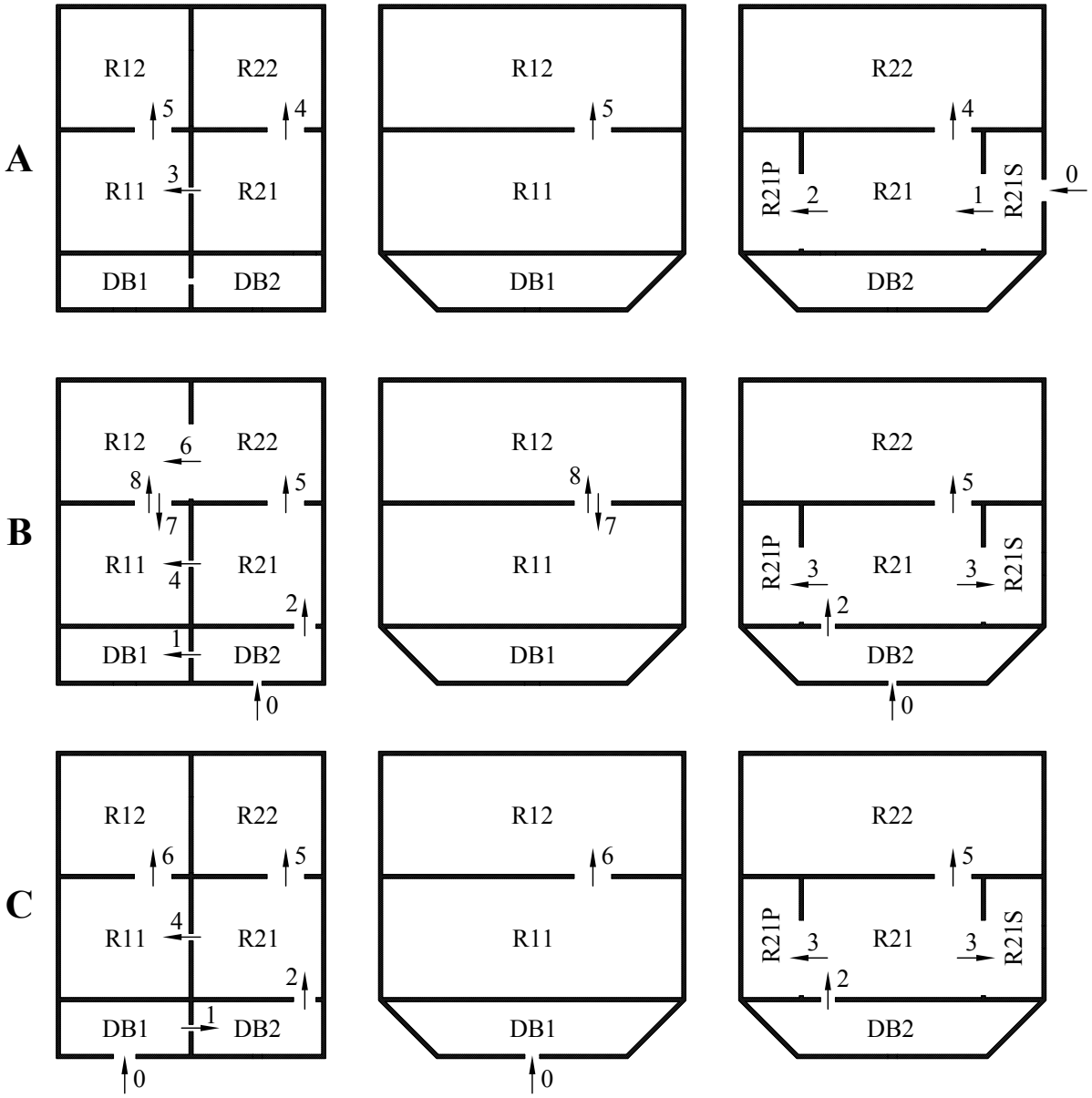

Figure 4.3 Recored flooding sequances in the damage scenarios A, B and C [9]

Also the opening R12-R22 is closed, preventing the down-flooding to occur. In this scenario the rooms DB1, DB2, R21, R11 and R12 crate a flooding sequence being progressively filled. The room R22 and the wing tanks add secondary branches to the main folding chin. The progressive flooding of the barge lasts about 820 s in scenario C.

4.1.3 Calibration of adaptive time step

In order to apply the LDAE formulation, first, the time step adaptation procedure shall be calibrated. It means that a proper value of k_{dt} parameter shall be chosen to assure the desired simulation accuracy. To this end, simulations of the three damage scenarios have been carried

out with different values of k_{dt} ranging between 0.005 and 0.050. The range and intermediate values have been chosen according to previous experiences gained investigating simple geometries (see Appendix B) by applying a multiple constant time steps. To study the effect of k_{dt} on the prediction of the barge floating position in time-domain has been studied. To describe the gap between the experimental and simulated records, an error ratio have been defined as:

$$e_{h_i}(t) = \frac{h(t, k_{dt_i})}{h_e(t)} \tag{4.1}$$

where h is the considered dimension (i.e. heel angle ϕ, trim angle θ or sinkage s) at the instant t simulated assuming the k_{dt_i} value whereas h_e is the experimental moving average evaluated at time instant t. Besides, a dimensional squared error $e^2_{h_i}$ has been also defined to study the results on heel angle in damage scenario A (which is the only non-simmetric one) and the convergence of the floating position at the final stage of flooding. The squared error reads:

$$e^2_{h_i}(t) = [h(t, k_{dt_i}) - h_e(t)]^2 \tag{4.2}$$

Figures figs. 4.4 to 4.6 reports the errors analysis for damage scenarios A, B and C, respectively. Figures figs. 4.7 to 4.9 provides the squared error evaluated at the final equilibrium position of the barge. As a reference, they are compared with the mean squared error *MSE* evaluated on the tail of the experimental record related to the steady state of flooding (see Section 2.1). The *MSE* is defined as:

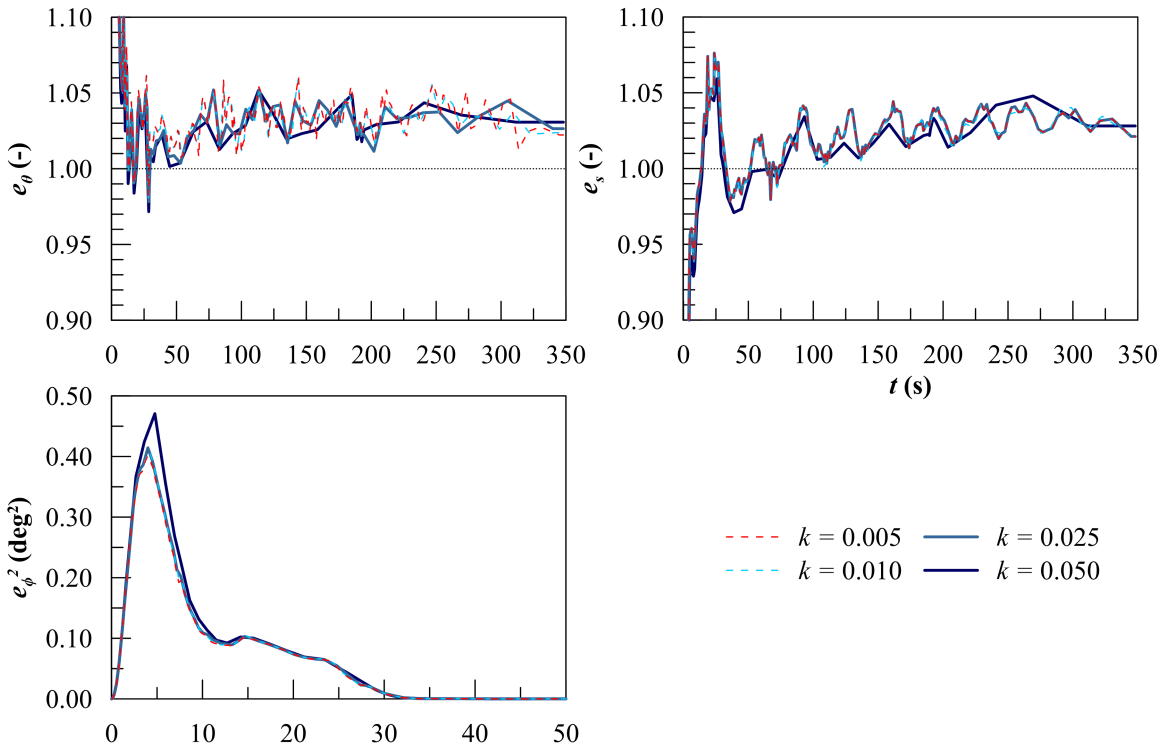

Figure 4.4 Time evolution of the errors evaluated for heel angle, trim angle and sinkage in damage scenario A

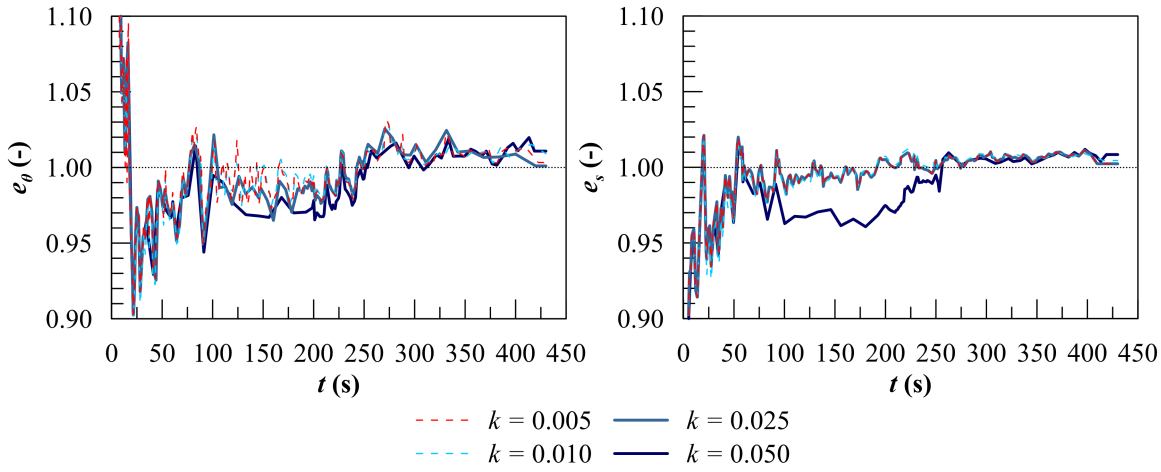

Figure 4.5 Time evolution of the error ratio evaluated for trim angle and sinkage in damage scenario B

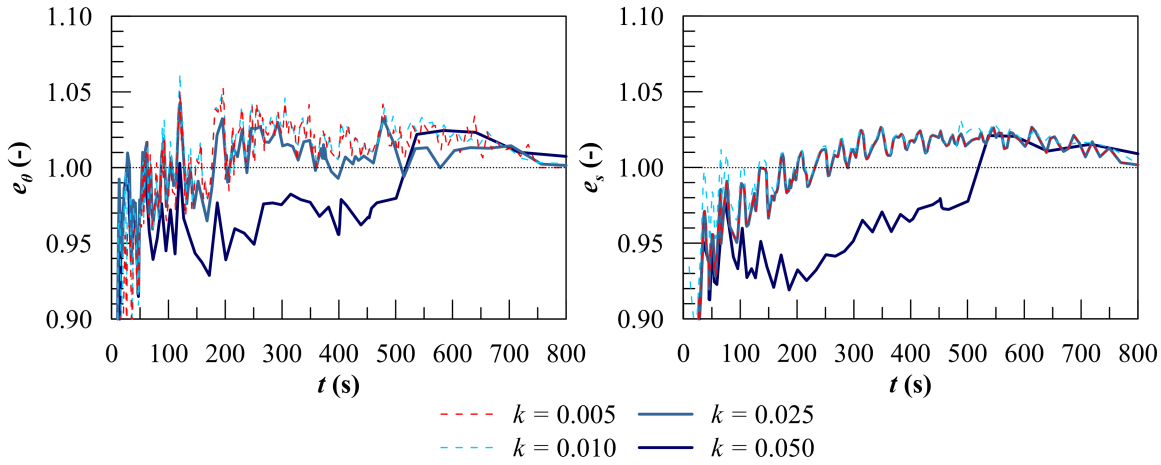

Figure 4.6 Time evolution of the error ratio evaluated for trim angle and sinkage in damage scenario C

$$MSE = \frac{\sum_{i=0}^{n} \left(h_i - \overline{h}\right)^2}{n} \tag{4.3a}$$

$$\overline{h} = \frac{\sum_{i=0}^{n} h_i}{n} \tag{4.3b}$$

where h_i is the i-th experimental data point among the n data point related to the final steady state. For the analysed damage scenarios related to the box-shape barge, it can be observed that the floating position at steady state converges to the experimental values. In more detail, the squared errors on the main floating position parameters are always lower than the *MSE* evaluated on the experimental record tail for $k_{dt} = 0.005$ and $k_{dt} = 0.010$. Considering the complete progressive flooding process in time domain, the error ratios related to trim angle and sinkage are lower than 0.05 in all the three studied scenarios.

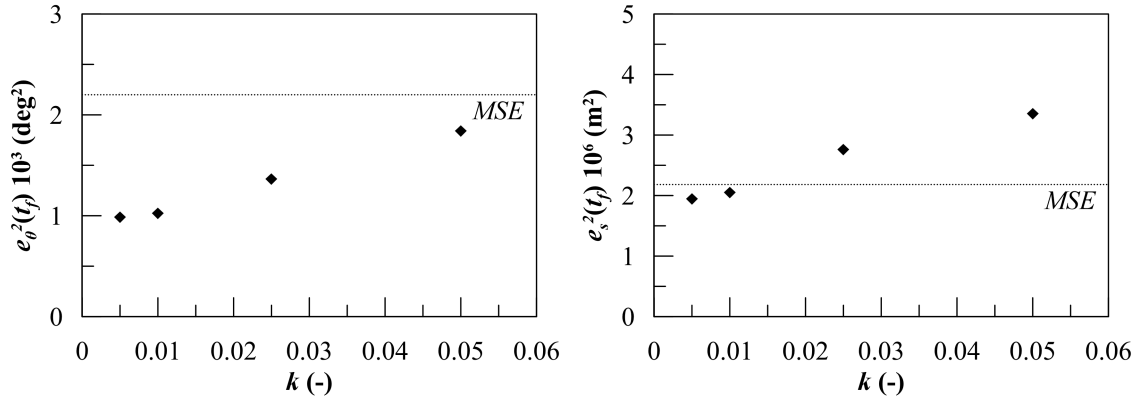

Figure 4.7 Squared error evaluated at the steady state for trim angle and sinkage in damage scenario A

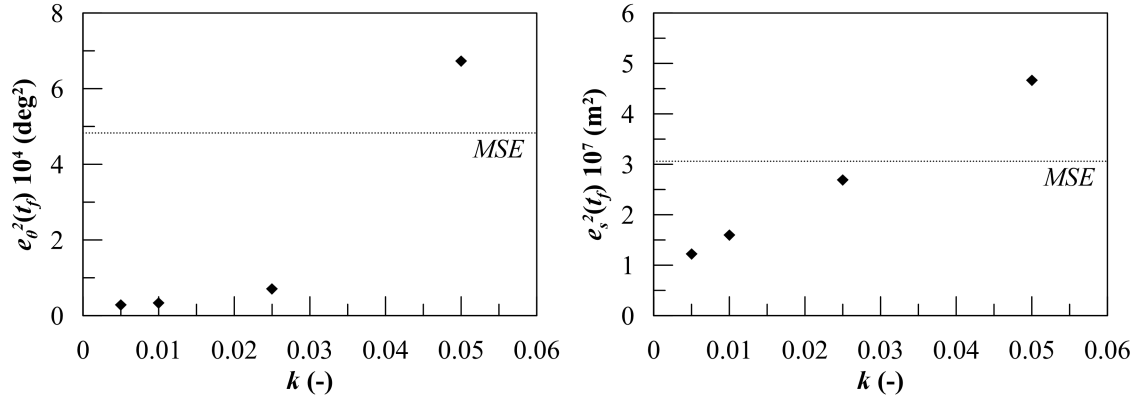

Figure 4.8 Squared error evaluated at the steady state for trim angle and sinkage in damage scenario B

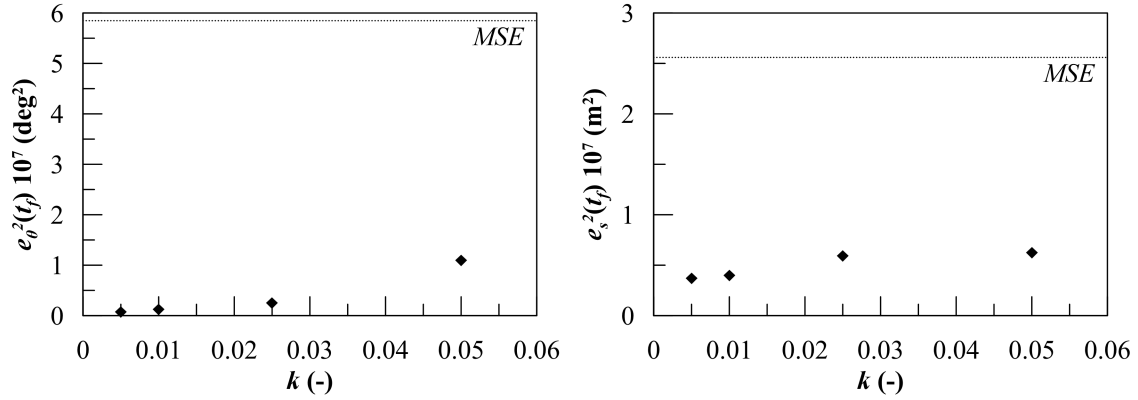

Figure 4.9 Squared error evaluated at the steady state for trim angle and sinkage in damage scenario C

Only at the very beginning of the simulation greater error ratios have been observed. However, they are due to the close to zero values of trim and sinkage rather than modelling issues. Moreover, the error ratios clearly converge to specific value as the $k_{\mathrm{d}t}$ decreases. In general, the adoption of higher $k_{\mathrm{d}t}$ values leads to an increasing delay during the intermediate stages of progressive flooding. Considering $k_{\mathrm{d}t} = 0.050$, trim and sinkage error ratios differ up to 0.04

and 0.08 in the damage scenarios C and D, respectively, whereas more limited differences have been observed in scenario A as well as applying lower k_{dt} values.

Simulations carried out assuming $k_{dt} = 0.005$ and $k_{dt} = 0.010$ has practically the same error ratios trends. Hence, the value $k_{dt} = 0.010$ can be considered a good vlaue for high accuracy simulations that does not present relevant issues in terms of the computational time. In fact, the latter decreases as the k_{dt} parameter increases as shown in Table 4.3. The tabular values relates to the computational performances obtained employing Intel® Core™ i7-4710 CPU (2.50 GHz) standard laptop. Assuming $k_{dt} = 0.025$, the simulations can be carried out in about the 2% of the time-to-flood in model scale, leading to a computational time reduction of over 5 times with respect to the one required by a constant time step simulation having comparable accuracy [8]. In such a case, only a small/moderate drewback on simulation accuracy is accepted, being may times lower than the errors that have been considered satisfactory for the onboard application [84]. Hence, when a fast simulation is required, such as during a flooding emergency, values up to $k_{dt} = 0.025$ can be still employed, being a good compromise between good simulation fidelity and limited computational effort.

Table 4.3 Computation time required by the linearised Differential-Algebraic Equation (LDAE) method for the flooding scenarios A, B and C.

Case	k_{dt}	t_f **(s)**	t_C **(s)**	$\%t_f$ **(%)**
A	0.005	350.0	21.1	6.0
A	0.010	350.0	13.3	3.8
A	0.025	350.0	7.4	2.1
A	0.050	350.0	5.3	1.5
B	0.005	440.0	48.3	11.1
B	0.010	440.0	30.8	7.0
B	0.025	440.0	11.2	2.5
B	0.050	440.0	9.7	2.2
C	0.005	820.0	28.9	3.5
C	0.010	820.0	18.5	2.3
C	0.025	820.0	10.8	1.3
C	0.050	820.0	10.2	1.2

4.1.4 Results and discussion

The simulations carried out applying LDAE formulation with adaptive time step ($k_{dt} = 0.010$) can be compared with the experimental data and with other simulation techniques. Figures figs. 4.10 to 4.12 provide, for damage scenarios A, B and C, respectiviely, the results coming from LDAE simulation method along with the Pressure Correction (PC) technique ones [80]. Moreover, for cases A and B, the comparison can be extended including also the results from a benchmark study on progressive flooding codes where, besides PC, other 5 simulations methods have been tested on the box-shaped barge [103].

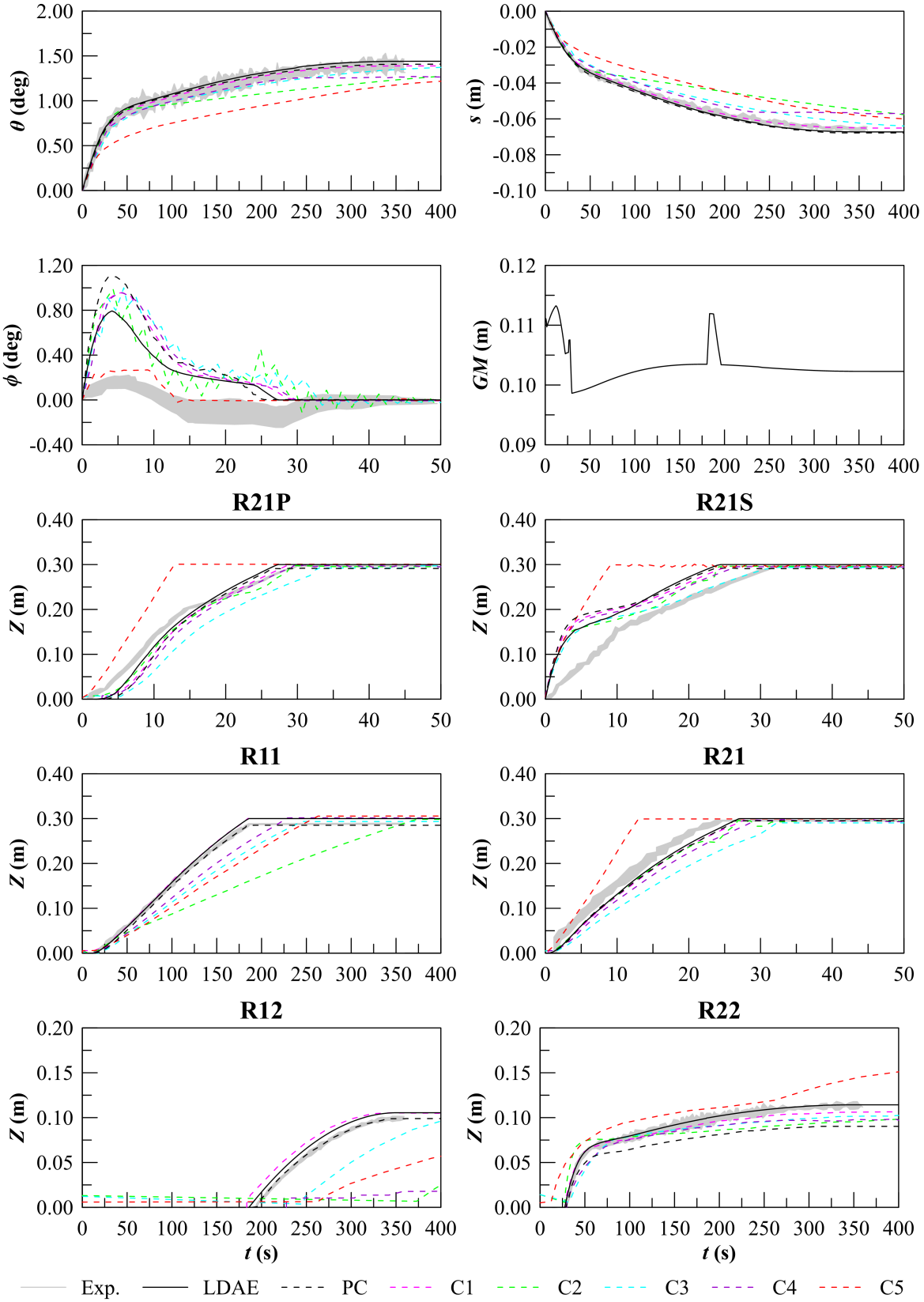

Figure 4.10 Comparison of the experimental records and the results provided by LDAE, PC, C1-C5 techniques for the damage scenario A

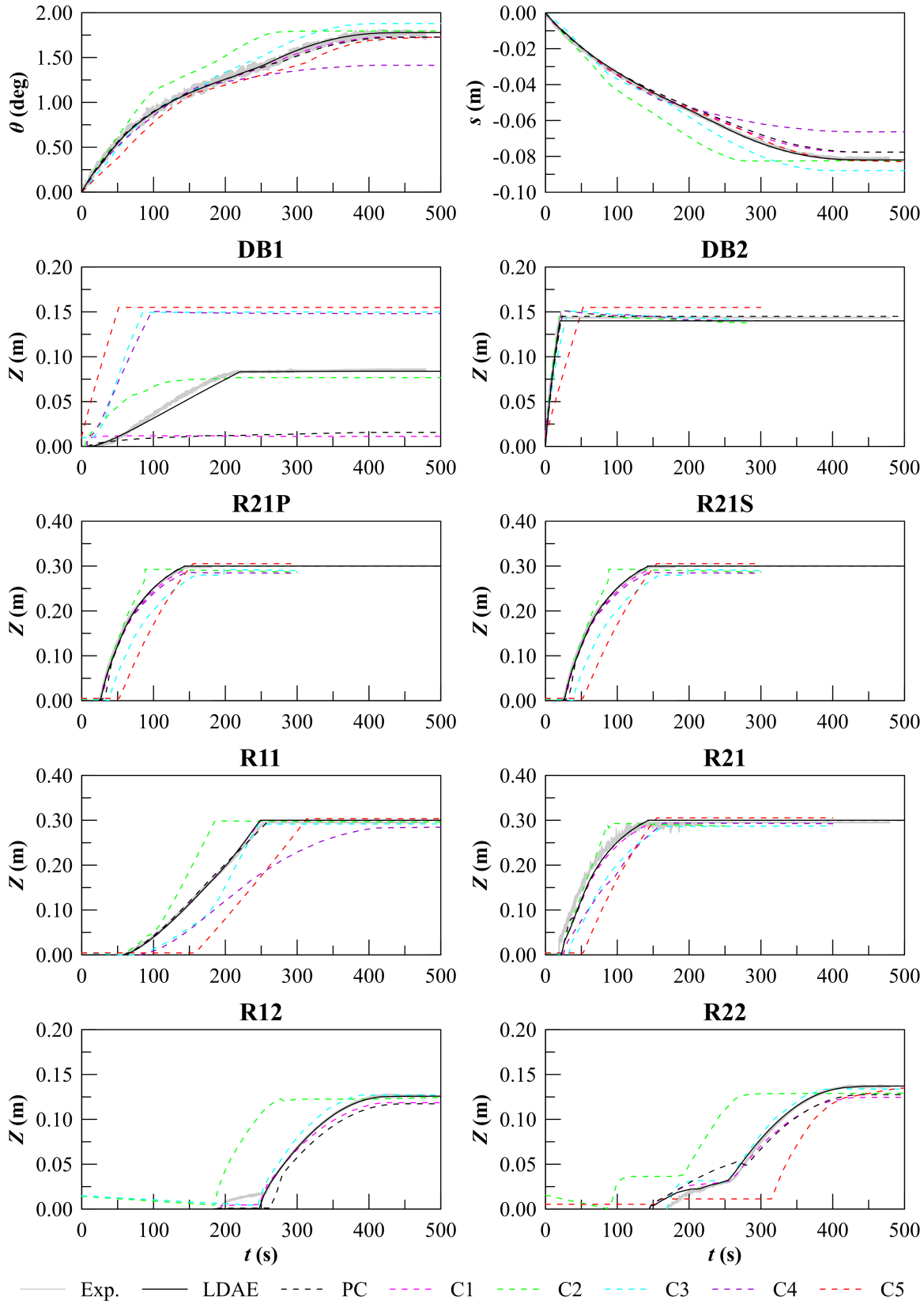

Figure 4.11 Comparison of the experimental records and the results provided by LDAE, PC, C1-C5 techniques for the damage scenario B

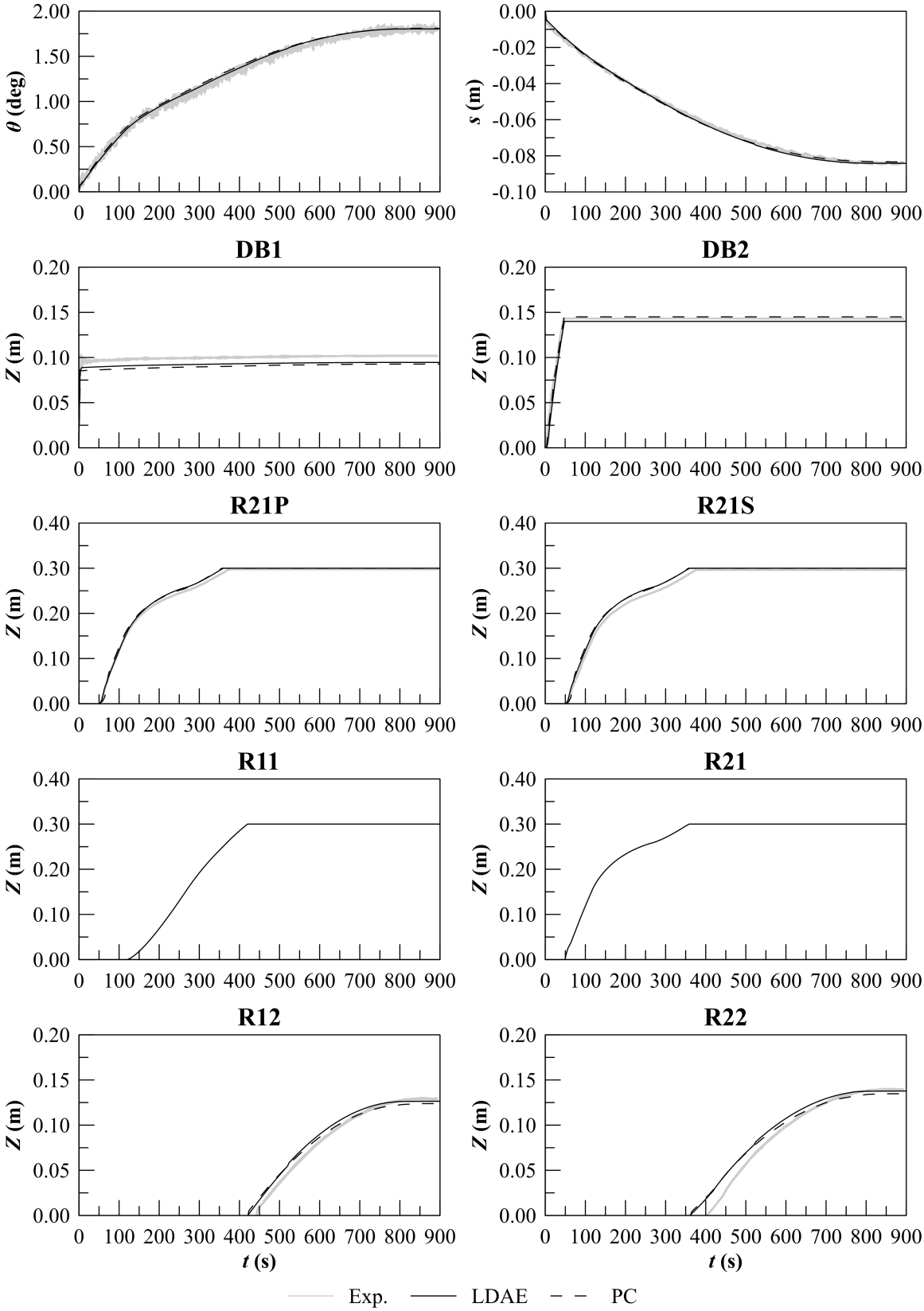

Figure 4.12 Comparison of the experimental records and the results provided by LDAE and PC techniques for the damage scenario C

The results coming from LDAE method are in very good agreement with the PC technique ones in all the damage scenarios A, B and C. Moreover, they are also very close with the experimental records, except for the heel angle in scenario A. In fact, the heel agle shows a small peak during the experiment whereas a larger overshoot is predicted by LDAE simulation. This discrepancy is mainly due to the dynamic behaviour of the barge during the transient flooding phase. Since the adopted method is quasi-static it is non-capable to model the dynamic behaviour in the barge. Nevertheless, the LDAE results on heel angle are in line with the ones provided by all the compared codes (except C5, which, then, is expected to be a dynamic code and appear to be less accurate in the prediction of all the other simulated dimensions).

A final consideration has to be done regarding the other mathematical formulations introduced in Chapter 3. The selected damage scenarios are not suitable for the application of the LODE method since they entail several filled rooms. Regrading LODEG technique, it does not involve particular issues for the box-shaped barge. In fact, the LODEG method has been validated for this damage scenarios and more details can be sought in [8]. However, it shall be noticed that the internal openings in the box shaped barge have quite large difference in size and location. Hence, to highlight the effect of the different mathematical formulations a different test layout is needed.

4.2 Comparison of results in full-scale

Up to now, the proposed simulation techniques have been applied only on very simple geometries or on model-scale experiments that are quite different from a large passenger vessel. The latter is a more challenging environment characterised by more complex internal layouts, non-wall-sided envelope and closable internal openings [6]. Moreover, the closed non-watertight openings can leak or collapse under excessive effective waterhead introducing additional complication into the simulation problem [82]. This is why here the behaviour of the three proposed mathematical formulations is studied on a full-scale geometry to better highlighting the issues that can arise from the adoption of a non-reliable simulation tool. Moreover, the study of the computational performances is also essential on a realistic scenario involving a large passenger vessel, in order to assure the applicability of the proposed methods in full-scale environments.

4.2.1 Test arrangement

The full-scale test case is a large cruise ship having the main particulars reported in Table 4.4. The ship is divided into 21 watertight compartments grouped in 7 main vertical zones. The internal layout of the vessel up to the second deck above the bulkhead deck is shown in Figure 4.13. As done for the model scale barge, the hull and the compartments 14 and 15 (Figure 4.14) have been, here, modelled by means of non-structured triangular meshes up to the bulkhead deck.

Table 4.4 Main particulars of test cruise ship

Description	**Symb.**	**Value**	
Length between perpendiculars	L_{BP}	285.38	m
Length overall	L_{OA}	315.600	m
Breadth	B	36.700	m
Draught	T	8.300	m
Depth	D	11.200	m
Displacing Volume	∇	56351.9	m^3
Displacement	Δ	57760.7	t
Longitudinal centre of mass	LCG	130.380	m
Transversal centre of mass	TCG	0.000	m
Vertical centre of mass	VCG	18.840	m
Metacentric Height	GM	2.109	m
Frame spacing		0.725	m

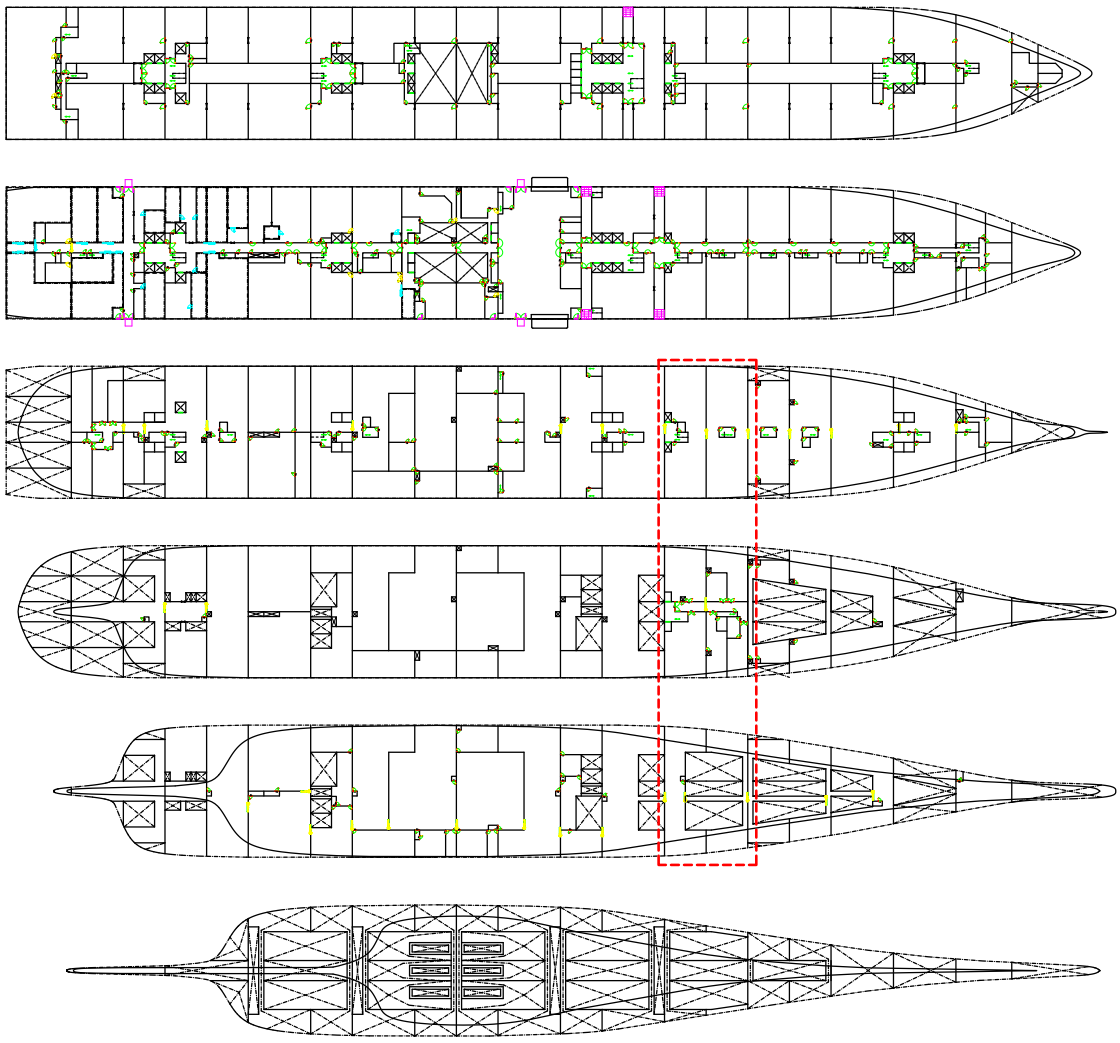

Figure 4.13 General arrangement of the test cruise vessel, highlighting the damaged compartments (red dashed line)

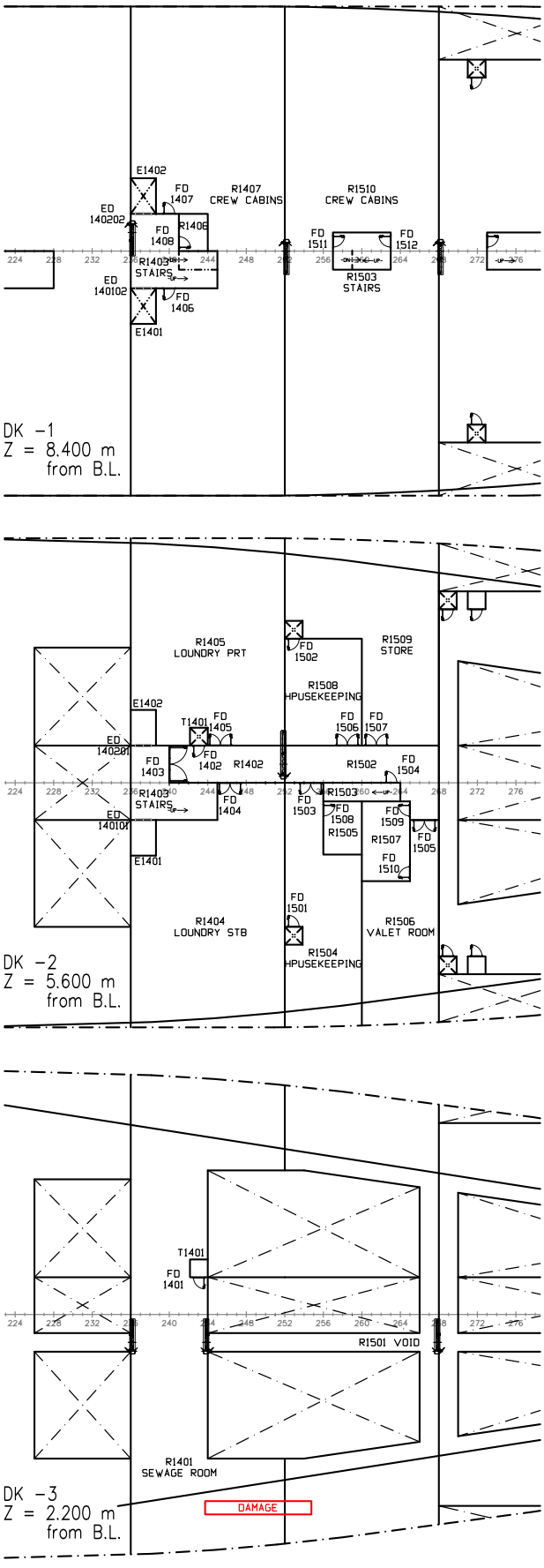

Figure 4.14 Detail of the internal layout of the damaged compartments

Figure 4.15 shows the resulting 3D model of the internal subdivision, whereas Table 4.5 more details regarding the rooms' type, capacity and permeability. Eventually, in Table 4.6 are reported the dimensions and location of all the opening connecting the modelled rooms Opening type refers to the codes reported in Table 2.1, where also related leakage and collapse coefficients are defined.

4.2.2 Flooding scenario

The selected flooding scenario has been chosen in order to check the simulation capabilities of the proposed code in a real full-scale environment. To this end, a medium-size damage has been selected affecting the compartments 14 and 15 as shown in Figure 4.14, having the dimensions specified in Table 4.6. Within the two compartments, a very complex internal subdivision is present under the bulkhead deck. Three decks are there fitted instead of two (as happened in most of the other watertight compartments): the decks are D-3, D-2 and D-1, located at 2.2 m, 5.6 m and 8.4 m above the baseline, respectively.

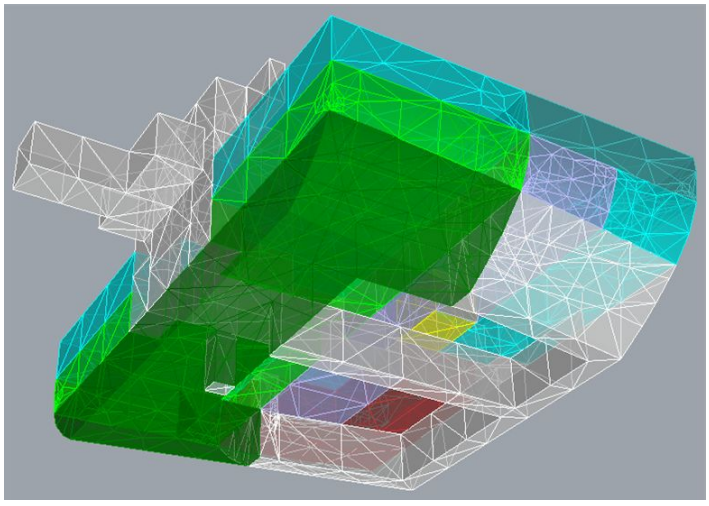

Figure 4.15 3D model of the internal spaces in compartments 14 and 15

Table 4.5 Details regarding the test ship rooms within the compartments 14 and 15

Id	**Type**	V_e (m^3)	μ **(-)**	**Id**	**Type**	V_e (m^3)	μ **(-)**
E1401	Void	55.026	0.95	R1501	Void	476.942	0.95
E1402	Void	55.026	0.95	R1502	Void	105.296	0.95
R1401	Machinery	695.207	0.85	R1503	Void	71.209	0.95
R1402	Void	64.798	0.95	R1504	Accommodations	228.328	0.95
R1403	Void	428.501	0.95	R1505	Machinery	27.608	0.85
R1404	Machinery	445.699	0.85	R1506	Accommodations	180.238	0.95
R1405	Machinery	408.543	0.85	R1507	Accommodations	57.855	0.95
R1406	Stores	10.231	0.60	R1508	Accommodations	124.042	0.95
R1407	Accommodations	1016.614	0.95	R1509	Stores	206.080	0.60
T1401	Void	10.576	0.95	R1510	Accommodations	1100.951	0.95

Table 4.6 Main characteristics of the test ship internal openings within the compartments 14 and 15. $C = (X_C, Y_C, Z_C)$ is the centre of the opening in ship-fixed reference system

Id	**Type**	**Room A**	**Room B**	X_C **(m)**	Y_C **(m)**	Z_C **(m)**	A (m^2)	**Status**
D14	damage	SEA	R1401	179.700	-17.000	4.000	6.000	Open
D15	damage	SEA	R1401	183.700	-17.000	4.000	2.000	Open
ED140101	E	R1403	E1401	172.006	2.800	6.575	1.658	Closed
ED140102	E	R1403	E1401	172.006	2.800	9.375	1.658	Closed
ED140103	E	R1403	E1401	172.006	2.800	12.175	4.290	Closed
ED140104	E	R1403	E1401	172.006	2.800	15.150	3.120	Closed
ED140201	E	R1403	E1402	172.006	-2.800	6.575	3.120	Closed
ED140202	E	R1403	E1402	172.006	-2.800	9.375	1.658	Closed
ED140203	E	R1403	E1402	172.006	-2.800	12.175	1.658	Closed
ED140204	E	R1403	E1402	172.006	-2.800	15.150	1.658	Closed
FD1401	B2 H	R1401	T1401	176.175	-2.800	3.175	2.340	Closed
FD1402	B2 H	R1402	T1401	176.175	-2.800	6.575	2.340	Closed
FD1403	B2 DL	R1402	R1403	174.000	-1.400	6.575	2.340	Closed
FD1404	B2 DL	R1404	R1402	178.713	0.000	6.575	2.340	Closed
FD1405	B2 DL	R1405	R1402	178.350	-2.800	6.575	2.340	Closed
FD1406	B2 H	R1407	R1403	174.000	2.800	9.375	2.340	Open
FD1407	B2 H	R1407	R1403	174.000	-2.800	9.375	2.340	Open
FD1408	B2 H	R1406	R1403	174.725	-0.670	9.375	2.340	Closed
FD1501	B2 H	R1504	R1501	183.425	10.840	6.575	1.658	Closed
FD1502	B2 H	R1508	R1501	183.425	-10.840	6.575	1.658	Closed
FD1503	B2 DL	R1504	R1502	184.513	0.000	6.575	3.120	Closed
FD1504	B2 H	R1502	R1503	190.675	0.000	6.575	1.658	Closed
FD1505	B2 DL	R1506	R1502	193.213	2.800	6.575	3.120	Closed
FD1506	B2 DL	R1508	R1502	187.413	-2.800	6.575	3.120	Closed
FD1507	B2 DL	R1509	R1502	189.588	-2.800	6.575	3.120	Closed
FD1508	B2 H	R1505	R1504	185.600	2.090	6.575	1.658	Closed
FD1509	B2 H	R1507	R1502	192.125	2.090	6.575	1.658	Closed
FD1510	B2 H	R1507	R1506	192.125	6.730	6.575	1.658	Closed
FD1511	B2 H	R1503	R1510	186.325	-0.710	9.375	1.658	Open
FD1512	B2 H	R1503	R1510	190.675	-0.710	9.375	1.658	Open

In compartment 14, the sewage treatment plant is installed at the lower deck, while in compartment 15 there is a void space embracing three freshwater tanks that are, here, considered intact. Above, at deck D-2, the ship's main laundry and housekeeping spaces are fitted in compartment 14 and 15, respectively. As can be noted in Figure 4.14, the internal layout at such a deck is anything but straightforward, having a dense non-watertight subdivision composed of spaces divided by steel boundaries. The upper deck spaces, in both the selected compartments, are allocated to crew accommodations. Due to the low restriction to flooding that is typical of cabins doors and boundaries [90], these spaces have not been modelled in deep detail. Hence, only the steel-bounded spaces have been included in the ship model.

The damaged compartments are located in the ship's forebody, characterised by heavily non-wall-sided external boundaries and leading to trim the ship by the bow. Moreover, most of

the internal boundaries are strongly non-symmetric (in terms of both geometry and/or permeability), hence, relevant heeling angles are also expected. Besides, most of the non-watertight openings are initially closed (see Table 4.6), requiring the proper modelling of their leakage and collapse behaviour. Therefore, the selected damage scenario can be considered very challenging for a progressive flooding simulation code and this is why has been, here, studied among all the other possible options.

4.2.3 Results and discussion

The damage scenario has been simulated in the time domain having an overall duration of about four hours and, eventually, the ship reaches a new equilibrium floating position. The simulated flooding scenario is characterised by a fast pace during the first 20 minutes (1200 s) and very slow propagation of floodwater on the upper deck due to the closed doors FD1408 and FD1504, which withstands the waterhead without collapsing during the whole process. Figure 4.16 shows the time evolution of heel angle, trim angle, sinkage and metacentric height during the initial phase of progressive flooding simulated applying the three methodologies presented in Chapter 3.2: LODE, LODEG and LDAE. Figure 4.17 presents the floodwater levels assessed with the three methods in two rooms that are particularly critical in the selected scenario: the sewage room R1401 and its escape trunk T1401.

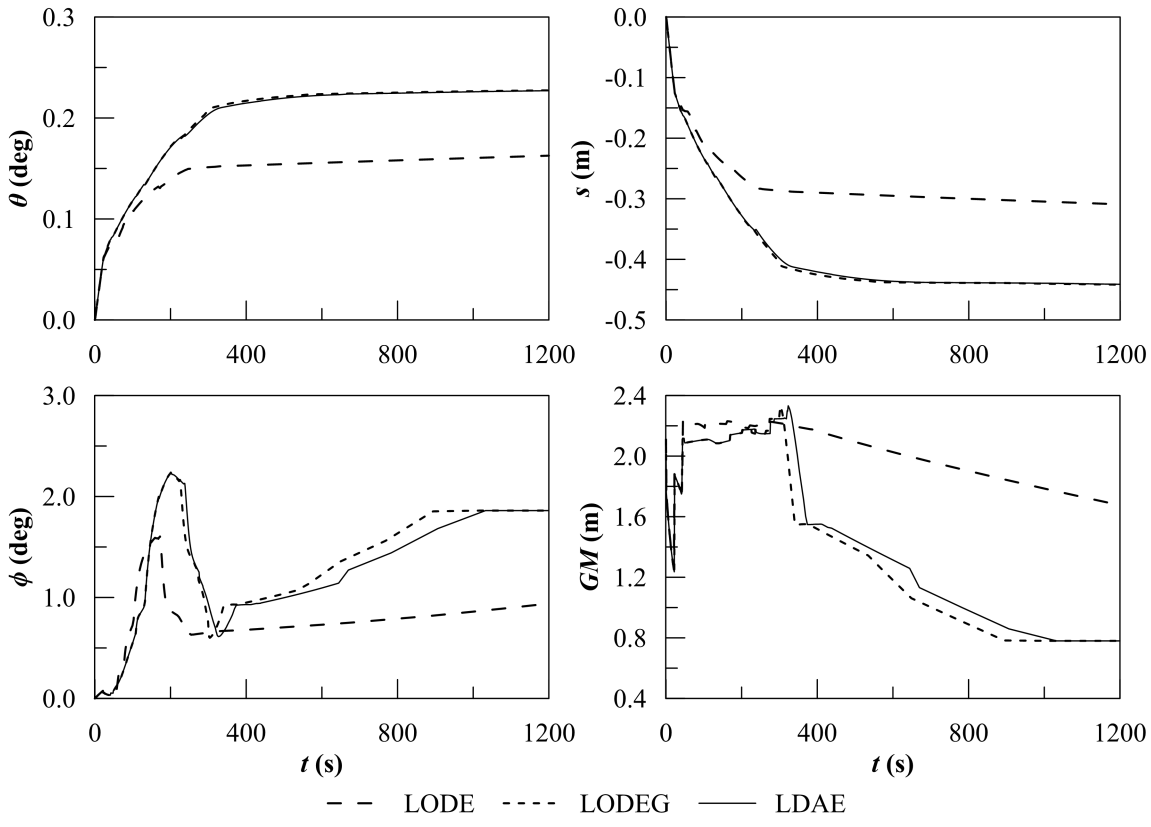

Figure 4.16 Evolution of the floating position of the test ship in the studied damage scenario

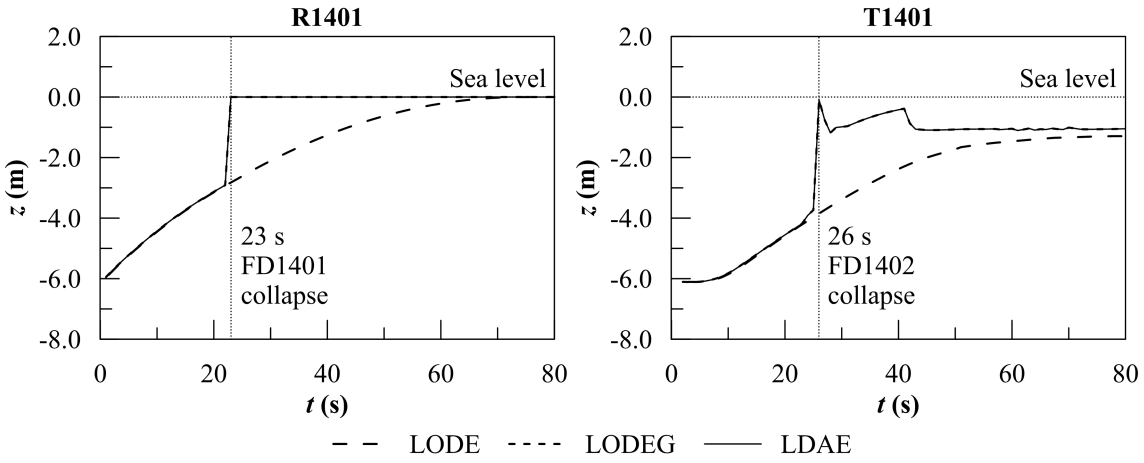

Figure 4.17 Floodwater levels inside the rooms R1401 and T1401 in the very initial phases of flooding in the earth-fixed reference system

In Table 4.7 the main events that occurred during the simulation are reported, including the occurrence time assessed with the three methodologies.

It is worth to notice that the LDAE and LODEG methods provide a similar trend, whereas the LODE method shows a completely different time prediction. This is caused by the too simplistic assumptions: applying the LODE formulation when the room R1401 is filled, the conservation of mass is no more fulfilled. It means that the waterhead in the room rises with the same slope observed in the partially filled condition, instead of reaching almost the seawater level in about 2 s, as can be observed in the LODEG and LDAE simulations (Figure 4.17). In general, when the waterplane area S_i is assumed to be equal to the top area of a completely filled room, a delay is experienced applying LODE formulation. This drives to an optimistic estimation of the time-to-flood, which is not acceptable when the code is used for support decisions onboard during an emergency. Besides, in the studied test case, the LODE method presents an even worst issue: it fails in recognising the collapse of the fire door FD1401, which connects the damaged room R1401 to the escape trunk T1401. The previously discussed slower rising of floodwater in R1401 avoids the overtaking of the collapse effective waterhead (2.5 m for the FD1401). The withstanding door leads to a complete distortion of the simulation in all the subsequent flooding phases. In particular, the time-to-flood is severely overestimated, being more than 12 hours (three times the values estimated with the other two formulations). Such values are not completely ascribable to the FD1401 withstanding, since the collapse of several other openings on the D-2 does not occur too. Nevertheless, in the selected damage scenario, the LODE method is not capable to obtain either a qualitative simulation of the progressive flooding process.

Hereinafter focus will be made only on the LODEG and LDAE formulation. As mentioned, the two techniques provide fully comparable results at the beginning of progressive flooding process as can be observed in Figure 4.18. Both the methods are capable to model properly the FD1401 collapse and predict qualitatively the same time evolution.

Table 4.7 Details regarding the test ship rooms within the compartments 14 and 15

Id	**event**	t **(s)**		
		LODE	**LODEG**	**LDAE**
FD1401	leakage	1	1	1
FD1401	collapse	n.a.	23	23
FD1402	leakage	36	26	26
FD1402	collapse	36	26	26
FD1405	leakage	62	27	27
FD1404	leakage	60	27	27
FD1403	leakage	56	27	27
ED140101	leakage	114	38	38
ED140201	leakage	186	39	39
FD1405	collapse	n.a.	41	41
FD1404	collapse	n.a.	41	41
FD1403	collapse	n.a.	41	41
FD1501	leakage	45	44	44
FD1502	leakage	45	44	44
FD1501	collapse	50	48	48
FD1502	collapse	50	48	48
FD1503	leakage	59	50	50
FD1508	leakage	59	50	50
FD1506	leakage	63	50	50
FD1505	leakage	62	70	70
FD1509	leakage	75	71	71
FD1504	leakage	75	72	72
FD1507	leakage	77	74	74
FD1506	collapse	78	108	108
FD1510	leakage	92	116	116
FD1505	collapse	99	129	129
FD1507	collapse	168	223	235
FD1406	open	15086	294	312
ED140102	leakage	n.a.	294	313
FD1408	leakage	16453	299	317
FD1407	open	15086	301	321
ED140202	leakage	n.a.	301	322
FD1512	open	n.a.	374	410

However, a small delay can be noted in LDAE prediction after 200 s. The delay is again ascribable to room-filling. In fact, in LODEG method the filled rooms are grouped, leading to discarding the pressure losses on the opening connecting the grouped rooms. In the studied scenario, as R1504 is filled, it is grouped with the room R1501. The same happened to R1508. A similar situation is observed also in CMP14 at about 270 s, where again LODEG pace is faster than LDAE one. Such a distortion, yet contained, is far reduced as the level in grouped rooms approaches the sea free surface. The reason can be found in the pressure loss that becomes again negligible, in the final pahases of progressive flooding. In fact, the pace is mainly due to the

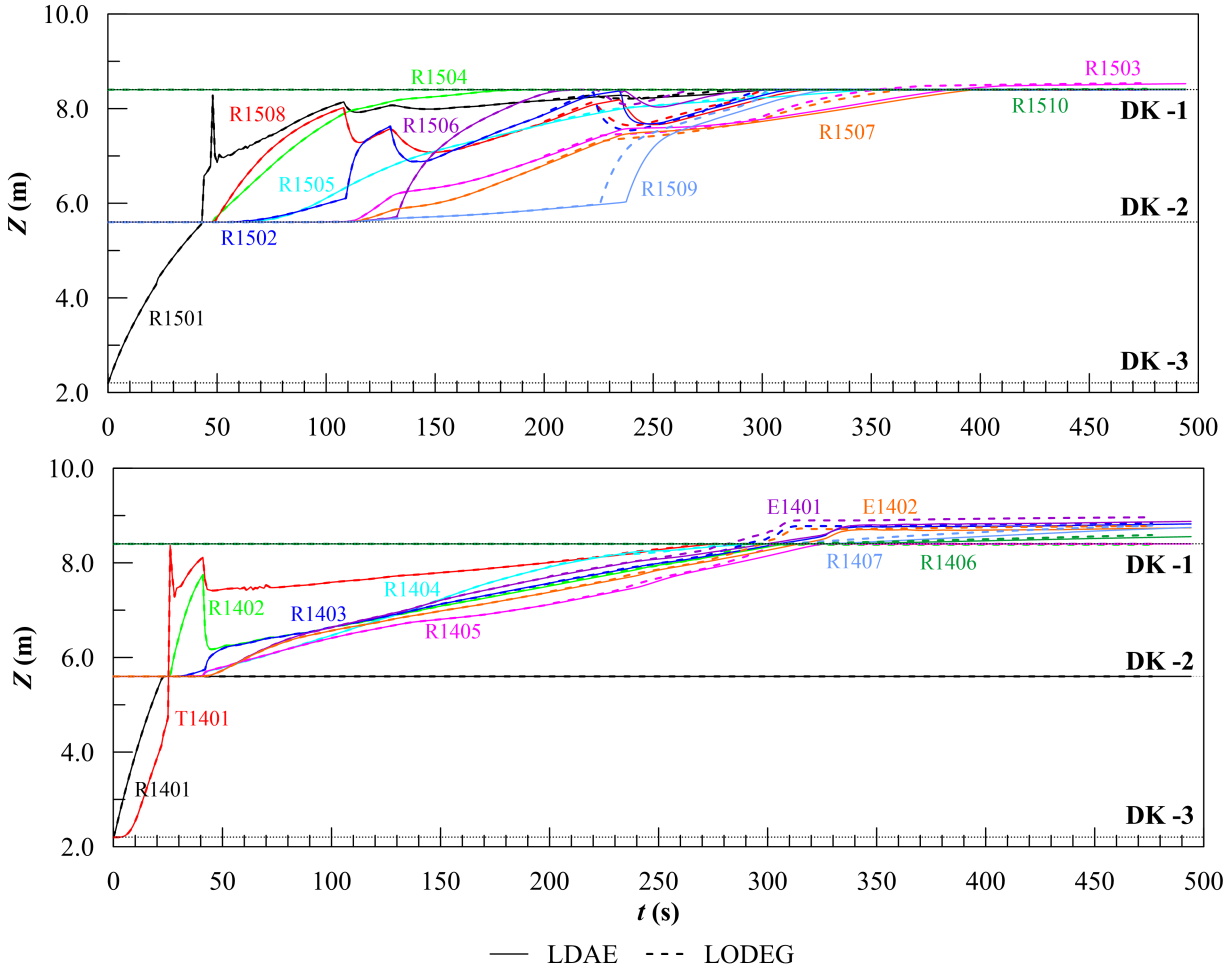

Figure 4.18 Floodwater level inside the compartments 14 and 15 in ship-fixed reference system

already mentioned closed doors FD1407, FD1406 and FD1504 and their higher pressure losses compared to the previous openings in the flooding chain, which are all collapsed. Since the discarded pressure losses related to collapsed openings are negligible compared to the closed ones, a comparable slope of all the level curves obtained applying LODEG and LDAE methods are then re-established until the simulation end.

However, the choice of the LODEG or the LDAE method has a very limited impact on the floating position and initial stability prediction in the studied test case, as shown in Figure 4.16. In particular, they could be negligible when compared to the uncertainties that might affect the definition of the discharge coefficients [77], the room permeabilities, the initial loading condition [7] or the critical waterheads that usually are only roughly estimated in a real operative environment. Moreover, the time-to-flood underestimation coming from the application of LODEG method is far more acceptable for decision support purposes. Although all the previous considerations, LDAE formulation should be preferred to the LODEG one for full-scale simulations, especially if non-watertight openings are present. This statement is far reinforced by the very small drawback in terms of computational time per cycle introduced by the LDAE

application, which is mainly due to the more complex set up of matrices [8]

In conclusions, the overall computational effort required by the most reliable methodology shall be commented on. The LDAE simulation has been carried out with adaptive time step assuming a $k_{dt} = 0.005$, resulting in a computation time $t_C = 117$ s employing an Intel® Core™ i7-4710 CPU (2.50 GHz) laptop. Among the total computation time, the 44% relates to the equilibrium assessment, the 38% to the evaluation of the initial floodwater levels from volumes at the equilibrium floating position, whereas the 15.5% to the assessment of floodwater volumes from the new levels. The solution of the linearised problem takes less than 2.5% of the computation time. Hence, large improvements are still achievable by applying faster methods to compute the hydrostatics: now the pressure integration technique is applied [96], being several times slower than methods based on section area integration. Nevertheless, the computation time is already less than the 1% of the overall duration of the studied damage case, which can be considered appropriate for onboard application.

Chapter 5

Application of Machine Learning

In the present chapter, the theoretical background is presented related to the prediction of the main damage consequences from the evolution of the damaged ship floating position. The relation between them is established by applying Machine Learning driven by datasets of progressive flooding simulations. Hence, after a brief introduction of the main concepts and terminology used in the machine learning field, the specific problems addressed in this work and the overall process applicable in a new generation of onboard emergency DSSs are described. Then, focus is made on machine learning algorithms and different database generation algorithms that have been tested in the present work.

5.1 Introduction to Machine Learning

Machine learning refers to statistical techniques giving a computer system the capability to "learn" on the base of previous examples, without being explicitly programmed [110]. The term was introduced by Arthur Samuel in 1959 referring to the solution of checkers game by machine. Machine Learning gives a computer program the ability to reproduce a behaviour that has not been programmed by a human being and, thus, might show behaviours that cannot be foreseen by the author. Machine learning is based on three pillars [57]: the availability of a large amount of data, a metric to quantify error/distance of the current behaviour from the ideal one and a feedback mechanism employing the metric to produce better behaviour. Machine Learning algorithms can be classified into four main learning models [75]:

- *Supervised Learning*: where a dataset is given consisting of labelled data. In other words, including the right answer, usually called response or label or, more traditionally, dependent variable(s), corresponding to each data point (or item). Each data point is defined by the so-called predictors, also known as features or, more traditionally, independent variable(s). predictors define the so-called input space of the learner. Hence, a supervised learning algorithm shall map the key characteristics of an item to the right answer. Then, given a new data point, it shall properly predict the unknown answer. Most common tasks

addressed by supervised learners are [68]:

  - *Classification*: where the response is categorical;
  - *Regression*: where the response is a real value;

- *Unsupervised Learning*: where the learner receives only unlabelled data. It means that the learner does not know the right answer and autonomously makes predictions for all unseen data. Typical tasks addressed by unsupervised learners are [68]:

  - *Clustering*: aiming at partitioning a set of data points into homogeneous subsets;
  - *Dimensionality reduction*: also known as *manifold learning*, aims to transform an initial representation of data-point (a large set of features) into a lower-dimensional one while preserving some of the properties of the dataset;

- *Semi-supervised Learning*: where the dataset includes both labelled and unlabelled data; it is usually employed where responses are for some reasons not available for the complete dataset or if they are too expansive to obtain. Hence, in semi-supervised learning, the unlabelled data hopefully help to achieve better accuracy than the one attainable with the labelled data only.

- *Reinforcement Learning*: where the training is performed separately from the previous models, involving feedback from the external environment. In reinforcement learning, the model parameters are constantly updated to maximise the reward and minimising the penalty coming from an action, which is the output generated by the model [56].

Machine learning algorithms can be further classified into three types, according to the estimation method [3]:

- *Parametric estimation*: when a model is defined over the whole input space (defined by predictors), learning its parameters from all training dataset;

- *Nonparametric estimation*: when the model splits the input space into local regions and, given a new data point, the corresponding local model is used. Each local model is defined employing the training data available in the local region;

- *Instance-based estimation*: where the model, given a new item, evaluates the distance from all the items in the training dataset.

For each of the main classes defined above a large set of algorithms and solutions have been developed. However, a more detailed literature review is not provided in the present work, which aims to study a specific set of problems where only some of the machine learning algorithms are applicable.

5.2 Prediction of damage consequences

Damage due to collision or grounding drive to change the ship floating position due to the amount of floodwater loaded onboard. As mentioned in Chapter 2, after the possible dynamic transient, the process can be considered quasi-static. Thus the spreading of floodwater through the internal non-watertight openings can be easily predicted employing fast simulation codes since the flowrates are governed by the well-known hydraulic laws. As shown in Chapter 4, employing the LDAE method presented in Chapter 3 it is possible to cope with main challenges related to progressive flooding simulations in model-scale as well as full-scale. The stability of the damaged vessel can be checked during the progressive flooding identifying the outcome of the damage scenario. It means, whether the ship will survive with a sufficient residual degree of safety or whether she will sink or capsize due to insufficient buoyancy or residual stability, respectively.

Therefore, given a specific damage scenario, a predictable evolution of the floating position and main progressive flooding consequences is defined and can be simulated in a design environment as well as during an emergency [84]. Focusing on the design phase, employing a quasi-static code large databases of time-domain simulations of the progressive flooding can be built in a reasonable time. Exploiting a database of progressive flooding simulations, a link can be searched between the time evolution of the floating position following a defined damage scenario and its main consequences [102]. The direct comparison of the current evolution of the floating position with pre-calculated simulations has been found unreasonably time-consuming for onboard application. Due to the nature of the problem (involving a large amount of labelled data), the application of supervised machine learning techniques has been proposed [13]. Hereinafter, this methodology is presented focusing on the main progressive flooding consequences and the possible application to improve onboard decision support. Finally, the metrics to evaluate the reliability of the proposed process are described.

5.2.1 Studied problems

During the progressive flooding of a damaged ship, some information is essential to support the master and crew decisions. This is even more important on a large passenger vessel where, usually, thousands of human lives are at stake. In this work, a basic set of data has been selected to provide the bridge with the essential knowledge about progressive flooding consequences and enable proper reaction based on evidence, instead of impressions. Moreover, due to the low computational power required by trained learners, the main flooding consequences can be immediately available, having a positive impact also on the emergency reaction time of the crew. In this context, the three most important consequences have been identified as follows:

1. *Final Fate* (FF),
2. *Flooded Compartments* (FC),

3. *Time-to-flood* (t_f).

The ship final fate is the most important information during a flooding emergency since provides information about the ship survival [63]. To apply machine learning, the final fate can be organised into classes describing the outcome of the damage scenario. Here, instead of a binary problem (survival/non-survival), a more complex classification has been proposed in order to provide some additional information about the cause of non-survival scenarios. Hence, the following classes have been defined:

- *Equilibrium* (E): this class belongs to survival scenarios, in which the ship reaches a new safe equilibrium floating position after the progressive flooding process;
- *Capsize* (C): this is a non-survival case, in which the ship exhausts the stability reserve;
- *Sink* (S): this is a non-survival case, in which the ship exhausts the buoyancy reserve. It is worth to notice that this type of failure does not likely happen on large cruise vessels, more susceptible to capsize;
- *Excessive heeling* (EH): this is also considered a non-survival case. In fact, if the ship reaches too large heeling angles during the progressive flooding, she cannot be considered safe any more [51]. This fact is confirmed by the great attention given to counter ballasting [62, 11] especially in the naval framework [43, 67, 20, 26];
- *Time Exceeded* (TE): in a real environment, some damage scenarios drive to very large values of time-to-flood. For instance, the ones having a very small area or involving non-watertight doors that withstand the effective waterhead. In such cases, to contain the computational time during database generation, a maximum threshold on time-to-flood shall be defined. Hence, all damage scenarios lasting more than the threshold are classified as TE. From a DSS perspective, the impact of such is limited by the presence of the bilge system and by the large time anyway available before reaching an unknown condition.

Another essential information during the progressive flooding is the identification of flooded watertight compartments [109]. In fact, this knowledge is essential for many reasons: first to limit/avoid the floodwater spreading to intact watertight compartments. Furthermore, for ships compliant with the Safe Return to Port (SRtP) rules [46], the extension of damage determines whether the ship has to be abandoned or the main essential functionalities can be recovered. To take such a decision flooded compartments shall be known as soon as possible to reduce the reaction time. The identification of flooded compartments can be again considered a classification problem, where the classes are all the possible sets of damaged compartments involved by every potential progressive flooding process.

Finally, the time-to-flood is the last information claimed as essential to support master decision during the progressive flooding of a damaged ship [87], in particular for non-survival

damage scenarios. In such cases, the ship shall be evacuated and abandoned. Rushing evacuation might lead to panic, increasing the risk for passengers and crew. On the other side, as was clearly shown by the Costa Concordia accident, the waste of time when ship abandonment is required might have more severe consequences and lead to the loss of lives. Therefore, to properly manage evacuation, knowing the time frame of the events is of the utmost importance. Up to now, only systems based on flooding sensors fitted in each space of the ship, or at least in some of them [59], have been proved capable to estimate time-to-flood [89, 109]. Here, the estimation of the time to flood is performed without flooding sensors. Since the time-to-flood is a numeral its assessment is modelled as a regression problem.

Usually, if the damaged ship survives, the new equilibrium floating position is reached with a decreasing pace. At a first approximation, the trend can be modelled as limited exponential [8]. Therefore, the accurate definition of t_f is not an easy task for survival scenarios. In some preliminary evaluations was observed that the large uncertainty connected to survival damage scenarios heavily affects the performances of the regression on t_f. On the contrary, in non-survival scenarios, a well defined time instant can be identified, at which the ship sinks, capsizes or reaches an unsafe position. Hence, the time-to-flood can be easily identified reducing the drawback on the regression accuracy. Therefore, also considering that the time-to-flood is not essential in survival cases, only the lost-ship scenarios (S, C, EH) have been considered studying the third problem.

5.2.2 Proposed process

In the present work, the three main problems presented above are addressed by applying machine learning techniques on the base of the time evolution of the floating position of the damaged ship. The floating position of the damaged ship is univocally defined by three parameters: heel angle ϕ, trim angle θ and sinkage s. The proposed process, sketched in Figure 5.1, can be split into two main phases:

1. *Offline preparation*: in such a phase the data required by the onboard system are generated and the learners are trained,

2. *Onboard DSS*: in such a phase, the learners are used onboard to forecast the main flooding consequences. Since at the moment, no onboard software has been implemented, the proposed method will be tested by means of progressive flooding simulations.

The main purpose of the first phase is to generate the training database of progressive flooding simulation. First, the damage cases are defined according to a generation algorithm. Multiple options to generate training databases will be discussed later in detail. For each damage case, the progressive flooding is simulated employing the LDAE method presented in Chapter 3. Since this work aims to forecast the main damage consequences for onboard decision support, the application of a quasi-static method is deemed sufficient. In fact, there is no interest in

a detailed simulation of damages leading to immediate capsize during the dynamic transient, where the time is anyway insufficient to apply any countermeasure for assuring ship survival. Moreover, the application of fast simulation methods helps to reduce the computational effort required to build the databases. Hence, it can ease the configuration of the onboard system.

Besides the training database employed to train the learners for the onboard system, here, an additional database of progressive flooding simulations has been defined for validation purposes. The validation database shall be independently generated from the training one and shall be as far as possible representative of the damages which might occur in a real environment (comprising the probability distribution of their main dimensions and location).

The validation database is then employed to mimic and test the behaviour of the possible new onboard DSS, providing the predictors (time-domain evolution of the floating position) as well as the right classes for classification problems (i.e. FF, FC) and the observed value for the regression problem (i.e. t_f). Considering a defined time instant t^* after the damage occurrence, the evolution of the time evolution of the floating position of the damaged vessel is known up to t^*. Assuming that it has been recorded and the records have been filtered to remove noise [39], the values of ϕ, θ and s taken at defined time instants can be utilized as predictors of three supervised learners, each one responding to one of the previously stated problems. Hence, as time goes by, triplets (ϕ, θ, s) can be added to the predictors set and three new learners, trained considering the extend predictor set, can be employed to update the system response. In the present work, the floating position is assumed to be recorded assuming a constant time step dt and values are retrieved from the validation database for each single damage case.

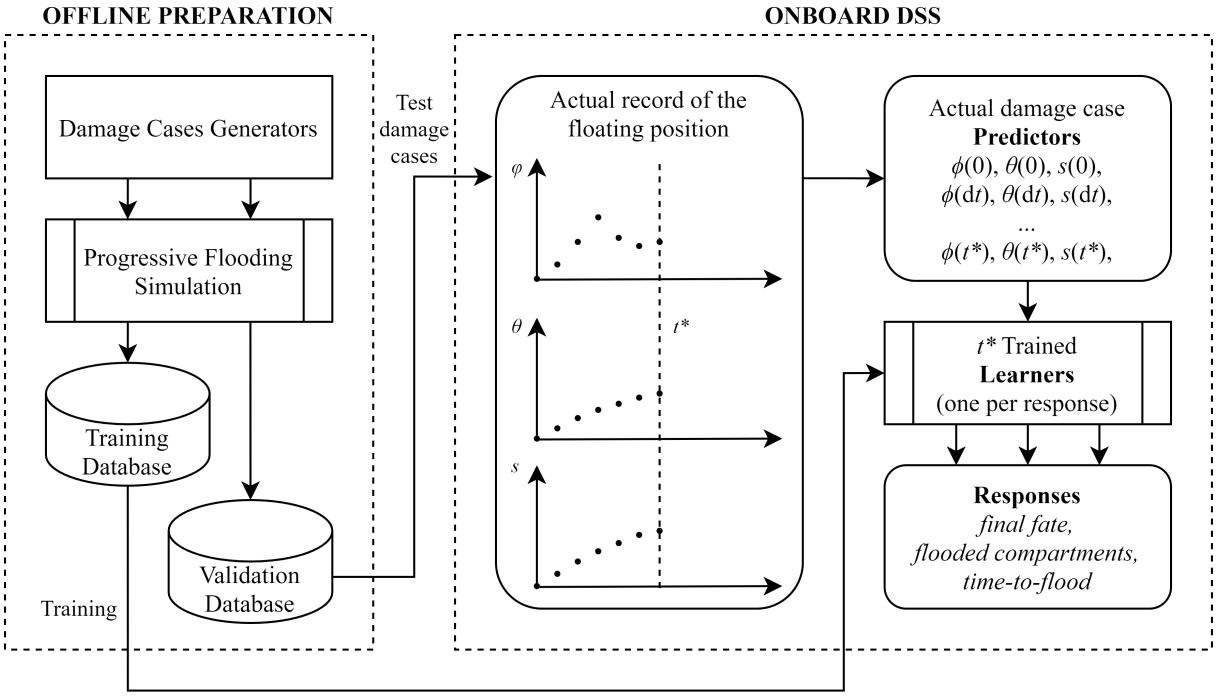

Figure 5.1 Flowchart showing the proposed process to prepare data and explit them to provide onboard decision support

5.2.3 Performance metrics

As mentioned the performances of the proposed methodology to predict the progressive flooding consequences is tested employing a properly large validation database. Hereinafter, the metrics used to measure these performances are presented, distinguishing between the classification and regression tasks.

The accuracy rate of a classification task is usually defined as the capability to assign a specific item, i.e. a damage scenario from the validation database, to the correct feature, i.e. the correct class [42]. Namely, considering a specific classification learner related to the time instant t^*, its accuracy can be defined as:

$$Acc(t) = 100\frac{N_c}{N} \tag{5.1}$$

where N_c is the number of the correctly classified damage scenarios and N the total number of the scenarios induced in the validation database.

To focus on the forecast capability of the model, a so-called ongoing accuracy $Acc^*(N^*)$ has been also defined, considering only the damage cases having t_f greater than t^*. It means, considering only the scenarios that at that point have not already reached the final stage:

$$Acc^*(N^*) = 100\frac{N_c^*}{N^*} \tag{5.2}$$

where N_c^* is the number of the correctly classified ongoing damage scenarios and N^* the total number of the ongoing scenarios induced in the validation database. Furthermore, besides the previously defined metrics, also the confusion matrices have been here employed to analyse in more detail the classification results at each time instant t^*.

Regarding the regression task, several statistical indicators are available in the literature to assess their performances [24]. In the present work, the coefficient of determination $R^2(t)$ has been adopted:

$$R^2(t) = 1 - \frac{SSE(t)}{SS_{tot}} \tag{5.3a}$$

$$SSE(t) = \sum_{i=1}^{N} \left(y_i - y_i'\right)^2 \tag{5.3b}$$

$$SS_{tot} = \sum_{i=1}^{N} (y_i - \bar{y})^2 \tag{5.3c}$$

where y_i are the observed values (known responses), $\bar{y}$ their mean values and y_i' the responses predicted by the model. similarly to classification tasks, an ongoing coefficient of determination $R^{2*}(N^*)$ has been defined based only on the N^* ongoing damage scenarios. Besides, the predicted-observed plot has been employed to analyse in more detail the regression results at

each time instant t^*.

5.3 Tested machine learning algorithms

In the present section, the ML algorithms employed in this study are presented. Due to the nature of the studied problems, having a dataset including both the predictors and related right responses, focus has been made on supervised algorithms only, as already mentioned. Moreover, since both classification and regression tasks shall be addressed, learners that can be employed in both of them have been preferred rather than ones applicable to a single task, e.g. linear regressions. Among the most popular supervised machine learning algorithms the following families has been then considered in the present work:

- *Decision Trees* (DT),
- *K-Nearest Neighbours* (KNN),
- *Support Vector Machines* (SVM)

It is worth noticing that another very popular family of algorithms has been excluded: the probabilistic learners, e.g. Naive-Bayes. This is due to their basic assumption of predictor distribution independence [56], which cannot be respected in the studied problems. In fact, the predictors are records of the damaged ship floating position taken at subsequent time instants that are correlated by definition. A preliminary attempt to adopt Naive-Bayes algorithms performed very badly compared to the other tested algorithms, confirming the rightfulness of their exclusion.

5.3.1 Decision Trees

DT are a very popular nonparametric supervised algorithms, based on recursive binary decisions that drive to define the local region of the input space. Their name descends to the shape of the decision process, which can be represented as a tree. The process starts from a root and then moves node by node according to predictors values up to the leaves, i.e. the predicted responses. The typical structure of a decision tree is given in Figure 5.2. Starting from the route, the decisions are taken at each node among two possible directions according to the outcome of a test function $f(\mathbf{x})$. The most common decision function is binary. It means that it is based on the value of a single predictor x_j, which divides the input region into two sub-regions at each node. The predictor and its threshold value are chosen in order to gain the best fit. A similar process is carried out at each passed node, until, decision by decision, a leaf is reached, corresponding to the response [41].

DT can be adopted in both classification and regression problems. Considering the latter task, they provide a so-called piecewise approximation of the response function, i.e. a constant response r_k provided at each k-th leaf of the tree [57].

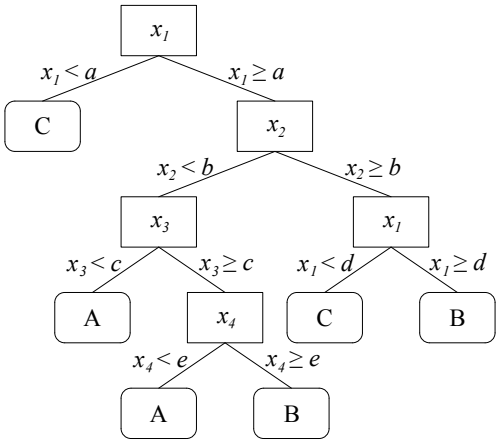

Figure 5.2 Sketch of a simple decision tree employed in a three-class classification problem in a four-dimensional input space [13]

Trained DT can be easily interpreted, since the mimic quite well a human-like behaviour in approaching a machine learning problem. Another advantage is that DT can be directly applied to non-numeral data, e.g. categorical, and can be easily applied to linear as well as non-linear problems without changing their core logic.

In the present work the different DT learners have been tested for the three studied problems:

- Decision Trees (DT): classification/regression is performed by a single decision tree. For the classification tasks, Gini's diversity index has been applied as an impurity measure in the splitting criterion and testing all the predictors at each node to select the one that maximises the split-criterion gain. In the regression task the maximisation of the *MSE* reduction evaluated on the node's items has been employed [15];

- Boosted Decision Trees (DTB): the method is an ensemble algorithm, i.e. an algorithm employing multiple decision trees. In general, boosting paradigm combines the outcomes of multiple weak learners to define an overall robust learner. The single learners are trained sequentially using at each iteration a new random sample data set to focus on the specific items where previous learners did not perform well. Since each tree is dependent on the output of the previous, boosted algorithms are usually slower to train compared to other ensemble methods where learners can be trained in parallel. In the present work, different boosting algorithms have been adopted for the classification and regression tasks. Considering the studied classification problems, the final fate can easily result in a large majority of survival or non-survival scenarios, leading to imbalanced training data. Even more skewed training-data are expected for the flooded compartments. This is why, for classification tasks has been selected a random undersampling boosted decision trees, based on a hybrid sampling/boosting algorithm [98], that proved to better deal with skewed training-data while assuring a reasonable computational effort. On the other hand, for the regression task, boosted decision trees have been employed based on a Least Squares (LS) boosting algorithm [41];

- Random Forests (RF): The method is an ensemble method employing bagging paradigm, i.e. the application of multiple weak learners trained in parallel [14]. Like bootstrap aggregation (the most common bagging algorithm), instead of a single tree trained with the complete database, the problem is decomposed in a set of trees trained with a random partition of the database. The final response of the overall model is selected according to the vote of the multiples trees for classification problems and as the average of responses for regression ones. Besides, to decorrelate the trees in the ensemble, random forests employs also a random selection of a predictors' subset each time a split in a tree is considered [54]. The random forests have proved to be more resilient to noise/missing data and more capable to deal with higher dimensionality data. This is why it has been selected for testing on the studied problems, which involve a large number of predictors for higher t^* while might be affected by the progressive flooding uncertainties [77, 7].

In the present work, the DT model is grown up to 200 splits. Moreover, 30 decision trees have been employed to grow both the RF and the DTB models.

5.3.2 K-Nearest Neighbour

KNN is another popular family of instance-based learners where the response associated with a new item is assigned according to votes given by its nearest neighbours in the input space or the mean value of their response for classification and regression problems, respectively. Hence, they are based on the implicit assumption that similar items are closer together [75]. The process is schematically shown in Figure 5.3 for a three-class classification problem in two-dimensional input space. The k parameter stands for the number of neighbours involved in the evaluation process. For instance, if $k = 1$, the response related to a given set of predictors is equal to the one of its nearest neighbour, i.e., the training data point having the lower distance in the predictors' space. Therefore the metric adopted for distance measurement is the key component of a KNN algorithm.

Due to the above described behaviour of a KNN algorithm, the training phase is limited to storing and, possibly, standardise the training data, whereas, during prediction, the distance of a new item shall be evaluated from each item in the training dataset. Hence, despite the algorithm is very simple and effective it can present a high memory and computational cost [57]. For this reason, is often called "lazy algorithm".

Given two data points identified by $n \times 1$ predictors' vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ respectively, the following different algorithms employing different metrics have been here tested:

- Quadratic K-Nearest Neighbour (KNN2): employing the Euclidean distance defined as:

$$d(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{\sum_{j=1}^{n} \left(x_{1_j} - x_{2_j}\right)^2} \tag{5.4}$$

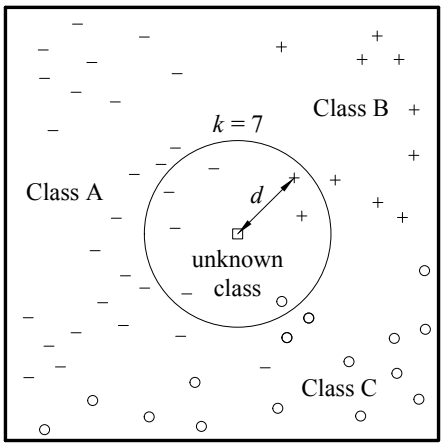

Figure 5.3 Sketch of the behaviour of a K-Nearest Neighbour algorithm for a three-class classification problem in a two-dimensional input space [13]

- Cubic K-Nearest Neighbour (KNN3): employing the third-degree Minkoswki distance defined as:

$$d(\mathbf{x}_1, \mathbf{x}_2) = \sqrt[3]{\sum_{j=1}^{n} \left| x_{1_j} - x_{2_j} \right|^3} \tag{5.5}$$

- Cosine K-Nearest Neighbour (KNNC): employing the cosine distance defined as:

$$d(\mathbf{x}_1, \mathbf{x}_2) = 1 - \frac{{\mathbf{x}_1}^T \mathbf{x}_2}{\sqrt{({\mathbf{x}_1}^T \mathbf{x}_1)({\mathbf{x}_2}^T \mathbf{x}_2)}} \tag{5.6}$$

- Weighted K-Nearest Neighbour (KNNW): in this method, the Euclidean distance is still employed but a different weight is applied to each neighbour response [33]. The closer is the neighbour is to the new item, the higher its response weight will be. Here, the squared inverse of the Euclidean distance is used as weight. Thus, in classification tasks, votes are weighted accordingly, whereas, for the regression task, the response is evaluated as:

$$y(\mathbf{x}) = \frac{\sum_{j=1}^{k} y_j / \left(d(\mathbf{x}, \mathbf{x}_j) \right)^2}{\sum_{j=1}^{k} 1 / \left(d(\mathbf{x}, \mathbf{x}_j) \right)^2} \tag{5.7}$$

In the present work for each of the tested methods, the k parameter is assumed to equal 10. Moreover, the training dataset is standardised: for each predictor defining the input space, the mean value and standard deviation are computed for centring and scaling the training data, respectively. Mean values and standard deviations are, then, stored and employed to preliminary standardise predictors of each new item submitted to the learner for response evaluation.

5.3.3 Support Vector Machine

SVMs are supervised learners algorithms based on statistical learning frameworks that can be employed in both classification and regression problems. They were first introduced to face binary classification [108] by defining the best hyperplane separating the elements belonging to the two different classes. Separation is performed employing a minimum number of items from the training dataset, colled support vector. For instance, in Figure 5.4a the support vector is represented by the encircled items. The support vector is composed of the closest data points to the hyperplane which is selected to maximise the margin between the two classes, i.e. the distance between two parallel hyperplanes defining a region not containing any data point [27].

Although basic SVMs are binary, they have been extended also to multi-class problems employing Error-Correcting Outputs Codes (ECOC). Since both the classification tasks studied in this work are multi-class problems, ECOC shall be employed to reduce a multiclass problem to a set of binary classification problems that can be addressed by conventional SVMs. To do so, ECOC model comprises two main components:

- *Coding design*: which is a matrix defining which classes are trained by a specific binary learner;
- *Decoding scheme*: which aggregates the results of the single binary classifiers determining the prediction of the multi-class problem.

In the present work, a one-versus-one coding design has been employed [35].

SVMs have been also generalized for regression, defining the hyperplane that best fits the training data [107]. Its equation $f(\mathbf{x}) = \mathbf{x}^T\mathbf{w} + \mathbf{b}$ is obtained minimising the norm value $\mathbf{w}^T\mathbf{w}$ provided that, for each item, all the prediction errors are inside an accepted error ε. In such a case the problem is called feasible. Otherwise, for infeasible problems, slack variables can be added to deal with data points having error exceeding ε [3].

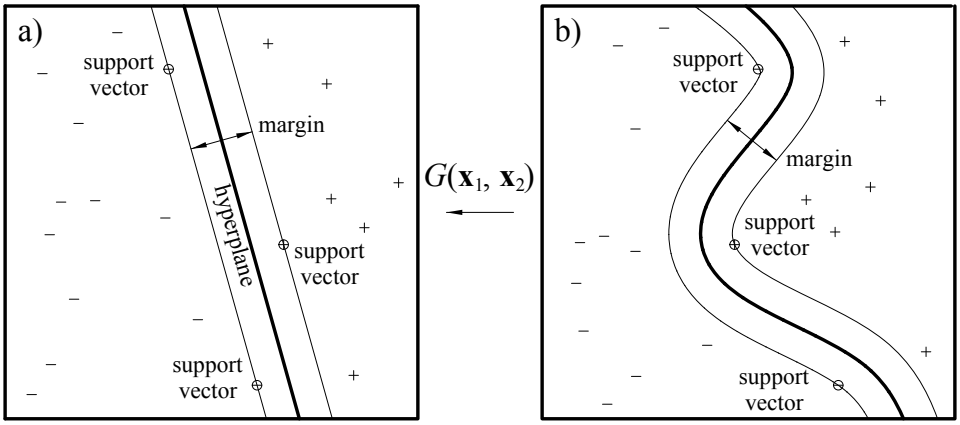

Figure 5.4 Sketch of a binary Support Vector Machine, highlighting the behaviour of the kernel function G mapping the space b) to the a) one [13]

SVMs are by definition linear classifiers, however, they can be extended to face non-linear problems too, by applying the so-called kernel trick. When the separation among classes or the regression function cannot be defined by a simple hyperplane the separating criterion can be reconducted to a hyperplane by replacing the dot product with a proper kernel function $G(\mathbf{x}_1, \mathbf{x}_2)$ [31]. In other words, mapping the original space into a higher-dimensional space where the linear solution of the problem is still possible [97]. Figure 5.4b provides a qualitative example for a binary non-linear classification problem.

In the literature, a large number of single or multiple kernels can be found, opening the problem of the selection of the proper one. The selection is not an easy task [75]. To address this issue for the three studied problems, the multiple kernels have been tested in the present work:

- Linear Support Vector Machine (SVM1): employing a linear kernel function, i.e. the dot product:
$$G(\mathbf{x}_1, \mathbf{x}_2) = {\mathbf{x}_1}^T \mathbf{x}_2 \tag{5.8}$$

- Quadratic Support Vector Machine (SVM2) employing a second-order polynomial kernel function defined as:
$$G(\mathbf{x}_1, \mathbf{x}_2) = \left(1 + {\mathbf{x}_1}^T \mathbf{x}_2\right)^2 \tag{5.9}$$

- Cubic Support Vector Machine (SVM3): employing a third-order polynomial kernel function defined as:
$$G(\mathbf{x}_1, \mathbf{x}_2) = \left(1 + {\mathbf{x}_1}^T \mathbf{x}_2\right)^3 \tag{5.10}$$

- Gaussian Support Vector Machine (SVMG): employing a radial basis function kernel defined as:
$$G(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\left\|\mathbf{x}_1 - \mathbf{x}_2\right\|^2\right) \tag{5.11}$$

Same as KNN, all the predictors defining the input space have been, here, standardised to define the training dataset.

5.4 Database generation

As mentioned, to assess the damage consequences by applying machine learning, a training dataset is needed, composed of progressive flooding simulations. The simulations are driven by a damage case generation algorithm. In the present section, several options are proposed to generate the training database according to a different characterisation of the damages.

In the field of damage stability and flooding simulation, it is a common practice to assume the damage as a parallelepiped box intersecting the hull [38, 17, 86]. The surface of the hull shell enclosed in the damage box is considered open to the sea, defining the damage. Considering

a box-shaped damage due to side collision (always crossing the waterline), it is completely defined by five parameters (Figure 5.5):

- damage length L_d;
- the longitudinal position of the damage centre X_d;
- transversal damage penetration B_d measured from shell side ($B/2$);
- vertical height Z_{max} of the highest tip of the damage above Base Line (BL);
- vertical height Z_{min} of the lowest tip of the damage above BL;

In the present work focusing on flooding simulation, all the internal structures have been considered intact, hence the damage penetration has been neglected.

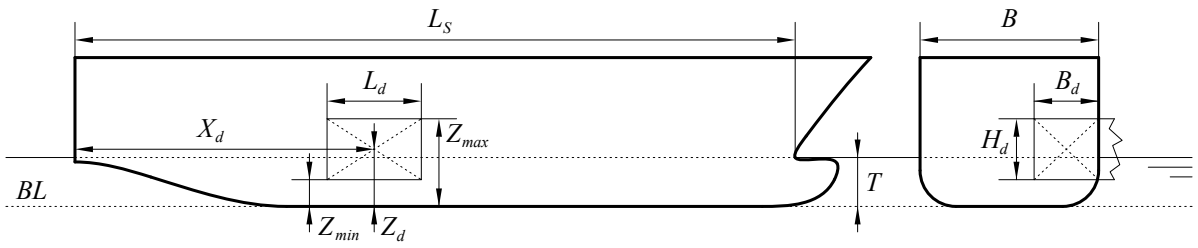

Figure 5.5 Bow-shaped damage parameters

Two main methods for database generation have been tested, one based on MC generation of side damage cases, the second based on a parametric generation aimed to cover all the possible damage scenarios involving multiple rooms laying on the hull shell [10]. Applying MC sampling, the damage cases can be generated following the probability distribution of their parameters [61]. Here three different MC options are explored, one based on SOLAS probability distributions and the other two based on uniform distributions of different damage parameters.

5.4.1 Monte Carlo with SOLAS probability distributions

Current SOLAS probabilistic rules for ship damage stability are based on the statistical analysis of a database of side collision accidents [49]. SOLAS rules employ a so-called zonal approach. However, the probability distributions adopted in the rules definition can be directly applied in the so-called non-zonal approach [18]. SOLAS embeds the probability distribution of most of the damage parameters: i.e. L_d, X_d, B_d (leading to the definition of the p-factor) and Z_{max} (considered in the v-factor). The Z_{min} is not defined, since SOLAS adopts a worse case approach in s-factor determination to consider horizontal subdivision below the waterline. Here, Z_{min} probability distribution is defined according to the statistical analysis of collision damage data [17]. In the following, adopted probability distributions are given:

- damage length is modelled with a bilinear probability density function, leading to the following cumulative distribution:

$$cdf(L_d) = \begin{cases} 0 & \text{if} \quad J \leq 0 \\ \dfrac{b_{11}}{2}J^2 + b_{12}J & \text{if} \quad 0 \leq J \leq J_k \\ \dfrac{b_{11}-b_{21}}{2}J_k^2 + (b_{12}-b_{22})J_k + \dfrac{b_{21}}{2}J^2 + b_{22}J & \text{if} \quad J_k < J \leq J_m \\ 1 & \text{if} \quad J > J_m \end{cases} \tag{5.12}$$

  where $J = L_d/L_S$ and all the other parameters are defined as for SOLAS Ch.II-1 Part B-1 Regulation 7-1 [51].

- the longitudinal position of the damage centre is uniformly distributied along the ship subdivision length L_S length:

$$cdf(X_d) = \begin{cases} 0 & \text{if} \quad X_d \leq 0 \\ \dfrac{X_d}{L_S} & \text{if} \quad 0 < X_d < L_S \\ 1 & \text{if} \quad X_d \geq L_S \end{cases} \tag{5.13}$$

- vertical height Z_{max} is modelled with a bilinear cumulative density function:

$$cdf(Z_{max}) = \begin{cases} 0 & \text{if} \quad Z_{max} - T \leq 0 \text{ m} \\ 0.8\dfrac{Z_{max}-T}{7.8} & \text{if} \quad 0 \text{ m} \leq Z_{max} - T \leq 7.8 \text{ m} \\ 0.8 + 0.2\dfrac{Z_{max}-T-7.8}{4.7} & \text{if} \quad 7.8 \text{ m} < Z_{max} - T \leq 12.5 \text{ m} \\ 1 & \text{if} \quad Z_{max} - T > 12.5 \text{ m} \end{cases} \tag{5.14}$$

- vertical height Z_{min} is modelled with a linear probability density function leading to the following cumulative distribution:

$$cdf(Z_{min}) = \begin{cases} 0 & \text{if} \quad Z_{min} \leq 0 \\ 1.4\dfrac{Z_{min}}{T} - 0.4\left(\dfrac{Z_{min}}{T}\right)^2 & \text{if} \quad 0 < Z_{min} < T \\ 1 & \text{if} \quad Z_{min} \geq T \end{cases} \tag{5.15}$$

5.4.2 Monte Carlo with uniform distribution of damage dimensions

In this second damage cases generation algorithm, the SOLAS values regarding the maximum damage dimensions are still applied. However, the damage length and height are assumed to

follow a uniform distribution. The applied cumulative density functions are, then, defined as:

- damage length is assumed as uniformly distributed between 0 and the maximum admissible non-dimensional length according to SOLAS:

$$cdf(L_d) = \begin{cases} 0 & \text{if} \quad X_d \leq 0 \\ \dfrac{L_d}{J_m L_S} & \text{if} \quad 0 < X_d < J_m L_S \\ 1 & \text{if} \quad X_d \geq J_m L_S \end{cases} \tag{5.16}$$

- the longitudinal position of the damage centre is already uniformly distributed in SOLAS. Hence, Equation (5.13) can be still applied.

- the damage height $H_d = Z_{max} - Z_{min}$ is uniformly distributed between 0 and $T + 12.5$ m, i.e. the maximum value according to SOLAS:

$$cdf(Z_{max}) = \begin{cases} 0 & \text{if} \quad H_d \leq 0 \\ \dfrac{H_d}{T + 12.5 \text{ m}} & \text{if} \quad 0 < H_d < T + 12.5 \text{ m} \\ 1 & \text{if} \quad H_d \geq T + 12.5 \text{ m} \end{cases} \tag{5.17}$$

- As the damage height is defined, the vertical position of the damage centre Z_d is defined assuring that the damage is crossing the waterline in compliance with SOLAS. Hence, the Z_d is assumed as uniformly distributed in the interval:

$$\left[\max\left(\frac{H_d}{2},\, T - \frac{H_d}{2} \right),\, T + \min\left(12.5\ (m) - \frac{H_d}{2},\, T + \frac{H_d}{2} \right) \right] \tag{5.18}$$

5.4.3 Monte Carlo with uniform distribution of damage area inverse

This last damage case generation algorithm based on MC sampling is quite different from the previous ones. It considers a uniform distribution of the inverse of the damage area $A_d = L_d(Z_{max} - Z_{min})$. The main objective of such a method is to reach a more uniform distribution of the time-to-flood, compared to the SOLAS one. In fact, the application of SOLAS probability distributions likely leads to a great number of large-area breaches having, consequently, short time-to-flood. Such damages are not very interesting for decision support purposes. Furthermore, lead to a small number of long damage scenarios, which might affect the forecast accuracy of the learners due to lack of data. This is why the present generation method has been designed.

In a preliminary study reported in Appendix B, it was observed that for very simple geometries the time-to-flood is proportional to the inverse of damage area. Hence, the following cumulative probability function has been applied to draw damage areas:

$$cdf\left(\frac{1}{A_d}\right) = \begin{cases} 0 & \text{if } \frac{1}{A} \leq \frac{1}{A_{max}} \\ \dfrac{\frac{1}{A} - \frac{1}{A_{max}}}{\frac{1}{A_{min}} - \frac{1}{A_{max}}} & \text{if } \frac{1}{A_{max}} < \frac{1}{A} < \frac{1}{A_{min}} \\ 1 & \text{if } \frac{1}{A} \geq \frac{1}{A_{min}} \end{cases} \tag{5.19}$$

where A_{min} and A_{max} are the minimum and maximum damage area that can be defined for each different ship. In a real application, floodwater inflow due to very small damages can be taken under control by the bilge system. Hence, it can be assessed considering the bilge pumps capacity. The maximum area can be defined as the maximum damage area according to SOLAS: $A_{max} = (T + 12.5)\, J_m\, L_S$.

Once the area is defined the other damage parameters shall be assessed. The longitudinal position of the damage centre is assessed as for SOLAS according to Equation (5.13). Then, two alternative procedures are applied to the 50% of generated damage cases, respectively:

- the heights Z_{min} and Z_{max} are drawn following the SOLAS cdf provided in Equations (5.15) and (5.14), respectively. Then, defined the damage height $H_d = Z_{max} - Z_{min}$, the damage length is determined as $L_d = A_d / H_d$;
- the damage length is drawn following the SOLAS cdf provided in Equation (5.12) and damage height H_d is determined as $H_d = A_d / L_d$. Then, the vertical position of damage centre Z_d is randomly drawn within the range defined in Equation 5.18 to have the damage crossing the waterline.

Damages resulting not feasible (i.e. the ones having height or length outside the ship boundaries) are discarded and generated randomly again.

5.4.4 Parametric method

The parametric generation of damage cases is not based on MC sampling and is divided into two phases. In the first, the box-shaped damages are generated considering every single room on the hull shell. In the latter, these damage cases are parametrically combined to define damage cases involving neighbouring rooms.

Considering a single room, it was observed that the longitudinal position of the damage centre has only a limited impact on progressive flooding (See Appendix A). Hence, all the room damages have been applied at half of the room longitudinal extension assuming an L_d equal to the room longitudinal extension. On the contrary, for each room, at least three different vertical positions of damage centre L_d have been considered: at the room's bottom, half-height and top. For rooms extending over multiples decks more intermediate positions have been considered corresponding to the main decks' hights from baseline. In order to define damages within

room boundaries, the damage centre height shall be corrected for top and bottom damages, considering the applied damage area as shown in Figure 5.6. For each possible damage location (X_d, Z_d), multiple damage sizes shall be applied. The area of the i-th damage in j-th room, considering the k-th centre is evaluated as:

$$\frac{1}{A_{ijk}} = \frac{1}{A_{max_{jk}}} + \frac{k}{n_d}\left(\frac{1}{A_{max_{jk}}} - \frac{1}{A_{min_{jk}}}\right) \tag{5.20}$$

with $k = [1, 2, ..., n_d]$, where n_d is the so-called number of divisions, which is the main parameter governing the database size. Besides, A_{min} and A_{max} are defined with a different procedure compared to the previously described. Regarding the minimum area, for each position, it can be defined as the one corresponding to an initial inflow equal to bilge pumps capacity. On the other hand, very large damages drive to the almost instantaneous filling of the damaged room. In the present study, the maximum damage area for each room has been estimated for each location as the area that causes the room-filling in 15 s. The maximum area has been defined solving recursively the equations provided in Appendix B valid for simple geometries. Furthermore, every single room is assumed as lost at the beginning of progressive flooding (instantaneous flooding) defining an additional single-room damage case. As the single room damage cases have been defined, they are combined with the ones related to the neighbouring rooms. In detail, combined damage cases are elaborated considering all the possible combinations of damage areas of damages having the same centre height (bottom, half-height, top) and sharing a boundary (watertight bulkhead, deck). For instance, at the intersection of a deck and a transverse bulkhead, the combinations are defined considering up to four rooms. In the present work, only one or two compartments damages have been considered. However, the proposed generation technique can be easily extended to a higher number of contiguous compartments.

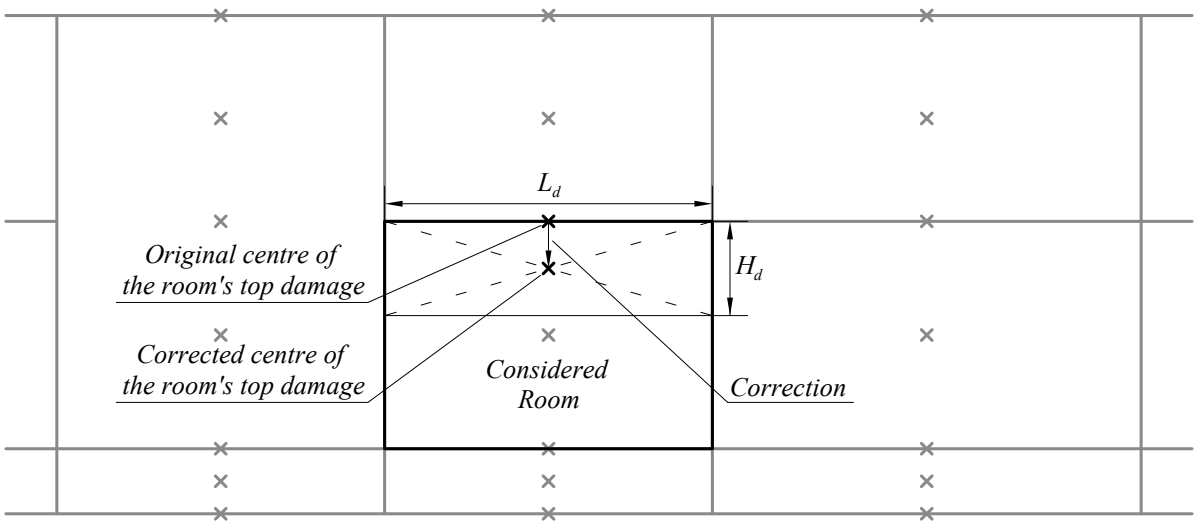

Figure 5.6 Correction of the vertical position of the damage centre according to damage area for a top damage

Chapter 6

Machine Learning test case

In the present chapter, the methodology presented in Chapter 5 is applied to a quite simple test geometry. The main objective is to assess the feasibility of the damage consequences recognition based on the time evolution of the floating position of the damaged ship. After a description of the test arrangement, multiple learners have been tested and compared. Finally, the effect of different algorithms for training database generation is studied.

6.1 Test arrangement

In the present section, the chosen test arrangement is introduced and described. First, the test geometry is described in detail, then some considerations are made to explain its choice.

6.1.1 Geometry

The test geometry is a box-shaped barge having the main particulars provided in Table 6.1 and general arrangement depicted in Figure 6.1. The barge has three decks: DK1 located nearby the waterplane at 5 m above the BL, the DK2 located at 10 m above the BL and the bulkhead deck (DK3) located 15 m above the baseline. Beneath the bulkheads deck, the barge is divided into five watertight compartments. The lowest rooms in first and third compartments extend from the barge bottom to the DK2. Longitudinal bulkheads are fitted on the centreline in the lowest room within compartments 1 and 3. Moreover, another longitudinal bulkhead is fitted in compartment 5 between DK2 and DK3 to allow the generation of a long flooding chain.

Table 6.1 Main particulars of the test geometry

Description	**Symb.**	**Value**		**Description**	**Symb.**	**Value**	
Length overall	L_{OA}	75	m	Breadth	B	20	m
Draught	T	6	m	Depth	D	17.5	m
Displacing Volume	∇	7500	m^3	Metacentric height	GM	2.685	m

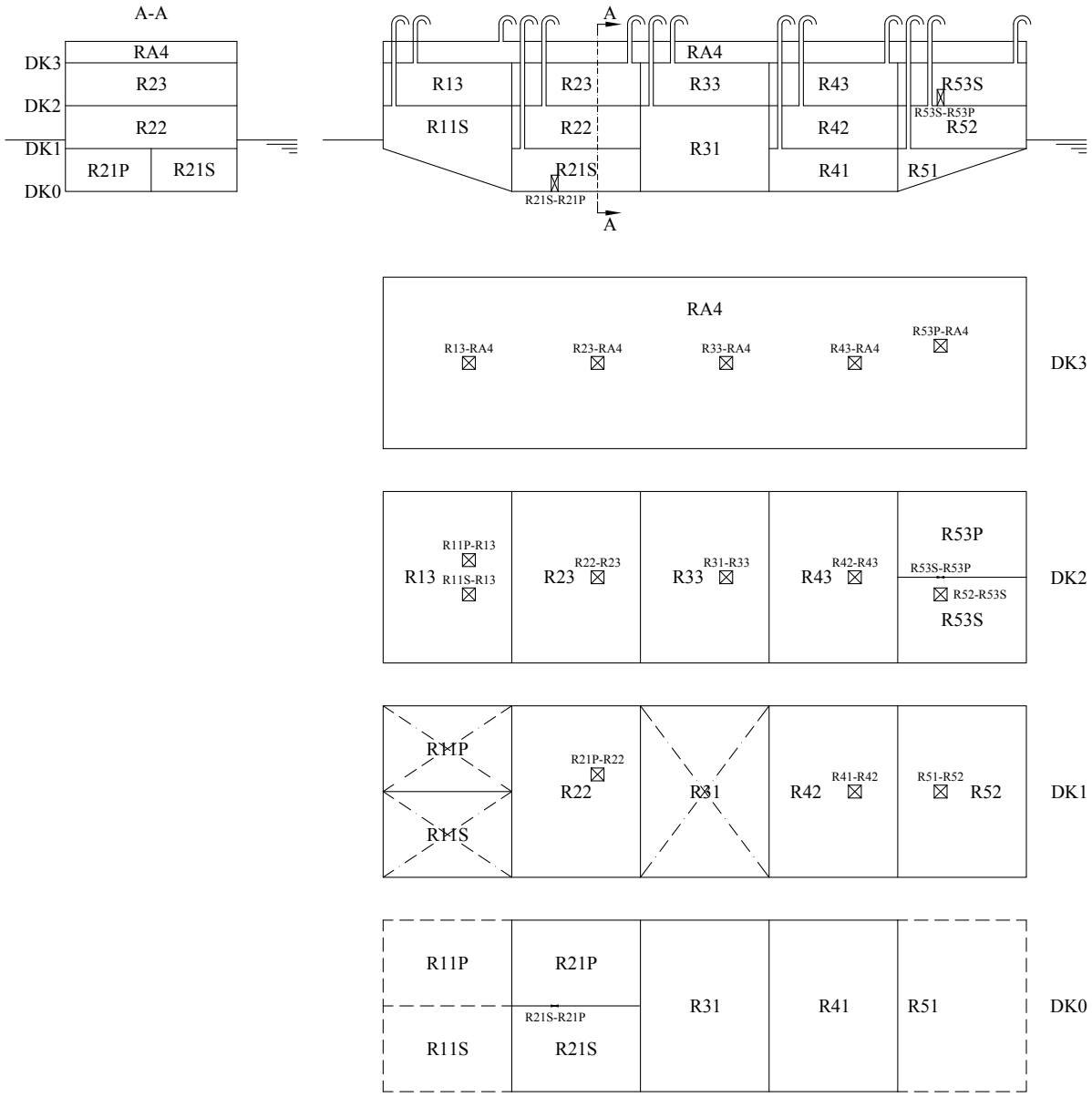

Figure 6.1 General arrangement of the test geometry

All the rooms have unitary permeability and are considered fully vented. The rooms are interconnected by a set of openings as shown in Figure 6.1. Details about the openings size and position are provided in Table 6.2. Both the rooms and the openings have been modelled with non-structured triangular meshes having maximum panel are of 3.125 m^2 and 0.25 m^2, respectively. A sketch of the 3D model of the test geometry is shown in Figure 6.2.

6.1.2 Motivation

The test case has the main purpose to test the proposed methodology based on machine learning. It shall be considered the first attempt in such a direction devoted to proving its feasibility for application in a flooding-sensor-agnostic DSS.

Table 6.2 Main characteristics of the test geometry openings. $C = (X_C, Y_C, Z_C)$ is the centre of the opening in ship-fixed reference system

id	**type**	h **(m)**	w **(m)**	X_C **(m)**	Y_C **(m)**	Z_C **(m)**
R11S-R13	Horizontal	1.5	1.5	10	-2	10
R11P-R13	Horizontal	1.5	1.5	10	2	10
R13-RA4	Horizontal	1.5	1.5	10	0	15
R21P-R22	Horizontal	1.5	1.5	25	2	5
R22-R23	Horizontal	1.5	1.5	25	0	10
R23-RA4	Horizontal	1.5	1.5	25	0	15
R31-R33	Horizontal	1.5	1.5	40	0	10
R33-RA4	Horizontal	1.5	1.5	40	0	15
R41-R42	Horizontal	1.5	1.5	55	0	5
R42-R43	Horizontal	1.5	1.5	55	0	10
R43-RA4	Horizontal	1.5	1.5	55	0	15
R51-R52	Horizontal	1.5	1.5	65	0	5
R52-R53S	Horizontal	1.5	1.5	65	-2	10
R53P-RA4	Horizontal	1.5	1.5	65	2	15
R21S-R21P	Longitudinal	1.9	0.8	20	0	0.95
R53S-R53P	Longitudinal	1.9	0.8	65	0	10.95

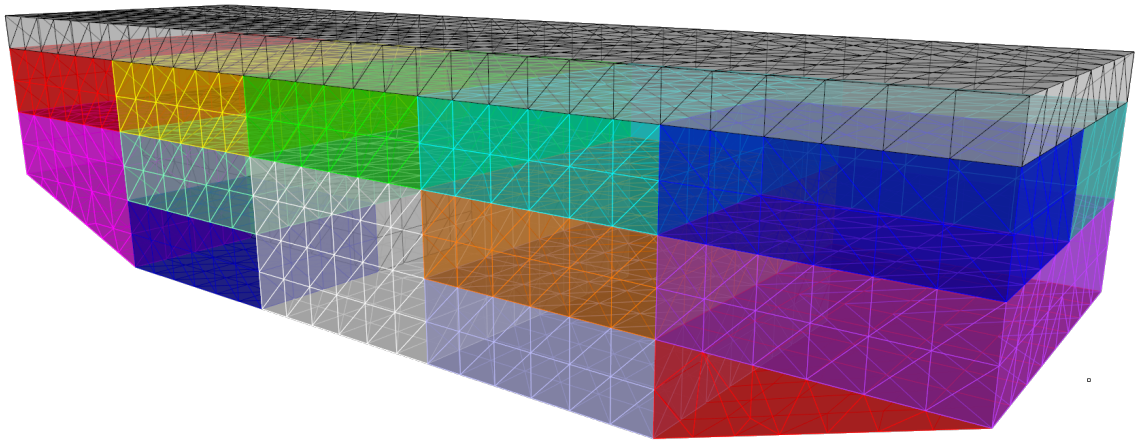

Figure 6.2 Mesh of the internal rooms of the test geometry

Following the previous considerations, it has been chosen to test the new methodology on quite simple geometry. Hence, the box-shaped barge and its internal subdivision have been designed. The barge geometry can be easily reproduced in other studies, providing a first benchmark test case to compare the outcomes of learners/techniques not addressed in the present work. This is considered a best practice when a new approach to a damage stability problem is proposed [19]. Furthermore, the application on a full-scale passenger ship has been deemed too complex at this stage, where neither the best machine learning technique nor the best database generation strategy is known. The application on a full-scale environment, which is anyway essential before onboard application of the proposed technique, will be addressed in future works,

following the proof-of-the-concept provided in the present study.

Nevertheless, although the test geometry is simpler than a large passenger ship one, the box-shaped barge has been designed keeping in mind the typical general arrangement of a large cruise vessel, characterised by multiple decks below the bulkhead deck and having some spaces divided by longitudinal bulkheads, e.g. the Primary Electric Motor (PEM) rooms or the steering gear rooms. Moreover, it has also considered that many spaces below the bulkhead deck pass through multiple decks to accommodate the main machinery. All these aspects can be found in the selected test geometry.

6.2 Tested learners

In the present section, all the learners described in Chapter 5 are tested on the box-shaped barge. First, the density of predictors and the number of damage cases in the training database are established. Then, a fair comparison of the performances associated with each tested learner is provided and discussed.

6.2.1 Sensitivity on number of predictors and damage cases

In this preliminary phase, both the training and validation databases are generated according to SOLAS probability distributions for side collision damages. Before testing all the learners, a suitable value of dt spearing two subsequent records of the floating position shall be defined as well as a proper dimension of the training database.

Given a large validation database, it is expected that as the number of damage cases included in the training database increases, also the performances of classification/regression will improve. To assess the minimum number of damage cases comprised in the training database, the following databases have been independently grown-up:

- MC01: composed of 1000 damage cases (173 non-survival cases);
- MC02: composed of 2500 damage cases (398 non-survival cases);
- MC05: composed of 5000 damage cases (768 non-survival cases);
- MC10: composed of 10000 damage cases (1544 non-survival cases);
- MC15: composed of 15000 damage cases (2286 non-survival cases);
- MC20: composed of 20000 damage cases (3082 non-survival cases);
- MC30: composed of 30000 damage cases (4686 non-survival cases);
- MC40: composed of 40000 damage cases (6263 non-survival cases);

- MC50a: composed of 50000 damage cases (7886 non-survival cases);
- MC50b: composed of 50000 damage cases (8059 non-survival cases);

The maximum simulated time has been set at 2250 s, since the large majority of the progressive flooding simulations have a shorter t_f. This threshold, for the test geometry, drives to about 0.6% of damage cases classified as TE.

Assuming a dt = 15 s and employing the RF learner, all the databases have been used as training database except for the MC50b, which was always employed as the validation one. Figure 6.3 provides the results of the analysis. Each curve refers to the learning performance evaluated at a different t^* after damage occurrence.

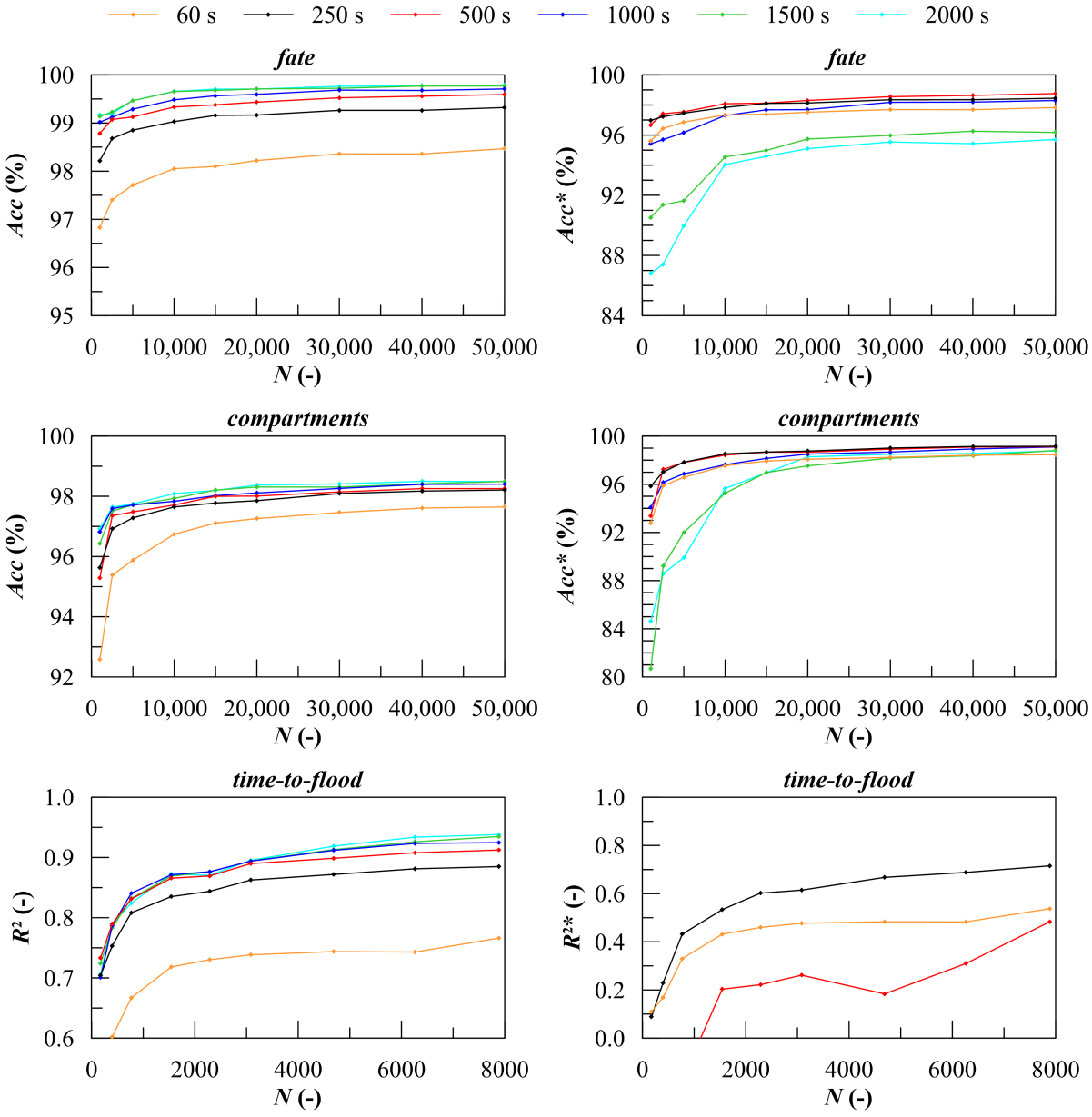

Figure 6.3 Performances evaluated at different time instants t^* as function of number of damage cases in the training database employing RF learners. Validation: MC50b

Almost all the curves converge at a maximum value of accuracy/coefficient of determination as N increases. The performances are already satisfactory for MC20 training database. Employing more than 20,000 damage cases marginally improves the accuracy while such gains are achieved at a high computational cost. Employing an Intel® Xeon® CPU E5-2630 v4 (2.20 GHz) workstation with 18 threads running in parallel to growing up the databases, about 1 hour is required to simulate 1000 damage cases. Therefore, 20,000 damage cases are considered a reasonable size for a training dataset based on MC sampling for the present test geometry.

Figure 6.4 shows the effect of different constant sampling time steps dt. Different behaviour is observed for classification and regression problems. Both FF and FC classification problems show a strong degrading of performances for very short dt.

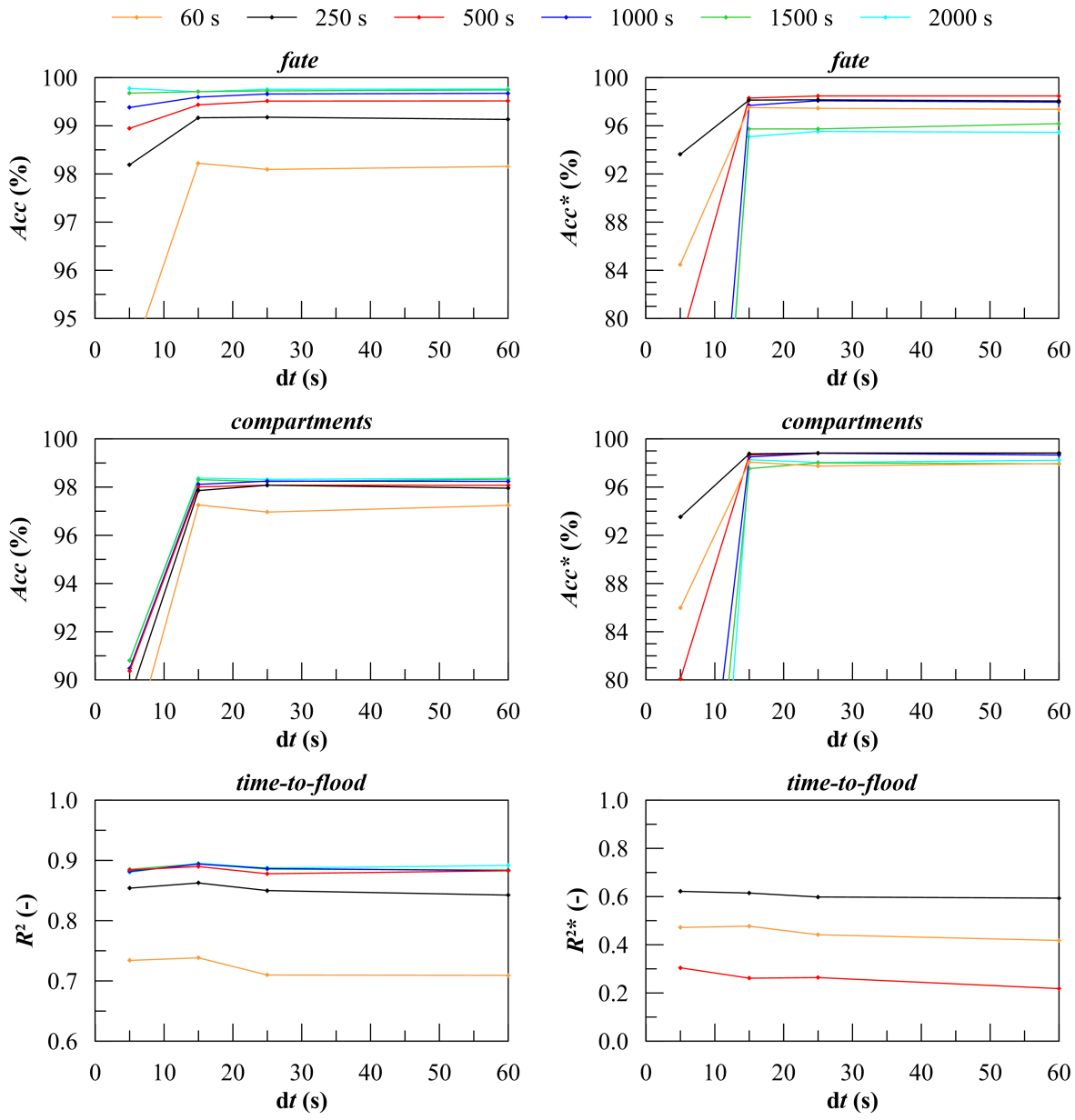

Figure 6.4 Performances evaluated at different time instants t^* as function of the sampling time step. Training: MC20; validation: MC50b

Then, for the test geometry, the accuracy is almost stable for 15 s $\leq$ dt $\leq$ 60 s. Considering regression on t_f, the coefficients of determination slightly grow as the sampling time step decreases in the studied range. Considering all the three problems, in this preliminary study a standard constant sampling time step has been chosen, equal to 15 s, since it assures good performances at the very beginning of the flooding process as shown by the line $t^* = 60$ s in Figure 6.4 and a negligible drawback for the classification problems in the next phases.

6.2.2 Comparison of learners performances

The learners presented in Chapter 5 have been trained using the MC20 database, using a sampling time step d$t = 15$ s for DT and KNN learners, whereas d$t = 60$ s has been used for SVM ones. In fact, SVMs does not usually work well with high-dimensional input space. Such a problem was observed also in the present work, driving to increase the sampling time step to reach a reasonable compromise between maximum accuracy and its fast convergence to the maximum value. Then, at each time instant, the learners have been tested through the validation database MC50b. The overall results of the analysis are provided in Figure 6.5, reporting the accuracy and the coefficients of determination for the classification and regression problems, respectively. Additional data can be found in Appendix C

Considering the overall performances, very good results have been obtained for all the three problems (FF, FC, t_f): at about $t^* = 500$ s the overall accuracy and determination coefficients converged for most of the tested learners. Considering the ongoing performances a quite different behaviour has been observed. In an initial phase both the Acc^* and R^{2*} increase up to a maximum value, which remains almost constant in a region defined according to the percentage N^* of ongoing damage scenarios included in the validation database. At a first approximation, for all three problems, this area ranges between 50% and 20%. When the $N^* < 20\%$, the ongoing performances become more unstable and decay below the maximum value. The decay and instability magnitude strongly depends on the studied problem and adopted learner.

At this stage, the best learners for each studied problem can be already identified. For both the classification problems (FF and FC) the RF definitely provide the best performance in terms of both overall and ongoing accuracy. On the other hand, considering the t_f regression, a clear preference cannot be inferred by the Figure 6.5. Again RF works well in the initial phases. Then, KNNW provides better results as $N^* < 10\%$. In the next section, these overall results are discussed in more detail, focusing on the best learners.

Finally, an analysis of the computational performances is provided in Tables 6.3-6.5. The computational time refers to a Intel® Core™ i7-4710 CPU (2.50 GHz) laptop. Considering the computational time required for the response prediction, all the tested learners can be considered suitable for the onboard application. In fact, the "laziest" KNN learners still provide an almost instantaneous response, which is several times lower than each DSS based on live simulation of the progressive flooding, which anyway requires a finite computation time.

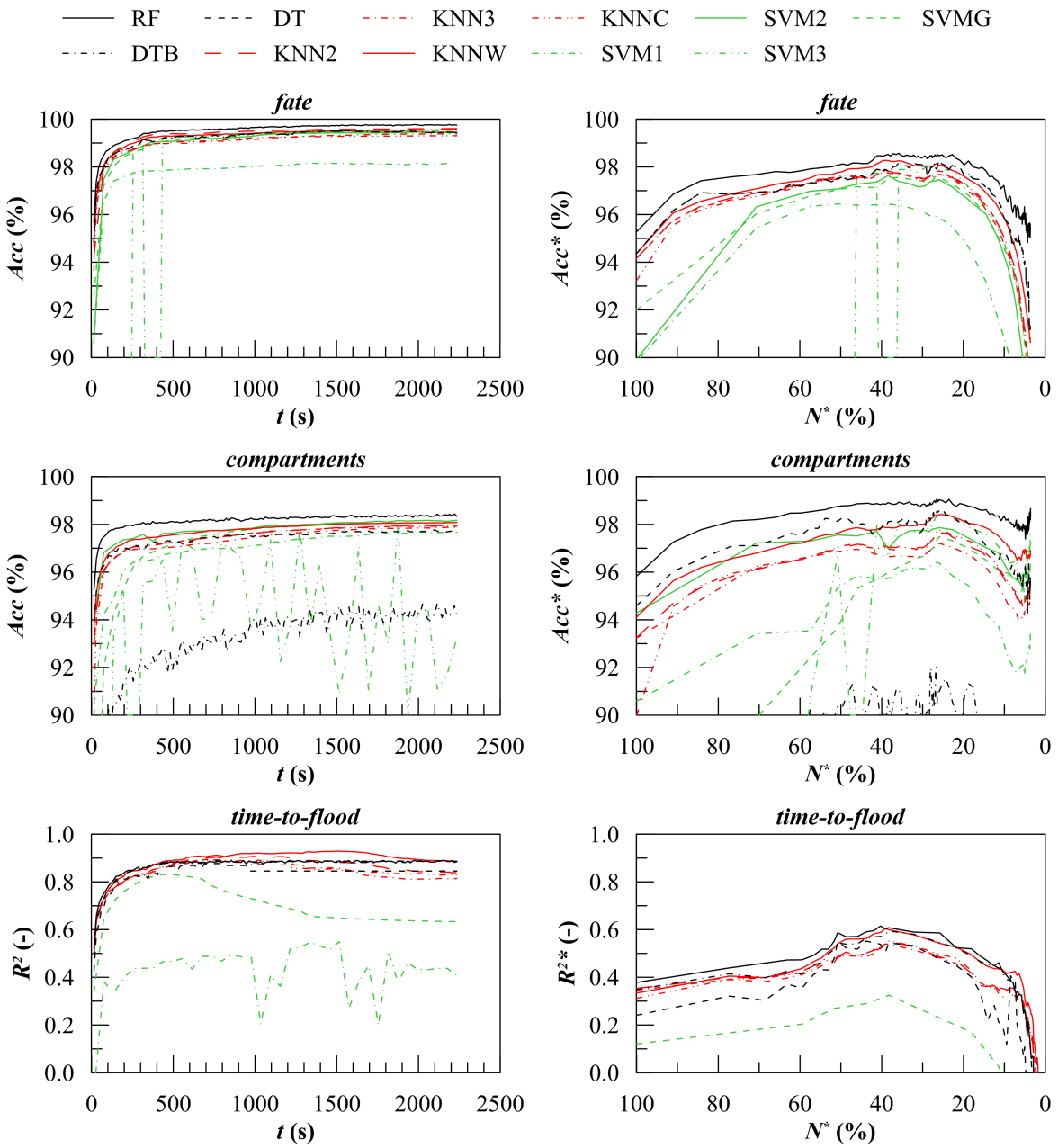

Figure 6.5 Comparison of the performances of tested learners. Training: MC20; validation: MC50b

Since the training is performed during the set-up of the system, the training time is less relevant. Nevertheless, reference values relating to the test geometry are given in tables for sake of completeness.

6.2.3 Discussion

It shall be noted that the maximum simulation time applied to the test geometry is quite limited. This is mainly due to the large volume of the compartments compared to the barge displacing volume combined with wide and free internal openings.

Table 6.3 Computational time required for training and prediction of a single damage case in ship final fate classification evaluated at different t^*

Methods	**training MC20 t_c (s)**			**prediction M50b t_c/N (s)**		
	$t^* = 250$ s	$t^* = 1000$ s	$t^* = 2000$ s	$t^* = 250$ s	$t^* = 1000$ s	$t^* = 2000$ s
DT	1.09	6.48	10.25	2.50E-06	2.50E-06	5.63E-06
DTB	1.42	2.64	3.61	1.56E-05	1.78E-05	2.13E-05
RF	8.00	12.38	12.59	2.53E-05	2.66E-05	2.47E-05
KNN2	0.11	0.47	0.92	1.18E-03	6.11E-03	1.20E-02
KNN3	0.33	0.66	1.06	4.27E-02	1.68E-01	3.35E-01
KNNC	0.17	0.42	0.84	1.20E-03	5.75E-03	1.16E-02
KNNW	0.20	0.42	0.84	1.21E-03	6.02E-03	1.20E-02
SVM1	9.91	24.63	12.50	3.75E-06	1.31E-05	2.31E-05
SVM2	43.34	11.94	10.09	1.19E-05	1.59E-05	1.02E-04
SVM3	762.56	17.42	6.61	9.38E-06	1.41E-05	5.91E-05
SVMG	4.27	5.52	8.13	1.88E-05	1.09E-04	2.69E-04

Table 6.4 Computational time required for training and prediction of a single damage case in flooded compartments classification evaluated at different t^*

Methods	**training MC20 t_c (s)**			**prediction M50b t_c/N (s)**		
	$t^* = 250$ s	$t^* = 1000$ s	$t^* = 2000$ s	$t^* = 250$ s	$t^* = 1000$ s	$t^* = 2000$ s
DT	1.25	4.63	8.70	4.69E-06	6.88E-06	7.50E-06
DTB	2.03	3.13	4.06	3.53E-05	4.03E-05	4.44E-05
RF	6.61	10.31	12.89	4.72E-05	4.53E-05	4.53E-05
KNN2	0.20	0.50	0.83	1.19E-03	5.97E-03	1.20E-02
KNN3	0.14	0.63	1.00	4.37E-02	1.72E-01	3.37E-01
KNNC	0.20	0.56	0.98	1.15E-03	5.77E-03	1.17E-02
KNNW	0.13	0.42	0.86	1.18E-03	5.80E-03	1.20E-02
SVM1	16.89	11.56	15.92	8.47E-05	1.93E-04	3.38E-04
SVM2	37.16	33.58	32.09	1.01E-04	1.87E-04	4.62E-04
SVM3	93.05	69.78	75.66	1.05E-04	1.78E-04	4.10E-04
SVMG	8.83	9.30	13.50	2.32E-04	4.37E-04	1.54E-03

Besides, the SOLAS probability distributions lead to a high probability of significantly large damages. Nevertheless, the maximum simulation time $t_{max} = 2250$ s has been deemed sufficient to suit the purpose of the present study. In the following, the studied problems are discussed in detail. Considering the classification of the ship's final fate, the best performances have been obtained applying RF. For the test geometry, the maximum overall and ongoing accuracy is 99.8% and 98.6%, respectively. Moreover, RF result also in the lowest performances decay as the N^* decreases: the ongoing accuracy is always above 95%. All the other tested learners proved slightly worse results for both accuracy and decay. The additional techniques based on decision trees (single DT and DTB) provide almost the same results and are about 1% less accurate than the RF in terms of ongoing accuracy. The decay is again limited: $Acc^* > 90\%$.

Comparable performances have been also achieved applying the KNNW learner, whereas the other tested KNN algorithms based on non-weighted metrics perform slightly worse.

Table 6.5 Computational time required for training and prediction of a single damage case in time-to-flood regression evaluated at different t^*

Methods	**training MC20 t_c (s)**			**prediction M50b t_c/N (s)**		
	$t^* = 250$ s	$t^* = 1000$ s	$t^* = 2000$ s	$t^* = 250$ s	$t^* = 1000$ s	$t^* = 2000$ s
DT	0.28	1.08	1.94E-06	7.76E-06	1.55E-05	
DTB	6.98	11.94	7.76E-06	2.33E-05	1.74E-05	
RF	5.06	9.47	2.33E-05	3.10E-05	3.30E-05	
KNN2	0.02	0.02	3.43E-04	1.03E-03	1.87E-03	
KNN3	0.06	0.02	8.32E-03	2.40E-02	4.66E-02	
KNNC	0.06	0.06	2.58E-04	3.37E-04	3.94E-04	
KNNW	0.06	0.06	3.24E-04	1.04E-03	1.87E-03	
SVM1	12.77	38.69	3.10E-05	4.27E-05	7.56E-05	
SVM2	39.39	39.53	7.76E-06	4.27E-05	6.98E-05	
SVM3	43.48	60.84	3.10E-05	4.46E-05	7.76E-06	
SVMG	0.56	0.84	7.76E-06	5.82E-05	9.89E-05	

A clear preference for one of the tested metrics does not emerge in the FF classification. Eventually, the application of SVMs enables to outline some interesting insights, although the best SVMs are slightly less accurate than KNN, especially in the initial phases, where the lower number of predictors slows down the convergence to maximum accuracy. The linear kernel (SVM1) is not suitable to deal with FF problem, whereas both the quadratic (SVM2) and Gaussian (SVMG) kernels work well. Finally, the cubic kernel (SVM3) shows an unstable behaviour: large oscillations of the accuracy have been observed in the initial phase characterised by a small number of predictors and a large number of sample data. In this initial phase very high training times have been experienced, confirming a difficult and ineffective training of the model based on cubic kernel. Finally, Acc^* decay related to all the SVM algorithms is stronger than the KNN's one, confirming that this class of algorithms is the less appropriate for the forecast of damaged ship final fate.

Besides accuracy, misclassification shall be analysed too. Considering FF classification problem, two error type can be defined:

- *Type I error*: when a non-survival scenario is classified as survival one by the learner;
- *Type II error*: when a survival scenario is classified as a non-survival one by the learner.

The type I error is extremely more dangerous in the framework of onboard decision support. In fact, in such a case the master might be pushed by the DSS not to abandon the ship when it will sink or capsize, with imaginable consequences on people safety. Tables 6.6 and 6.7 shows the confusion matrices evaluated for the test geometry at $t^* = 250$ s and $t^* = 500$ s, respectively. It can be noted that the type I error has a very low probability of occurrence, while the type II one is slightly more probable. This behaviour was observed during all the analysed time frame. Considering the classification of flooded compartments, the best performances have been again obtained applying RF.

Table 6.6 Confusion matrix related to ship final fate classification problem evaluated at $t^* =$ 250 s. Training: MC20; validation: MC50b; method: RF

True Class	**Predicted Class (%)**			
	Capsize	**Equilibrium**	**Ex. Heeling**	**Time Exceeded**
Capsize	**12.79**	0.17	0.01	0.02
Equilibrium	0.07	**82.97**	-	0.12
Ex. Heeling	0.04	0.03	**3.05**	-
Time Exceeded	0.02	0.32	-	**0.37**

Table 6.7 Confusion matrix related to ship final fate evaluated at 500 s. Training: MC20; validation: MC50b; method: RF

True Class	**Predicted Class (%)**			
	Capsize	**Equilibrium**	**Ex. Heeling**	**Time Exceeded**
Capsize	**12.89**	0.09	-	0.01
Equilibrium	0.05	**83.00**	-	0.12
Ex. Heeling	0.01	0.02	**3.05**	-
Time Exceeded	0.02	0.17	-	**0.52**

Moreover, most of the considerations done for FF classification still yield. Hence, focus will be made on the discrepancies only. The maximum value of overall and ongoing accuracy is 98.4% and 99.0%, respectively. It shall be noted the ongoing accuracy shows a smaller decay when RF are applied, being always greater than 98%. Moreover, the Acc^* gap between RF and the other learners is larger than 1%. Hence, RF proved to be an excellent solution for dealing with FC classification. Compared to FF classification, boosted trees (DTB) are very ineffective in the identification of flooded compartments. On the other hand, SVM2 shows good results, comparable with the DT and KNNW ones. In such a case SVMG is providing bad performances, especially in the initial phase after damage. This suggests that the FF classification cannot be described well by radial basis kernel, nor by a linear one. Again SVM3 shows a strong instability on the overall accuracy, even though it provides very good results for ongoing accuracy for $t^* > 500$ s.

The second problem is also somehow more resilient to misclassification, despite the overall accuracy appears lower than the FF one. Tables 6.8 and 6.9 shows the confusion matrices evaluated for the test geometry at $t^* = 250$ s and $t^* = 500$ s, respectively. It can be noted that most of the errors do not relate to a complete misclassification of the flooded compartments, but to neglecting one of the flooded compartments. This likely happened when in an initial flooding phase the breach area in the neglected compartment is substantially smaller than the other flooded compartments ones. Such an error tends to vanish as the floodwater amount in the neglected compartment sufficiently grow, having a relevant effect on the ship floating position. Finally, it shall be noted that the training dataset generated according to SOLAS probability distributions is very skewed, leading to a higher error rate for less probable damage scenarios involving three or five adjoining compartments.

Table 6.8 Confusion matrix related to flooded compartments classification problem evaluated at $t^* = 250$ s. Training: MC20; validation: MC50b; method: RF

True Class	Predicted Class (%)												
	1	**2**	**3**	**4**	**5**	**1,2**	**2,3**	**3,4**	**4,5**	**1,2,3**	**2,3,4**	**3,4,5**	**all**
1	**17.04**	-	-	-	-	-	-	-	-	-	-	-	-
2	-	**13.37**	-	-	-	-	0.01	-	-	-	-	-	0.03
3	-	-	**13.62**	-	-	-	-	-	-	-	-	-	-
4	-	-	-	**13.43**	-	-	-	0.01	-	-	-	-	-
5	-	-	-	-	**16.53**	-	-	-	-	-	-	-	-
1,2	0.01	-	-	-	-	**1.53**	-	-	-	-	-	-	0.26
2,3	-	0.09	-	-	-	-	**5.67**	-	-	-	-	-	0.01
3,4	-	0.01	-	0.06	0.02	-	-	**5.92**	0.04	-	-	0.01	0.05
4,5	-	-	-	-	0.06	-	-	0.01	**4.56**	-	-	0.01	0.21
1,2,3	-	-	-	-	-	-	-	-	-	**0.02**	-	-	0.02
2,3,4	-	-	-	-	-	-	0.01	0.02	-	-	**0.07**	-	-
3,4,5	-	-	-	-	-	-	-	0.03	0.03	-	-	**0.01**	-
all	0.02	0.04	-	0.01	0.04	0.28	0.02	0.20	0.36	0.01	0.01	0.02	**6.18**

Table 6.9 Confusion matrix related to flooded compartments classification problem evaluated at $t^* = 500$ s. Training: MC20; validation: MC50b; method: RF

True Class	Predicted Class (%)												
	1	**2**	**3**	**4**	**5**	**1,2**	**2,3**	**3,4**	**4,5**	**1,2,3**	**2,3,4**	**3,4,5**	**all**
1	**17.04**	-	-	-	-	-	-	-	-	-	-	-	-
2	0.01	**13.39**	-	-	-	-	-	-	-	-	-	-	0.01
3	-	-	**13.62**	-	-	-	-	-	-	-	-	-	-
4	-	-	-	**13.42**	-	-	-	0.01	-	-	-	-	-
5	-	-	-	0.01	**16.52**	-	-	-	-	-	-	-	-
1,2	-	-	-	-	-	**1.52**	-	-	-	-	-	-	0.28
2,3	-	0.06	-	-	-	-	**5.70**	-	-	-	-	-	0.02
3,4	-	-	-	0.04	0.01	-	-	**5.96**	0.03	-	-	0.01	0.06
4,5	-	0.01	-	-	0.03	-	-	0.01	**4.63**	-	-	0.01	0.16
1,2,3	-	-	-	-	-	-	-	-	-	**0.03**	-	-	0.02
2,3,4	-	-	-	-	-	-	0.01	0.02	-	-	**0.07**	-	0.01
3,4,5	-	-	-	-	-	-	-	0.03	0.03	-	-	**0.01**	-
all	0.01	0.04	-	0.01	0.03	0.26	0.01	0.19	0.37	0.01	0.01	0.02	**6.23**

Considering the regression on the t_f, the best learner cannot be immediately identified, as for the FF and FC classification problems. In an initial phase, again, the RF provides the best performances, then the KNNW shall be preferred for both overall and ongoing coefficients of determination. Furthermore, other relevant differences from the previous problems can be identified. First, the decay of R^{2*} is more pronounced compared to the decays observed on Acc^*. In particular, the ongoing determination coefficient reaches a null value when the number of ongoing damages is lower than the 2% (which occurs at about $t^* = 735$ s). The cause can be searched in the training dataset that here is limited to the non-survival damage cases only. Hence, besides a reduced size (3082 of the 20,000 damage scenarios for MC20 database), it has been observed that for the tested geometry most of the non-survival damage cases have very short time-to-floods. This lead to a very difficult assessment of long damage scenarios due to a lack of data for large t^* values.

Focusing on regression learners, both the RF and KNNW reaches a maximum of R^{2*} for

$N^* = 40\%$ (which occurs at about $t^* = 250$ s). A consistent reduction of the ongoing coefficient of determination is observed compared to the overall one. The maximum values related to RF methods are $R^2 = 0.89$ and $R^{2*} = 0.62$ whereas the ones related to NKKW are $R^2 = 0.92$ and $R^{2*} = 0.60$. Regarding the other tested methods, DTB provides lower performances than RF in an initial phase, then they become comparable, being both overtaken by KNNW ones at about $N^* = 10\%$. All the other methods provide lower results especially in terms of R^{2*}. In particular, SVMs prove to be very ineffective. Only the application of a Gaussian kernel shows a limited capability in forecasting the t_f, showing that the dataset cannot be effectively fitted by linear, quadratic or cubic kernel functions.

Regression results can be further investigated using the predicted-observed plots. Figures 6.6 and 6.7 provide these plots for the two best methods (RF and KNN, respectively). Plots are evaluated at 200 s and 500 s from the damage occurrence. Hence, the coloured region refers to the t_f forecast including all the ongoing damage scenarios ($t_f > t^*$). Among them, for the ones laying on the red area of the plot the t_f is overestimated, otherwise, it is underestimated. It can be noted that applying both the RF and KNNW learners, as time goes by, the predictions tend to group nearby the diagonal. Hence, the reliability of the response increases. However, as time goes by, the training database density in the forecast region decreases, leading to higher errors for the longer damage cases. This is the main cause of the decay of R^{2*}. It shall be noted that in such a region, non-parametric methods, such as decision trees, become less reliable since it is hard to define the sub-regions having constant response. This is why RF does not provide any prediction above 1104 s despite it applies multiple trees, as can be noted in Figure 6.6. KNNW has not such limitations, thus provides better performances with limited data taking advantage of the weighting process. Nevertheless, KNNW suffers by lacking data too, leading anyway to the R^{2*} decay, although somehow delayed.

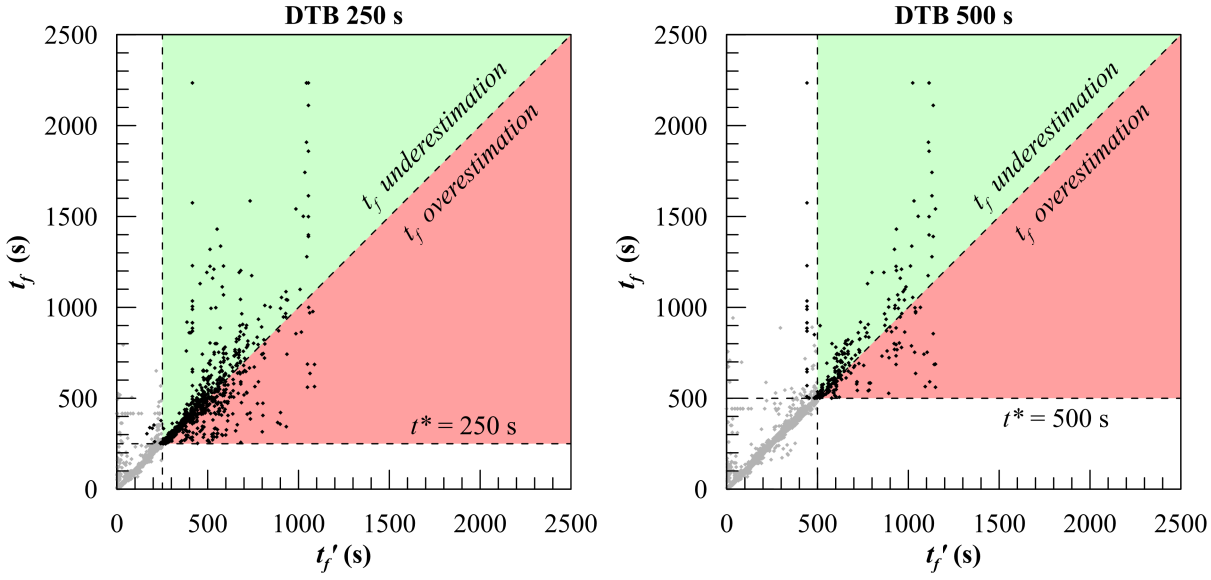

Figure 6.6 Predicted over obseved values of the t_f computed at $t^* = 250$ s and $t^* = 500$ s. Training: MC20; validation: MC50b; method: RF

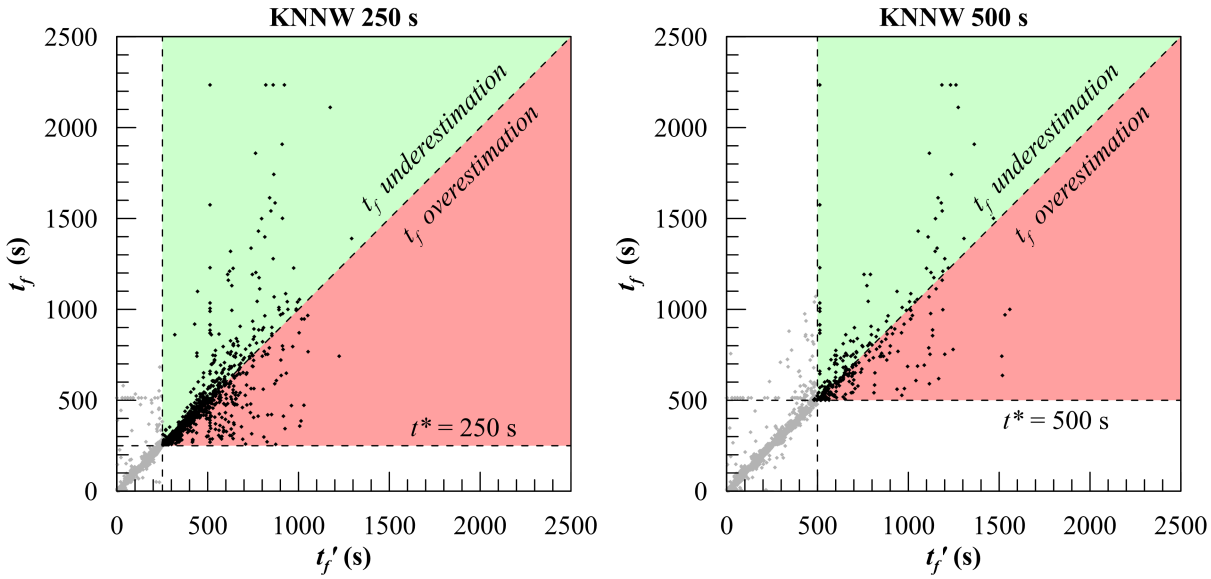

Figure 6.7 Predicted over obseved values of the time-to-flood computed at $t^* = 250$ s and $t^* = 500$ s. Training: MC20; validation: MC50b; method: KNNW

Figures 6.8 and 6.9 provides the t_f predictions related to the best two algorithms (RF and KNNW) for two damage cases from the validation database (MC50b): the DC051766 and the DC071109, respectively. Besides the t_f, figures includes also the predictors considered (dots in the heel angle, trim angle and sinkage plots). Damage case DC051766 involves rooms R23, R22, R21S and R11S, driving to asymmetric progressive flooding in two main compartments (1,2). As the floodwater reaches the upper deck, a stable position cannot be found anymore driving to the loss of the barge. Damage DC071109 involves rooms R53S, R43, R52 and R42. In this case, the progressive flooding is symmetric up to the flooding of R53S which lead to heel toward starboard and, then, capsize due to the insufficient residual stability of the barge. Both these damages are quite long, hence both RF and KNNW tend to underestimate the time-to-flood at the beginning of the progressive flooding. This is mainly due to the unknown dimension of damage above the waterline which cannot be accounted for in the initial phases of flooding. So, for very large damages in the upper part of the hull, an initial overestimation has been also observed. However, these damage scenarios show usually a very fast evolution, driving to immediate capsize. Thus, they are less relevant from a DSS point of view. After the initial phase, the two considered damage cases provide reliable predictions, even though RF slightly underestimate the t_f while KNNW overestimates it. Therefore, in a real operative environment, the crew might be forced to rush the evacuation procedures due to initial data and then relax the operations as the prediction have been updated. This behaviour might suggest providing results after a period sufficient to stabilise the outcomes of the machine-learning-based DSS.

In general, for all three problems, the distribution of the t_f in the training database plays a key role to ensure the forecast capabilities of the models, governing their decay. Obviously, this especially yields for the t_f regression, where a training database based on SOLAS probabilities

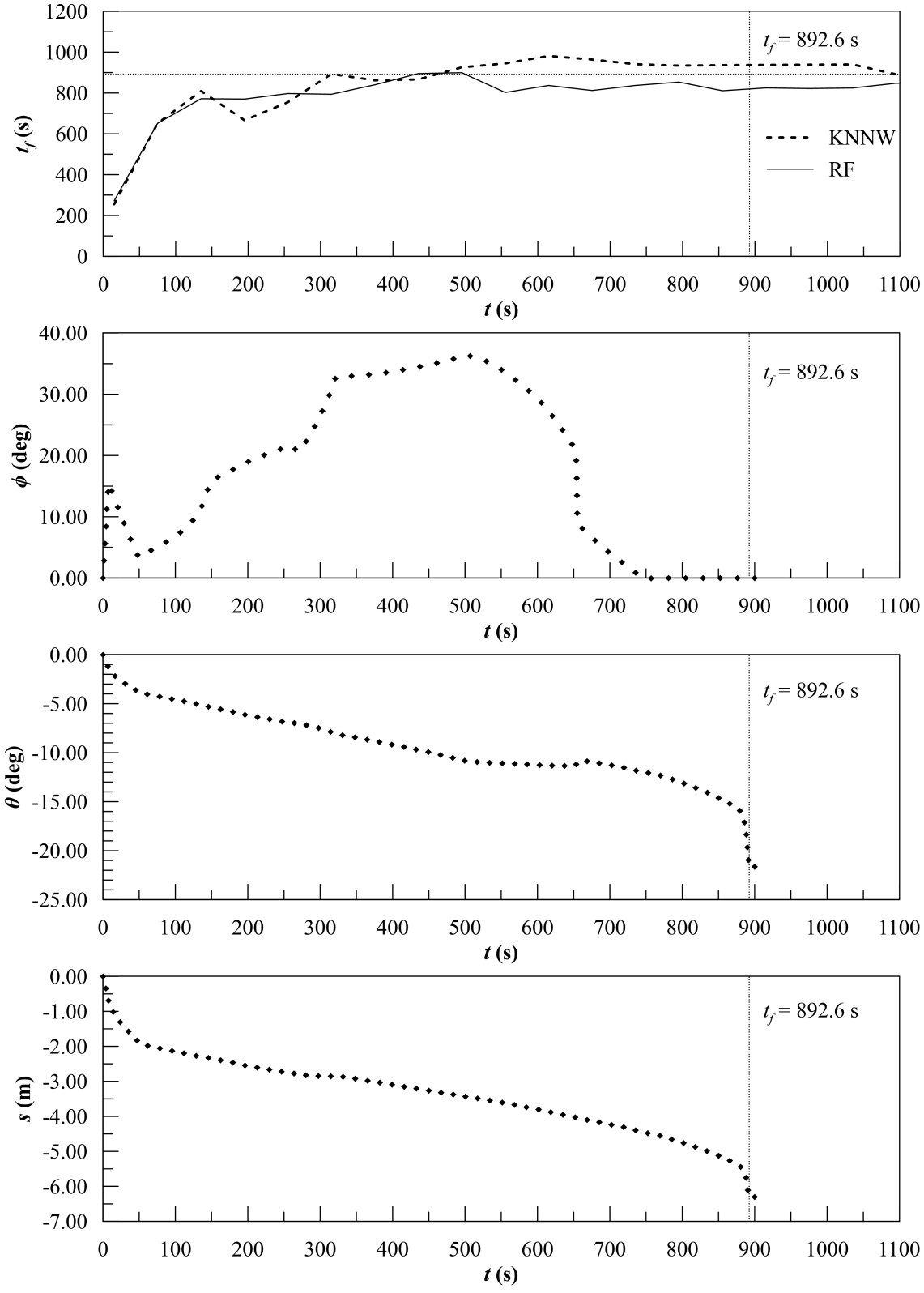

Figure 6.8 Time-to-flood prediction provided by the sysem based on RF and KNNW for damage case DC051766 according to the predictor values. Training: MC20

does not seem appropriate for the test geometry. Furthermore, RF appear to be the best choice in all three problems, proving that their resilience to noise and capability to deal with high-

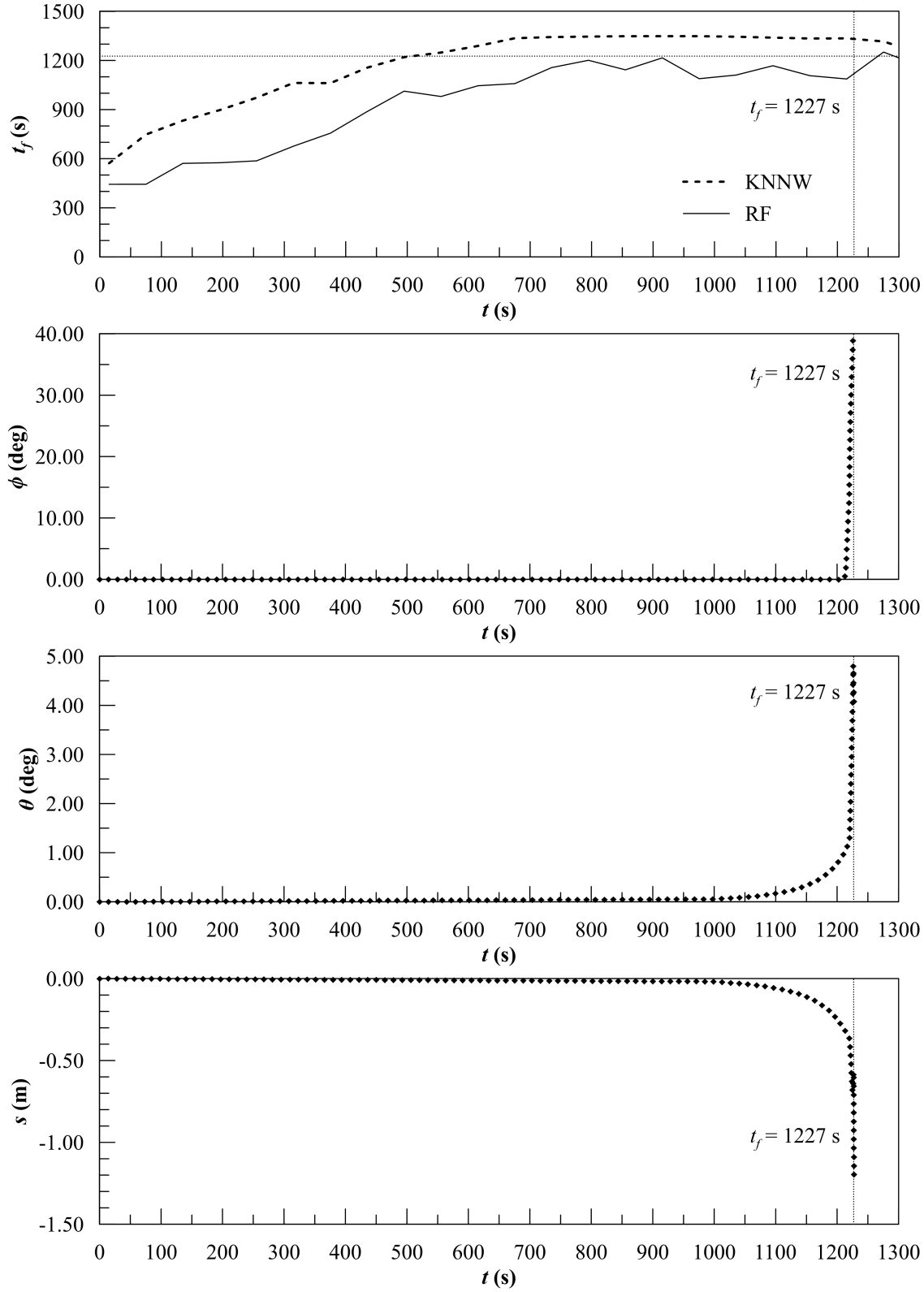

Figure 6.9 Time-to-flood prediction provided by the sysem based on RF and KNNW for damage case DC071109 according to the predictor values. Training: MC20

dimensional input spaces are of the utmost importance to forecast damage consequences from the time evolution of the floating position. In the present work, the constant dt = 15 s sampling

time has been adopted to enable a fast convergence of the learners' accuracy immediately after damage occurrence. However, the improvement of the performances obtained applying a $dt = 60$ s to SVMs suggests that a non-constant or t^* dependent sampling time step is worthy of further investigation in future works.

6.3 Tested database generation techniques

In the present section, the effect of the different database generation techniques presented in Chapter 5 is studied. Among all the learners tested in the previous section, here only the most effective one has been employed, i.e. RF with sampling time $dt = 15$ s. Besides, the different generation techniques have been applied to define the training database only. Here, the MC50b has been always employed as the validation database, since it is based on SOLAS probability distributions. As mentioned, such distributions have been developed with statistical analysis of real collision accidents, hence the MC50b database can be considered more representative of the damages that might occur to a ship during her operative life and, thus, more suitable for validation purposes.

6.3.1 Sensitivity on parametric generation

Before comparing the different database generation techniques, the effect of the number of divisions n_d on the RF performances has been studied. In fact, the assessment of minimum size related to MC generation does not yield for the parametric method, which is governed by n_d. To assess the minimum n_d for the test geometry, the following databases have been independently grown up limited to two adjoining damaged compartments:

- P3: having $n_d = 3$ and being composed of 1975 damage cases (1427 non-survival cases);
- P4: having $n_d = 4$ and being composed of 3486 damage cases (2615 non-survival cases);
- P5: having $n_d = 5$ and being composed of 5719 damage cases (4405 non-survival cases);
- P6: having $n_d = 6$ and being composed of 8874 damage cases (7019 non-survival cases);
- P7: having $n_d = 7$ and being composed of 13170 damage cases (10606 non-survival cases);
- P8: having $n_d = 8$ and being composed of 18854 damage cases (15388 non-survival cases).

Again the maximum simulated time has been set to 2250 s, being the TE cases about 0.8% of the total for the test geometry.

Assuming a $dt = 15$ s and employing the RF learner, all the databases have been used as training database. Figure 6.10 provides the results of the analysis.

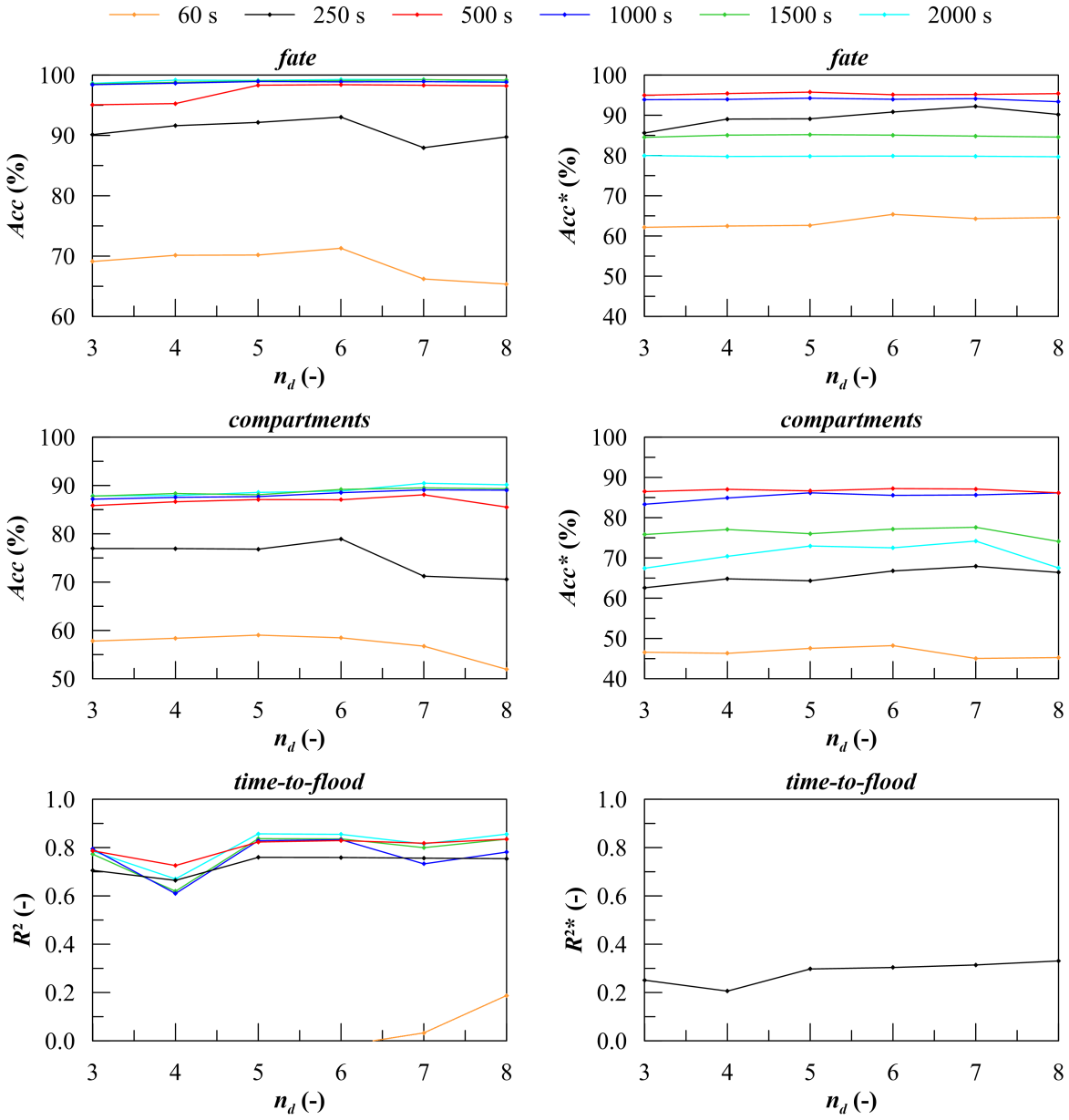

Figure 6.10 Performances evaluated at different time instants t^* as function of number of division n_d of the training database. Validation: MC50b; method: RF

It is worth noticing that the performances are almost insensible to the variation of n_d in the range $[3,8]$. Hence, quite good performances can be obtained with a very limited number of damage scenarios applying the parametric generation of training database (only 1975 for $n_d = 3$). However, the performances are inferior to the ones obtained with MC20 training database. Such a topic will be discussed in more detail later, when all the generation methods will be compared.

Only the regression on t_f shows some preference for the highest values of n_d, especially for short t^*, and a quite unstable behaviour for $n_d \leq 4$. For this reason, and also to compare results related to training databases having similar size, P8 database has been selected to be compared with other MC-based generation techniques.

6.3.2 Comparison of database generation techniques

The different techniques defined in Chapter 5 have been applied to grow-up four different training databases:

- MC20: based on SOLAS proability distributions and being composed of 20000 damage cases (3082 non-survival cases);
- MCD20: based on uniform distribution of damage dimensions and being composed of 20000 damage cases (8898 non-survival cases);
- MCA20: based on uniform distribution of damage area inverse and being composed of 20000 damage cases (2318 non-survival cases);
- P8: parametric database having $n_d = 8$ and being composed of 18854 damage cases (15,388 non-survival cases).

As mentioned, the MC20 and P8 databases have been selected among the multiple options to assure a good convergence of the RF performances and a fair comparison in terms of N. For the same reason, 20,000 damage cases have been included in MCD20 and MCA20 databases. Figure 6.11 shows the resulting cumulative density functions of the main dimensions, location and time-to-flood related to the four databases.

As long as the MC-based methods provide continuous distribution of all the parameters, the parametric one leads to scattered distributions except for the t_f.

As expected the MCA20 database has an almost uniform distribution of the t_f associated with small damages located nearby the free surface. Besides, the number of non-survival scenarios is lower than the MC20 one. On the other hand, both the MCD20 and P8 databases contain a considerably higher percentage of non-survival scenarios. Although the probability distributions of damage parameters are different, MC20 and MCD20 drive to a comparable cumulative distribution of the t_f.

Figure 6.12 shows the overall performances obtained applying the different training databases. It is worth noticing that the MC20 database employing the SOLAS probability distributions is not always associated with the best performances despite a SOLAS based dataset is always employed for validation.

In detail, the MCD20 database provides almost the same results for FF and FC classification problems compared to the MC20 one. Besides, MCD20 dataset is more accurate in the prediction of the t_f. The MCA20 database provides lower performances than MC20 and MCD20 ones for all the studied problems. However, although a lower maximum of R^2 and R^{2*} is reached, the MCA20 shows a larger region of stable t_f forecast: the R^{2*} decays under a null value at about $t = 1215$ s (corresponding to the 0.5% of ongoing damage scenarios in the validation database) instead of $t = 735$, which is related to MCD20 database. Regarding the parametric generation method, it always leads to poorer forecast capability compared to all the methods based on MC

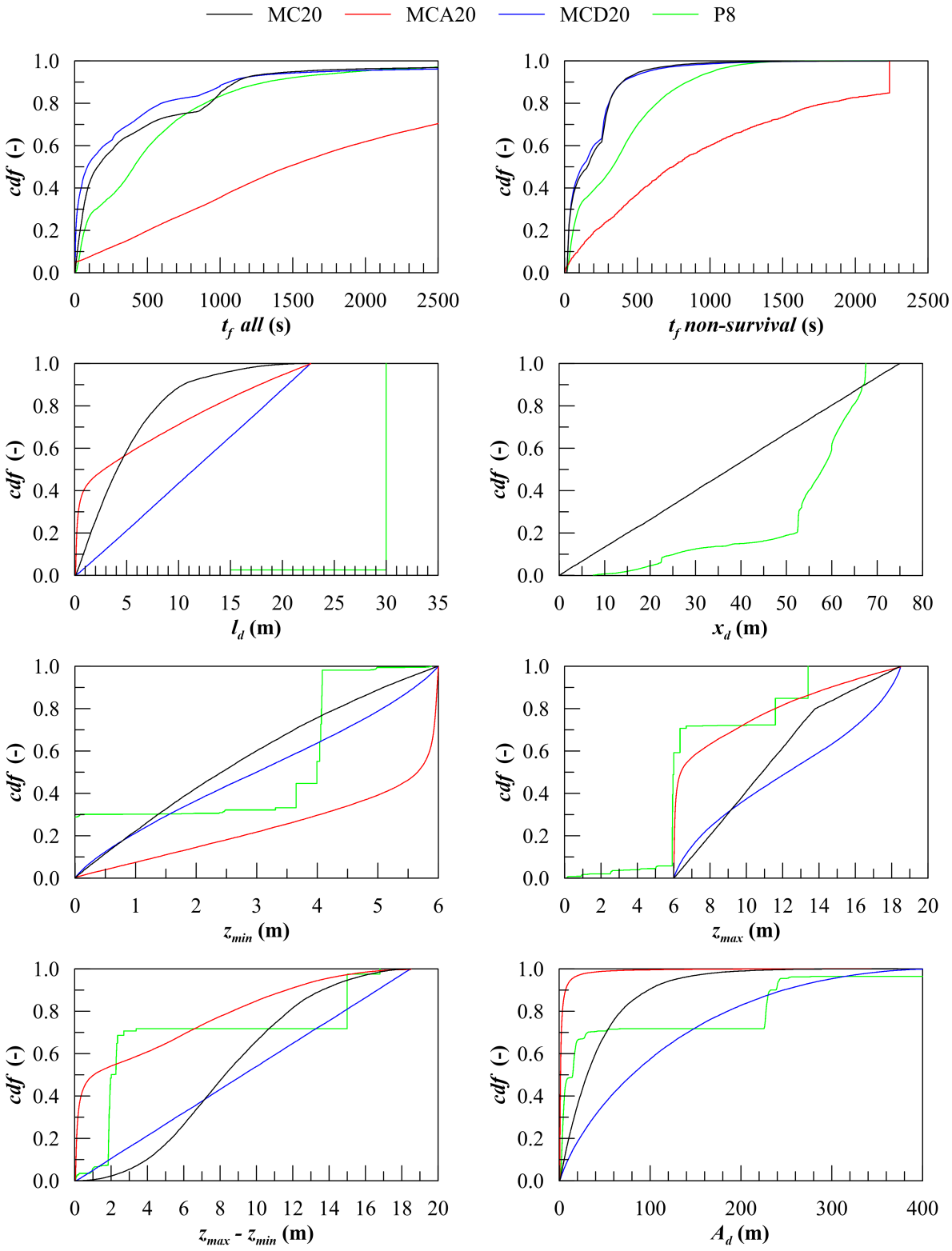

Figure 6.11 Cumulative density functions of damage dimensions, location and time-to-flood related to the tested database generation methods

generation. Moreover, a certain degree of accuracy instability has been observed for both the classification problems.

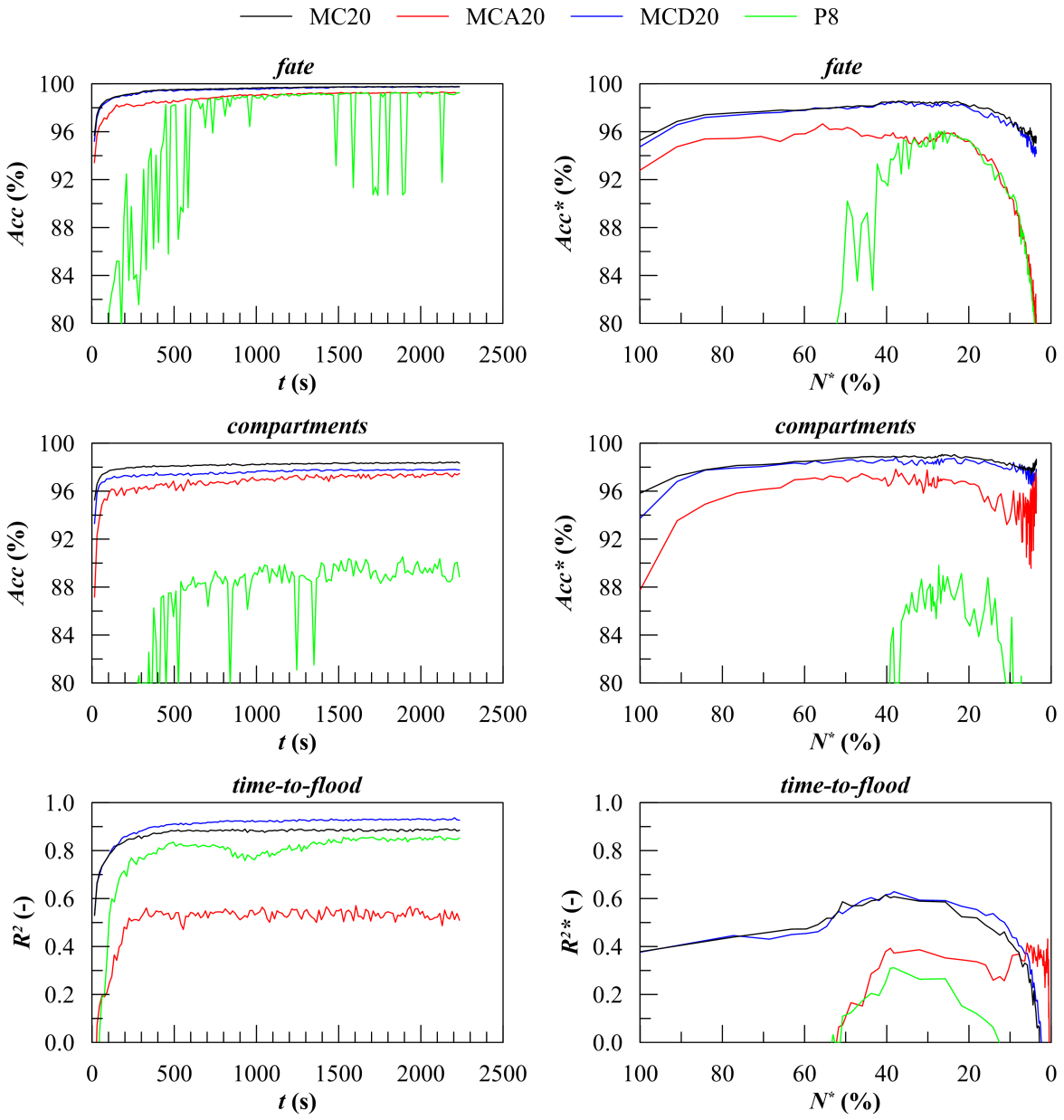

Figure 6.12 Comparison of the performances of tested database generation methods. Validation: MC50b; method: RF

6.3.3 Discussion

Regarding the first classification problem (FF), the adoption of MCD20 training database has a very limited effect on both the overall and ongoing accuracy. In detail, the larger number of non-survival damage cases leads to a less skewed training dataset, which slightly reduces the type I error (about 0.05%) while increasing the type II one (about 0.15%). The MCA20 training database is about 2% less accurate than MC20/MCD20 in forecasting the final fate. Moreover, it shows a strong decay of the ongoing accuracy for long damage scenarios. The main reason can be found in the larger number of damage scenarios exceeding the maximum simulation time compared to the other MC-based datasets: in MCA20, 30% of damage sce-

narios has $t_f > 2250$ s. Similar behaviour has been observed employing P8 training database, although the accuracy reaches an almost stable value later (at about 500 s). It shall be noticed that P8 database, comprises only damage cases affecting one or two adjoining compartments. On the other end, the validation database (MC50b) is based on SOLAS probability distribution. The SOLAS maximum damage length is 22.72 m, which exceeds 15 m, i.e. the length of each watertight compartment of the test barge. Thus, 7.4% of damage scenarios in the validation database affects 3 adjoining compartments. Considering that the accuracy gap for the FF classification problem in the stable region is between 1% and 3%, many of the three compartments damage scenarios are still correctly classified by RF trained with the P8 database.

On the other end, the classification of flooded compartments leads to an accuracy gap of about 9-10%, which is mostly due to the three compartments damage scenarios included in the validation database. For this second problem, the MCD20 and MCA20 databases have an overall accuracy 1% and 2% lower than the MC20 one, respectively. On the other end, the ongoing accuracy of MCD20 is comparable to the MC20 one whereas the MCA20 one is again 2% worse and shows a greater instability for large t^* value.

More interesting results have been obtained for the t_f regression in non-survival scenarios based on the different techniques for training database generation. Figures 6.13-6.16 shows the predicted-observed plots evaluated at $t^* = 500$ s and $t^* = 1000$ s.

As mentioned, the MCD20 training database leads to better results compared to the ones coming from the SOLAS probability distributions. MCD20 contains more than double non-survival scenarios compared to MC20, while the two databases have almost the same distribution of the time to flood. The resulting increased density of the training database allows the RF to better forecast the t_f up to $t^* = 1700$ s. Even better results can be reached at larger t^* values by employing the MCA20 training database: values are forecasted up to $t^* = 2000$ s. These results are achieved since the lower dimension of the generated damages leads to a higher density of training data in the most critical region. However, the model tends to overestimate the t_f as shown by Figure 6.14.

Besides, due to the limited number of non-survival damage scenarios, RF trained with MCA20 are not capable to deal with the largest damages having short t_f: bulky errors have been observed in this region leading also to poor values of the overall R^2, which is never greater than 0.55. Finally, it shall be noted that the parametric generation method is not capable to assure good results for the t_f regression problem.

In conclusion, the present test case proofs that the training database generation based on SOLAS probability distributions is not always the best option. Although a SOLAS-based validation dataset has been always employed, different distributions of damage parameters driving the MC sampling enable achieving equal or better results. The most sensible problem among the studied is the t_f regression, where uniform distributions of damage dimensions or inverse area improve the performances for large t^* values. Besides, the two adjoining compartments parametric generation seems to have quite lower performances for the studied geometry.

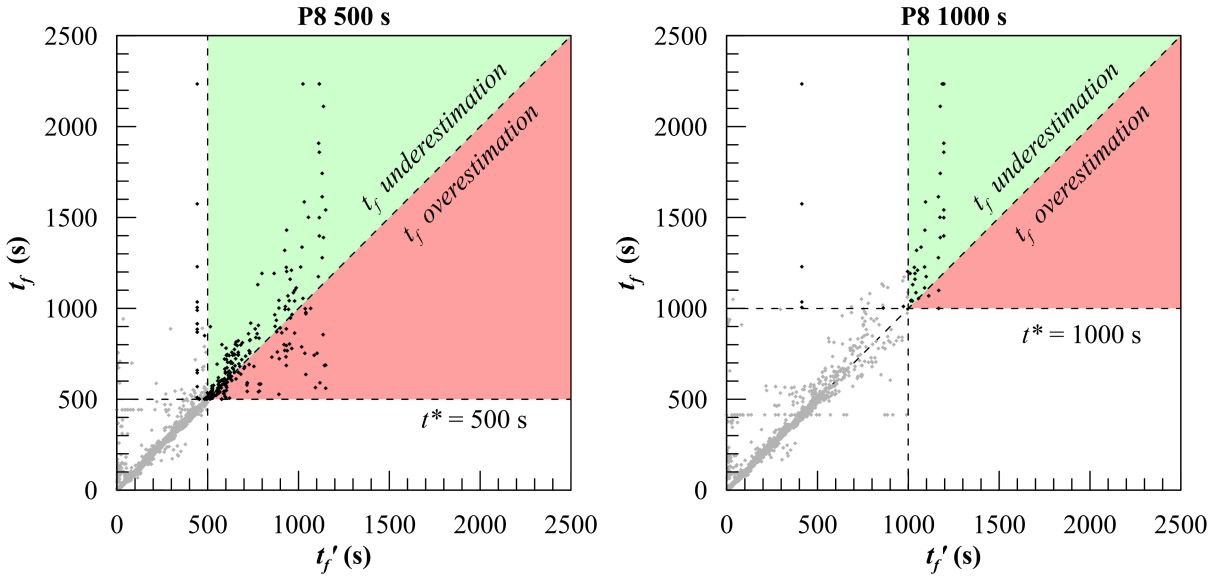

Figure 6.13 Predicted over obseved values of the t_f computed at $t^* = 500$ s and $t^* = 1000$ s. Training: MC20; validation: MC50b; method: RF

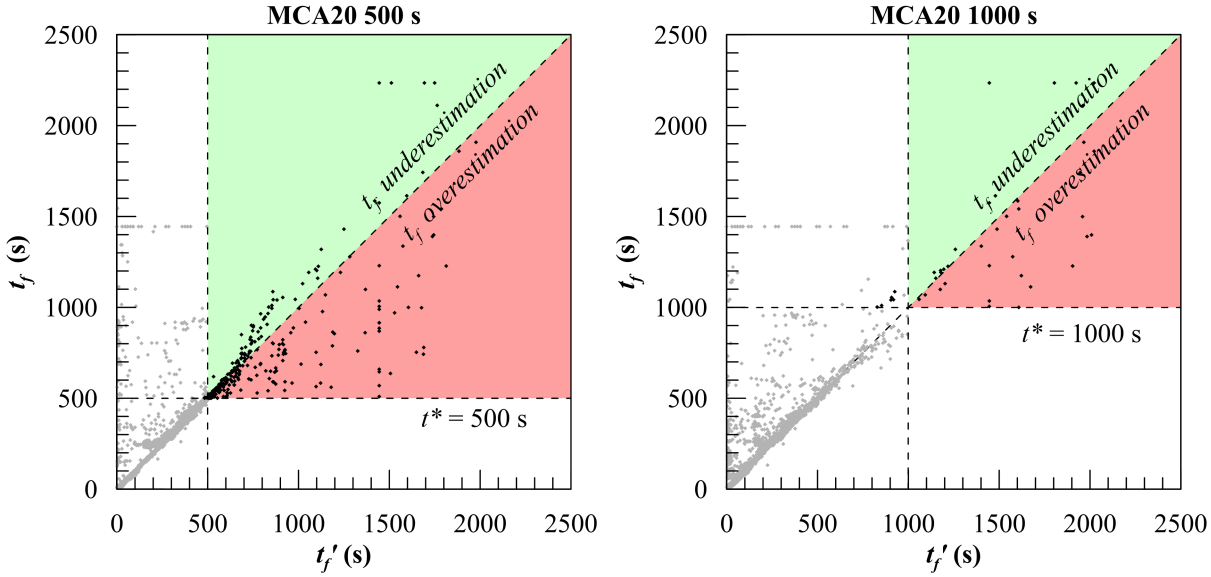

Figure 6.14 Predicted over obseved values of the time-to-flood computed at $t^* = 500$ s and $t^* = 1000$ s. Training: MCA20; validation: MC50b; method: RF

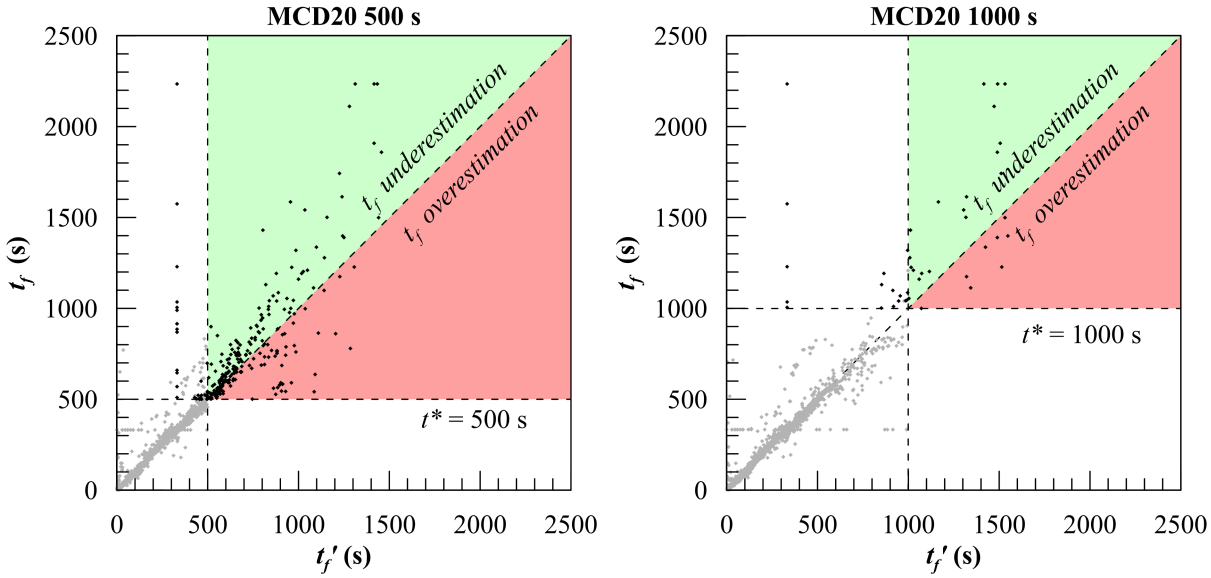

Figure 6.15 Predicted over obseved values of the time-to-flood computed at $t^* = 500$ s and $t^* = 1000$ s. Training: MCD20; validation: MC50b; method: RF

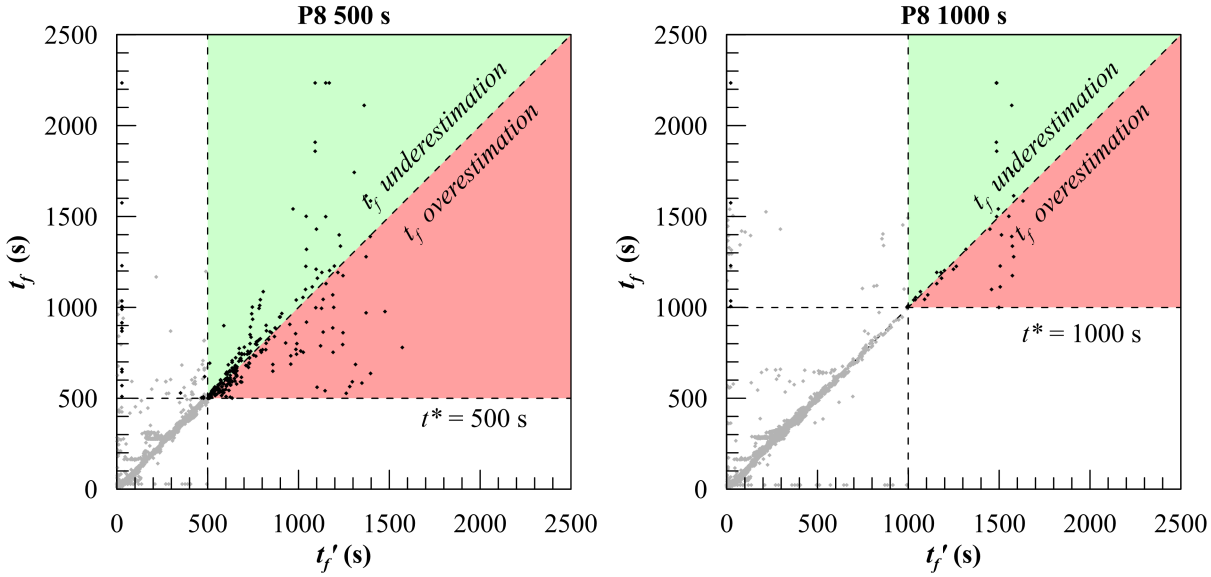

Figure 6.16 Predicted over obseved values of the time-to-flood computed at $t^* = 500$ s and $t^* = 1000$ s. Training: P8; validation: MC50b; method: RF

Chapter 7

Conclusions and Perspectives

The progressive flooded simulation and decision support are key aspects during a flooding emergency. The reliability and speed of direct simulations are essential to assure that the onboard systems are capable to provide good guidance to the master and officers to reduce the risk for the vessel and people onboard. In this context, the adoption of modern emergency DSS is hindered on vessels not equipped with a widespread flooding detection system. The present dissertation explored an alternative solution to provide decision support based on a very limited sensors' set by exploiting machine learning algorithms on a database of progressive flooding simulations. In the following, the main conclusions of the dissertation are presented along with some recommendations regarding future works in this new field.

7.1 Conclusions

The present dissertation has presented a new quasi-static method for progressive flooding simulation. Among the different studied mathematical formulations, the one based on linearised differential-algebraic equations provided reliable results compared to model scale experiments. Furthermore, it has been proved capable to deal with large passenger vessels and their very complex and challenging layout.

The application of an adaptive integration time step improves significantly the computational time compared to the constant one. The upper limitation of time step according to the estimated submersion time of openings and acceptable variation of the floating position reduced the errors, allowing to considerably extend the integration time, only when appropriate. A similar scheme might be applied also to other integration techniques available in the literature to improve their computational performances while limiting the drawbacks associated with too large time steps.

The new simulation technique has been also found satisfactory to build up large datasets of progressive flooding simulations, being a good basis for the second part of the present dissertation. There, on a simple geometry, the feasibility has been proved of the recognition and prediction of main flooding consequences based on the time evolution of the floating position.

Focusing on the three studied problems tested on a barge geometry with SOLAS-based training and validation datasets, the best results have been obtained applying the random forests for the classification of final fate and flooded compartments. Considering the regression on the time-to-flood, random forests should be preferred in an initial phase, whereas the weighted k-nearest neighbours provided better results for the longest damage scenarios.

In such a region, a decay of learners performances has been detected for all the studied problems, although with different magnitude. It is mainly due to the limited number of large time-to-flood damage cases included in the training database built according to SOLAS probability distribution. This is why other database generation strategies have been tested to build up training databases having a higher density of the input space in the large time-to-flood region. The tested options showed that SOLAS probability distributions are not always the best option for random forest training. In particular, considering the time-to-flood regression problem, the increased number of non-survival damage cases obtained applying uniform distribution of damage dimensions (less skewed dataset) lead to some gains of the ongoing accuracy. Besides, the uniform distribution of the inverse damage area, increasing substantially the percentage of long damage scenarios, led to heavily improve forecast capability of the models trained at large t^* values, which are the most interesting from an onboard decision support point of view. However, the accuracy for fast damage scenarios was affected.

Considering the parametric database generation algorithm, the results suggest that it is not suitable for the studied problems on the test geometry. However, it demonstrates to work with very small datasets (low number of divisions) and showed a certain capability of going beyond its intrinsic limits related to the hypothesis of two adjoining compartments only.

According to these conclusions the main scientific contribution of the present dissertation are:

- the development of a fast quasi-static method to simulate the progressive flooding based on the linearization of the governing equations;
- the development of an adaptive integration scheme applicable to progressive flooding simulations algorithms preventing the errors due to too large integration time step;
- the development of a methodology to predict the flooding consequences based on the time record of the floating position of the damaged ship;
- the selection of the random forests as the most promising machine-learning algorithm to forecast the ship final fate, flooded compartments and time-to-flood, among the tested ones;
- the satisfactory prediction of the ship final fate and flooded compartments employing a training database built according to SOLAS probability distributions for damage parameters (side collision damage);

- the demonstration of the strong effect of the training database composition on the prediction of the time-to-flood;
- in conclusion, the feasibility demonstration of machine-learning-based DSS to help masters and officers during a flooding emergency without any flooding detection device.

7.2 Perspectives

The future works should focus on two main topics: increasing the speed and reliability of onboard progressive flooding simulation algorithms and better explore the application of machine learning to assess flooding consequences. Regarding fast progressive flooding simulations, the main bottleneck in the present algorithm is represented by the evaluation of equilibrium and hydrostatics, which are based on the pressure integration technique. The replacement with an Archimedes-based hydrostatic code could further improve the computational performances. Besides, the current method is based on an iterative equilibrium process which cannot accounts for any dynamic phenomena. Since the roll angle is the most critical parameter in the equilibrium evaluation, the dynamic modelling of roll angle might improve both the accuracy and the computational load.

Regarding the application of machine learning, further work is advisable to investigate the effect of the training database composition. In particular, the parametric generation method shall be further addressed to overcome the issues related to the two-adjoining-compartments hypothesis. Besides, further research is still essential to develop and install onboard a flooding-sensor-agnostic DSS. First, the proposed methodology has been developed and tested on simple geometry and a single loading condition. Hence, it is required to test the application in real full-scale environments such as a cargo vessel and then a large cruise vessel. Furthermore, the effect of multiple loading conditions should be studied as well as the effect of the uncertainties that might affect the mass centre position, displacement, metacentric height, discharge coefficients and leakage/collapse effective waterheads in a real environment.

Moreover, here a constant sampling time has been employed to define the predictors (heel, trim, sinkage). A smart selection of a smaller predictors' set, carried out by dimensionality reduction or optimization, might further improve the forecast accuracy and, thus, it is worthy of investigation.

Eventually, although the present dissertation tested a quite large set of supervised learners and identified the preferred among them, it did not pretend to thoroughly cover all the algorithms present in the literature. Hence, other families of learners, such as neural networks, might be tested in future works. However, it shall be bared in mind that the present dissertation is an explorative study on the application of machine learning to provide decision support during flooding emergencies without flooding sensors. The main objective was to prove the feasibility of such a concept. In light of the provided results, such an objective can be considered achieved.

Bibliography

[1] M. Acanfora, E. Begovic, and F. De Luca. A fast simulation method for damaged ship dynamics. *J. of Mar. Sci. Eng.*, 7(4):111, 2019.

[2] M. Acanfora and A. Cirillo. A simulation model for ship response in flooding scenario. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, 231(1):153–164, 2017.

[3] E. Alpaydin. *Introduction to machine learning*. The MIT Press, Cambridge, USA, 3rd edition edition, 2014.

[4] I. Bačkalov, G. Bulian, A. Rosén, V. Shigunov, and N. Themelis. Improvement of ship stability and safety in intact condition through operational measures: challenges and opportunities. *Ocean Engineering*, 120:353–361, 2016.

[5] E. Boulougouris, J. Cichowicz, A. Jasionowski, and D. Konovessis. Improvement of ship stability and safety in damaged condition through operational measures: Challenges and opportunities. *Ocean Engineering*, 122:311–316, 2016.

[6] L. Braidotti, G. Degan, S. Bertagna, V. Bucci, and A. Marinò. A comparison of different linearized formulations for progressive flooding simulations in full-scale. *Procedia Computer Science*, 180:219–228, 2021. Proceedings of the 2nd International Conference on Industry 4.0 and Smart Manufacturing (ISM 2020).

[7] L. Braidotti, A. Marinò, and V. Bucci. On the effect of uncertainties on onboard progressive flooding simulation. In *Proceedings of The 3rd International Conference on Nautical and Maritime Culture - CNM 2019*, pages 21–30, Naples, Italy, 2019.

[8] L. Braidotti and F. Mauro. A new calculation technique for onboard progressive flooding simulation. *Ship Technology Research*, 66(3):150–162, 2019.

[9] L. Braidotti and F. Mauro. A fast algorithm for onboard progressive flooding simulation. *Journal of Maritime Science and Engineering*, 8:369, 2020.

[10] L. Braidotti, J. Prpić-Oršić, and M. Valčić. Application of decision trees to predict damage consequences during the progressive flooding. In *Proceedings of the 8th International Conference on marine Structures*, Trondheim, Norway, 2021.

[11] L. Braidotti, L. Sebastiani, S. Bisiani, and V. Bucci. A ballast allocation to minimise fuel consumption. In *Proceedings of The 19th International Conference on Ships and Maritime Research - NAV 2018*, pages 97–104, Trieste, Italy, 2018.

[12] L. Braidotti, G. Trincas, and V. Bucci. Analysis of the influence of pressure field on accuracy for onboard stability codes. In *Proceedings of The 19th International Conference on Ships and Maritime Research - NAV 2018*, pages 80–87, Trieste, Italy, 2018.

[13] L. Braidotti, M. Valčić, and J. Prpić-Oršić. Exploring a flooding-sensors-agnostic prediction of the damage consequences based on machine learning. *Journal of Marine Science and Engineering*, 9(3):271, 2021.

[14] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[15] L. Breiman, J. Friedman, Stone C.J., and R.A. Olshen. *Classification and Regression Trees*. CRC Press, Boca Raton, USA, 1984.

[16] S. Bu and M. Gu. Unified viscous and potential prediction method for the coupled motion of damaged ship and floodwater in calm water. *Ocean Engineering*, 210:107441, 2020.

[17] G. Bulian, M. Cardinale, A. Francescutto, and G. Zaraphonitis. Complementing SOLAS damage ship stability framework with a probabilistic description for the extent of collision damage below the waterline. *Ocean Engineering*, 186:106073, 2019.

[18] G. Bulian, D. Lindroth, P. Ruponen, and G. Zaraphonitis. Probabilistic assessment of survivability in case of grounding: Development and testing of a direct non-zonal approach. In *Proceedings of the 12th International Conference on the Stability of Ships and Ocean Vehicles*, Glasgow, UK, 2015.

[19] G. Bulian, D. Lindroth, P. Ruponen, and G. Zaraphonitis. Probabilistic assessment of damaged ship survivability in case of grounding: development and testing of a direct non-zonal approach. *Ocean Engineering*, 120:331–338, 2016.

[20] F. Calabrese, A. Corallo, A. Margherita, and A.A. Zizzari. A knowledge-based decision support system for shipboard damage control. *Expert Systems with Applications*, 39(9):8204–8211, 2012.

[21] A. Caldas, C. Zegos, S. Skoupas, and J. Jenkins. Ship survivability study using high fidelity cfd. In A. Marinò and V. Bucci, editors, *Proceedings of The 19th International Conference on Ships and Maritime Research - NAV 2018*, Trieste, 2018. IOS Press.

[22] D. Cangelosi, A. Bonvicini, M. Nardo, A. Mola, A. Marchese, M. Tezzele, and G. Rozza. Srtp 2.0 - the evolution of the safe return to port concept. In *Proceedings of The 19th International Conference on Ships and Maritime Research - NAV 2018*, pages 665–672, Trieste, Italy, 2018.

[23] B.-C. Chang and P. Blume. Survivability of damaged ro-ro passenger vessels. *Ship Technology Research*, 45(3):105–117, 1998.

[24] S. Chatterjee and J. S. Simonoff. *Multiple Linear Regression*. John Wiley & Sons, Hoboken, USA, 2013.

[25] H. Cheng, A.M. Zhang, and F.R. Ming. Study on coupled dynamics of ship and flooding water based on experimental and sph methods. *Physics of Fluids*, 29(10), 2017.

[26] J. Choi, D. Lee, H.J. Kang, S.Y. Kim, and S.C. Shin. Damage scenarios and an on-board support system for damaged ships. *International Journal of Naval Architecture and Ocean Engineering*, 6(2):236–244, 2014.

[27] N. Christianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, Cambridge, UK, 2000.

[28] H. Dankowski. *A Fast and Explicit Method for Simulating Flooding and Sinkage Scenarios of Ships*. PhD thesis, Technischen Universität Hamburg, Hamburg, Germany, 2013.

[29] H. Dankowski and S. Krüger. A fast, direct approach for the simulation of damage scenarios in the time domain. In *Proceedings of the 11th International Marine Design Conference - IMDC 2012*, Glasgow, UK, 2012.

[30] H. Dankowski, P. Russel, and S. Krüger. New insights into the flooding sequence of the costa concordia accident. In *Proceedings of the 33rd International Conference on Ocean, Offshore and Arctic Engineering - OMAE 2014*, San Francisco, USA, 2014.

[31] M.P. Deisenroth, A.A. Faisal, and C.S. Ong. *Mathematics for machine Learning*. Cambridge University Press, Cambridge, UK, 2020.

[32] J. Dillingham. Motion studies of a vessel with water on deck. *Marine Technology*, 18(1), 1981.

[33] S.A. Dudani. The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6(4):325–327, 1976.

[34] EMSA. Annual overview of marine casualties and incidents 2020. Technical report, European Maritime Safety Agency, Lisbon, Portugal, 2021.

[35] S. Escalera, O. Pujol, and P. Radeva. Separability of ternary codes for sparse designs of error-correcting output codes. *Pattern Recognition Letters*, 30(3):285–297, 2009.

[36] T. Fujiwara and T. Haraguchi. Roll motion of ro-ro passenger ship with flooded vehicle deck. *International Journal of Offshore and Polar Engineering*, 15(2):109–116, 2005.

[37] Z. Gao, Q. Gao, and D. Vassalos. Numerical simulation of flooding of a damaged ship. *Ocean Engineering*, 38(14):1649 – 1662, 2011.

[38] DNV GL. Evaluation of risk from raking damages due to grounding. Technical Report PP090623/3-1, DNV GL, Høvik, Norway, 2014.

[39] M.S. Grewal and A.P. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB*. Wiley & Sons, Hoboken, USA, 3rd edition edition, 2008.

[40] H. Hashimoto, K. Kawamura, and M. Sueyoshi. A numerical simulation method for transient behavior of damaged ships associated with flooding. *Ocean Engineering*, 143:282 – 294, 2017.

[41] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, New York, USA, 2nd edition edition, 2008.

[42] M. Hossin and M.N. Sulaiman. A review on evaluation metrics for data classification evaluation. *International Journal of Data Mining & Knowledge Management Process*, 5(2):1–11, 2015.

[43] L.F. Hu and K. Ma. Genetic algorithm-based counter-flooding decision support system for damaged surface warship. *International Shipbuilding Progress*, 55(4):301–315, 2008.

[44] IACS. UR L5 Rev3 CLN Computer software for onboard stability calculations. Technical report, International Association of Classification Societies, London, UK, 2017.

[45] IMO. SLF 47/INF.6 Survivability investigation of large passenger ship. Technical report, International Maritime Organisation, London, UK, 2004.

[46] IMO. Resolution MSC.216(82) amendments to the international convention for the safety of life at sea, 1974, as amended. Technical report, International Maritime Organisation, London, UK, 2006.

[47] IMO. MSC.1/Circ.1245 guidelines for damage control plans and information to the master. Technical report, International Maritime Organisation, London, UK, 2007.

[48] IMO. MSC.1/Circ.1291 guidelines for flooding detection systems on passenger ships. Technical report, International Maritime Organisation, London, UK, 2008.

[49] IMO. Resolution MSC.281(85) explanatory notes to the SOLAS Chapter II-1 subdivision and damage stability regulations. Technical report, International Maritime Organisation, London, UK, 2008.

[50] IMO. MSC.1/Circ.1400 guidelines on operational information for masters of passenger ships for safe return to port by own power or under tow. Technical report, International Maritime Organisation, London, UK, 2011.

[51] IMO. *SOLAS 2018 Consolidated Edition*, chapter Ch.II-1 Part B Subdivision and stability. International Maritime Organisation, London, UK, 2018.

[52] ITTC. Numerical simulation of capsize behaviour of damaged ship in irregular seas. Technical report, International Towing Tank Conference, 2017.

[53] Risto Jalonen, Pekka Ruponen, Mateusz Weryk, Hendrik Naar, and Sander Vaher. A study on leakage and collapse of non-watertight ship doors under floodwater pressure. *Marine Structures*, 51:188 – 201, 2017.

[54] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. Springer, New York, USA, 2017.

[55] A. Jasionowski. Decision support for ship flooding crisis management. *Ocean Engineering*, 38(14):1568–1581, 2011.

[56] T. Jo. *Machine Learning Foundations*. Springer, Cham, Switzeraland, 2021.

[57] A.V. Joshi. *Machine Learning and Artificial Intelligence*. Springer, Cham, Switzeraland, 2020.

[58] H.J. Kang, J. Choi, G.T. Yim, , and Haeseong A. Time domain decision-making support based on ship behavior monitoring and flooding simulation database for on-board damage control. In *Proceedings of the 27th International Ocean and Polar Engineering Conference*, San Francisco, USA, 2017.

[59] K.B. Karolius, J. Cichowicz, and D. Vassalos. Risk-based positioning of flooding sensors to reduce prediction uncertanty of damage survivability. In *Proceedings of the 13th International Conference on the Stability of Ships and Ocean Vehicles - STAB 2018*, pages 627–637, Kobe, Japan, 2018.

[60] J.H. Klote and J.A. Milke. *Principles of smoke management*. American Society of Heating, Refrigerating and Air-conditioning Engineers, Atlanta, USA, 2002.

[61] S. Kruger and H. Dankowsky. A monte carlo based simulation method for damage stability problems. In *Proceedings of the 38th International Conference on Ocean, Offshore and Arctic Engineering - OMAE 2019*, Glasgow, UK, 2019.

[62] D. Lee. Knowledge-based system for safety control of damaged ship. *Knowledge-Based Systems*, 19(3):187–191, 2006.

[63] D. Lee, S.S. Lee, B.J. Park, and S.Y. Kim. A study on the framework for survivability assessment system of damaged ships. *Ocean Engineering*, 32(8):1122–1132, 2005.

[64] G.J. Lee. A study on the dynamic orifice equation for the flooding simulation of a ship. *Journal of Ships & Ocean Engineering*, 55:17–27, 2014.

[65] G.J. Lee. Dynamic orifice flow model and compartment models for flooding simulation of a damaged ship. *Ocean Engineering*, 109:635–653, 2015.

[66] T. Manderbacka, T. Mikkola, P. Ruponen, and J. Matusiak. Transient response of a ship to an abrupt flooding accounting for the momentum flux. *Journal of Fluids and Structures*, 57:108–126, 2015.

[67] P.T. Martins and V.S. Lobo. Real-time decision support system for managing ship stability under damage. In *OCEANS 2011 IEEE - Spain*, pages 1–7, Santander, Spain, 2011.

[68] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. The MIT Press, Cambridge, USA, 2nd edition edition, 2018.

[69] A.İ. Ölçer and J. Majumder. A case-based decision support system for flooding crises onboard ships. *Quality and Reliability Engineering International*, 22(1):59–78, 2006.

[70] A. Papanikolaou. Review of damage stability of ships - recent developments and trends. In *Proceedings 10th International Symposium on Practical Design of Ships and Other Floating Structures (PRADS) 2007*, Houston, USA, 2007.

[71] A. Papanikolaou and D. Spanos. On the stability of fishing vessels with trapped water on deck. *Ship Technology Research*, 48(3):124 – 133, 2001.

[72] A. Papanikolaou, G. Zaraphonitis, D. Spanos, E. Boulougouris, and E. Elipoulou. Investigation into the capsizing of damaged ro–ro passenger ships in waves. In *Proceedings of the 7th International Conference on Stability of Ships and Ocean Vehicles*, Launceston, 2000.

[73] P. Pennanen, P. Ruponen, J. Nordström, and F. Goerlandt. Application of vessel TRIAGE for a damaged passenger ship. In *Proceedings of the 15th International Ship Stability Workshop*, pages 141–146, Stockholm, Sweden, 2016.

[74] P. Penttilä and P. Ruponen. Use of level sensors in breach estimation for a damaged ship. In *Proceedings of the 5th International Conference on Collision and Grounding of Ships*, pages 80–87, Espoo, Finland, 2010.

[75] G. Rebala, A. Ravi, and S. Churiwala. *An Introduction to Machine Learning*. Springer, Cham, Switzeraland, 2019.

[76] J.M. Rodrigues and C. Guedes Soares. A generalized adaptive mesh pressure integration technique applied to progressive flooding of floating bodies in still water. *Ocean Engineering*, 110:140–151, 2015.

[77] J.M. Rodrigues, A. Lavrov, M.A. Hinostroza, and C. Guedes Soares. Experimental and numerical investigation of the partial flooding of a barge model. *Ocean Engineering*, 169:586–603, 2018.

[78] P. Ruponen. Model test for the progressive flooding of a box-shaped barge, report m-292. Technical report, Helsinki University of Technology, 2006.

[79] P. Ruponen. Pressure-correction method for simulation of progressive flooding and internal air flows. *Ship Technology Research*, 53(2):63–73, 2006.

[80] P. Ruponen. *Progressive flooding of a damaged passenger ship*. PhD thesis, Helsinki University of Technology, Helsinki, 2007.

[81] P. Ruponen. Adaptive time step in simulation of progressive flooding. *Ocean Engineering*, 78:35–44, 2014.

[82] P. Ruponen. On the effects of non-watertight doors on progressive flooding in a damaged passenger ship. *Ocean Engineering*, 130:115–125, 2017.

[83] P. Ruponen, P. Kurvinen, I. Saisto, and J. Harras. Experimental and numerical study on progressive flooding in full-scale. *International Journal of Maritime Engineering*, 152(A4):197–208, 2010.

[84] P. Ruponen, M. Larmela, and P. Pennanen. Flooding prediction onboard a damage ship. In *Proceedings of the 11th International Conference on the Stability of Ships and Ocean Vehicles*, pages 391–400, Athens, Grece, 2012.

[85] P. Ruponen, D. Lindroth, and P. Pennanen. Prediction of survivability for decision support in ship flooding emergency. In *Proceedings of the 12th International Conference on the Stability of Ships and Ocean Vehicles*, Glasgow, UK, 2015.

[86] P. Ruponen, D. Lindroth, A.L. Routi, and M. Aartovaara. Simulation-based analysis method for damage survivability of passenger ships. *Ship Technology Research*, 66(3):180–192, 2019.

[87] P. Ruponen, P. Pennanen, and T. Manderbacka. On the alternative approaches to stability analysisin decision support for damaged passenger ships. *WMU Journal of Maritime Affairs*, 18:477–494, 2019.

[88] P. Ruponen, P. Penttilä, and M. Larmela. Estimation of damage and flooding extent from the flood sensor data. Technical Report D3.1, FLOODSTAND project, 2011.

[89] P. Ruponen, A. Pulkkinen, and J. Laaksonen. A method for breach assessment onboard a damaged passenger ship. *Applied Ocean Research*, 64:236–248, 2017.

[90] P. Ruponen and A.L. Routi. Guidelines and criteria on leakage occurrence modelling. Technical Report D2.2b, FLOODSTAND project, 2011.

[91] P. Ruponen, T. Sundell, and M. Larmela. Validation of a simulation method for progressive flooding. *International Shipbuilding Progress*, 54(4):305 – 321, 2007.

[92] E. Ruth, O. Olufse, and O. Rognebakke. CFD in damage stability. In *Proceedings of the 17th International Ship Stability Workshop*, pages 256–263, Helsinki, Finland, 2019.

[93] T.A. Santos and C. Guedes Soares. Study of damaged ship motions taking into account floodwater dynamics. *Journal of Marine Science and Technology*, 13(3):291 – 307, 2008.

[94] T.A. Santos and C. Guedes Soares. Numerical assessment of factors affecting the survivability of damaged ro–ro ships in waves. *Ocean Engineering*, 36(11):797 – 809, 2009.

[95] T.A. Santos, I.E. Winkle, and C. Guedes Soares. Time domain modelling of the transient asymmetric flooding of ro-ro ships. *Ocean Engineering*, 29(6):667–688, 2002.

[96] S. Schalck and J. Baatrup. Hydrostatic stability calculations by pressure integration. *Ocean Engineering*, 17(1):155–169, 1990.

[97] B. Scholkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond, Adaptive Computation and Machine Learning*. The MIT Press, Cambridge, USA, 2002.

[98] C. Seiffert, T. Khoshgoftaar, J. Hulse, and Napolitano A. Rusboost: Improving clasification performance when training data is skewed. In *Proceedings of the 19th International Conference on Pattern Recognition*, pages 1–4, Tampa, USA, 2008.

[99] P. Sen and C. Konstantinidis. A time simulation approach to the assessment of damage survivability of ro/ro cargo ships. *Transactions of Society of Naval Architects and Marine Engineers*, 95:337 – 355, 1987.

[100] J.R. Spouge. Technical investigation of the sinking of the ro-ro ferry european gateway. *Transactions RINA*, 128:49–72, 1986.

[101] F. Stefanidis, E. Boulougouris, and D. Vassalos. Ship evacuation and emergency response trends. In *Design & Operation of Passenger Ships 2019*, London, UK, 2019.

[102] G. Trincas, L. Braidotti, and L. De Francesco. Risk-based system to control safety level of flooded passenger ship. *Brodogradnja*, 68(1):31–60, 2017.

[103] F. van Walree and A. Papanikolaou. Benchmark study of numerical codes for the prediction of time to flood of ships: phase i. In *Proceedings of the 9th International workshop on Ship Stability*, Hamburg, 2007.

[104] R. van't Veer. Time to flood (ttf) simulations for a large passenger ship – final study. Technical Report 19289-1-CPS, MARIN, 2004.

[105] R. van't Veer and J.O. de Kat. Experimental and numerical investigation on progressive flooding and in complex compartment geometries. In *Proceedings of the 7th International Conference on Stability of Ships and Ocean Vehicles*, Launceston, Australia, 2000.

[106] R. van't Veer, W. Peters, A.L. Rimpela, and J. de Kat. Exploring the influence of different arrangements of semi-watertight spaces on survivability of a damaged large passenger ship. In *Proceedings of 7th International Ship Stability Workshop*, pages 30–41, Shangha, China, 2004.

[107] V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, USA, 1995.

[108] V.N. Vapnik and A.Y. Lerner. Pattern recognition using generazlied portraits method. *Automation and Remote Control*, 24:774–780, 1963.

[109] J.M. Varela, J.M. Rodrigues, and C. Guedes Soares. On-board decision support system for ship flooding emergency response. *Procedia Computer Science*, 29:1688–1700, 2014. 2014 International Conference on Computational Science.

[110] P. Wilmott. *Machine learning, an applied mathematics introduction*. Panda Ohana Publishing, Middletown, USA, 2019.

[111] C.D. Wood, D.A. Hudson, M. Tan, A.J. Sobey, and Y. Wang. Experimental investigation into factors affecting the transient flow of fluid through an orifice in realistic conditions. *Transactions of the Royal Institution of Naval Architects. Part A: International Journal of Maritime Engineering*, 158(3):1–12, 2016.

[112] X.L. Zhang, Z. Lin, P. Li, Y. Dong, and F. Liu. Time domain simulation of damage flooding considering air compression characteristics. *Water*, 11(4):796, 2019.

List of Symbols

Symbol	Unit	Description
$\dot{}$	1/s	Time derivative notation; it means $\partial/\partial t$
A	m^2	Submerged cross section of an opening (considering all the corrections due to free outflow and closure status)
Acc	-	Accuracy of a classification task
Acc^*	-	Ongoing accuracy of a classification task evaluated considering only damage cases having $t_f \geq t^*$
A_{ds}	m^2	Deeply submerged fraction of the cross section of an opening
A_e	m^2	Effective area equivalent to the free outflow fraction of the cross section of an opening applying deeply submerged formulation
A_{fo}	m^2	Free outflow fraction of the cross section of an opening
$A_{o_{ji}}$	m^2	Open area of the opening's submerged fraction
B	m	Ship breadth
B_d	m	Damage penetration (transversal extension of damage from $\pm B/2$ in the ship-fixed reference system
cdf	-	Cumulative density function
C_D	-	Non-dimensional discharge coefficient of an opening
c_t	-	Non-dimensional correction factor for evaluation of non-dimensional time-to-flood in a two-rooms scenario
d	-	Distance between two predictors' vectors
D	m	Ship depth
$\mathbf{D}$	-	Eigenvalues diagonal matrix
$\mathrm{d}t$	s	Time step
$\mathrm{d}t^*$	s	Previous time step
$\mathrm{d}t_c$	s	Maximum time step associated to an opening
$\mathrm{d}t_{max}$	s	Maximum time step
e_h	-	Error ratio of h quantity
e_h^2	-	Squared error of h quantity
g	m/s^2	Gravitational acceleration (assumed value: 9.80665 m/s^2)
G	-	Kernel function

Symbol	Unit	Description
GM	m	Metacentric height
h	m	Opening height or a generic quantity to define errors
H_d	m	Height of the damage in the ship-fixed reference system
J	-	Non-dimensional damage length according to SOLAS
$\mathbf{J}$	1/s	Jacobian matrix
J_m	-	Maximum non-dimensional damage length according to SOLAS
J_k	-	Knuckle non-dimensional damage length according to SOLAS
K	$m^{5/2}/s$	Opening dimensional coefficient
$k_{\mathbf{d}t}$	-	Non-dimensional coefficient to control time step adjustment
k_L	-	Pressure loss coefficient of an opening
L_{BP}	m	Length between perpendiculars
LCD	m	Longitudinal position of the centre of damage in ship-fixed reference system
LCG	m	Longitudinal position of the centre of mass in ship-fixed reference system
L_d	m	Damage length (longitudinal extension of the damage in the ship fixed reference system)
L_{OA}	m	Length overall
L_S	m	Subdivision Length
MSE	-	Mean of squared errors
n	-	Number of flooded rooms at a defined time instant
n_d	-	Number of divisions adopted in parametric damage cases generation
N	-	Number of damage cases within a database
N^*	-	Number of ongoing damage cases at a specific instant t^*
N_c	-	Number of correctly classified damage cases at a specific instant t^*
N_c^*	-	Number of correctly classified ongoing damage cases at a specific instant t^*
$OXYZ$	(m,m,m)	Ship fixed reference system having origin at intersection of base line and after perpendicular
$O'xyz$	(m,m,m)	Earth fixed reference system having origin on the sea free surface, axes x and y lying on the waterplane and axis z orthogonal to the waterplane (positive upwards)
p_a	kPa	Air pressure
Q	m^3/s	Volume flowrate
$Q_{fo_{ji}}$	m^3/s	Volume flowrate for free outflow
r	m	Variable of decoupled differential equations system
R^2	-	Coefficient of determination
R^{2*}	-	Ongoing coefficient of determination of a regression task evaluated considering only damage cases having $t_f \geq t^*$
s	m	Ship sinkage, defined as the difference between the actual and initial mean draught

Symbol	Unit	Description
S	m^2	Free surface area in a flooded room
$\mathbf{S}$	$1/m^2$	Diagonal matrix of rooms' inverse free surface area
SSE	-	Sum of squared estimate of errors
SS_{tot}	-	Total sum of squares
T	m	Ship design draught
T^*	m	Ship initial mean draught at a defined time instant
TCD	m	Transversal position of the centre of damage in ship fixed reference system
TCG	m	Transversal position of the centre of mass in ship fixed reference system
T_M	m	Ship mean draught (at midship perpendicular)
T_{M_0}	m	Ship initial mean draught at damage occurrence
t	s	Time
t^*	s	Initial/specific time instant
t_a	-	Non-dimensional time-to-flood
t_c	s	Time instant at which a defined level z_c is reached
t_C	s	Computational time
t_f	s	Time-to-flood
u	m/s	Flow velocity
v	m	known term in the decoupled differential equations system
V	m^3	Floodwater volume in a room
$\mathbf{V}$	-	Eigenvectors matrix
V_a	m^3	Air volume in a room
VCD	m	Vertical position of the centre of damage in ship fixed reference system
VCG	m	Vertical position of the centre of mass in ship fixed reference system
V_e	m^3	Capacity of a room (considering permeability)
V_w	m^3	Actual floodwater volume in a room with air pocket
w	m	Opening width
X_C	m	Longitudinal position of the opening's geometric centre in ship fixed reference system
X_d	m	Longitudinal position of the damage centre in the ship-fixed reference system
Y_C	m	Transversal position of the opening's geometric centre in ship fixed reference system
$\mathbf{Z}$	m^2/s	Symmetric matrix for Jacobian decomposition
Z_C	m	Vertical position of the opening's geometric centre in ship fixed reference system
$\hat{z}$	m	Corrected waterlevel of a room: maximum of the room level and the lowest tip of an opening

Symbol	Unit	Description
z	m	Floodwater level in non-filled rooms or vertical coordinate of a point in earth-fixed reference system
z^*	m	Initial level in flooded rooms
z^{**}	m	Initial level in flooded rooms at previous time step
z'	m	Difference between current and initial floodwater level
z_b	m	Maximum among lowest floodwater level and the vertical position of the lowest tip of the opening in the earth-fixed reference system
z_c	m	Critical effective waterhead for door collapsing or a critical level, in general
Z_d	m	Vertical position of the damage centre in the ship-fixed reference system
z_e	m	Effective waterhead applied to an opening
z_l	m	Critical effective waterhead for door leakage
z_{max}	m	Vertical position of the highest tip of the opening in the earth-fixed reference system
z_{min}	m	Vertical position of the lowest tip of the opening in the earth-fixed reference system
z_{rb}	m	Vertical posiotion of the room bottom in the earth-fixed reference system
z_t	m	Minimum among highest floodwater level and the vertical position of the highest tip of the opening in the earth-fixed reference system
z_w	m	Actual level in non-filled rooms with air pockets
z_0	m	Atmospheric pressure equivalent waterhead (Atmospheric pressure assumed value: 101.325 kPa)
Z_{max}	m	Vertical position of the highest tip of the damage in the ship-fixed reference system (from Base Line)
Z_{min}	m	Vertical position of the lowest tip of the damage in the ship-fixed reference system (from Base Line)
α	-	Non-dimensional leakage coefficient
β	1/m	Dimensional leakage coefficient
γ	-	free outflow correction factor
Δ	t	Displacement
ζ	m	Waterhead in filled rooms
θ	deg	Ship angle of trim
μ	-	Permeability of a room
ρ	t/m^3	Water density (assumed value: 1.025 t/m^3 and 1.000 t/m^3 for sea and fresh water, respectively)
ϕ	deg	Ship angle of heel
∇	m^3	Displacing volume

List of Abbreviations

Abbreviation	Description
BL	Base Line
C	Capsize
3D	three-dimensional
CFD	Computational Fluid Dynamics
CPU	Central processing unit
DAE	Differential Algebraic Equation
DOF	Degrees of Freedom
DSS	Decision Support System
DT	Decision Trees
DTB	Boosted Decision Trees
E	Equilibrium
EH	Excessive Heeling
FC	Flooded Compartments
FF	Final Fate
IMO	International Maritime Organization
ITTC	International Towing Tank Conference
KNN	K-Nearest Neighbours
KNN2	Quadratic K-Nearest Neighbours
KNN3	Cubic K-Nearest Neighbours
KNNC	Cosine K-Nearest Neighbours
KNNW	Weighted K-Nearest Neighbours
LDAE	Linearised Differential Algebraic Equation
LODE	Linearised Ordinary Differential Equations
LODEG	Linearised Ordinary Differential Equations with Grouping
MC	Monte Carlo
ODE	Ordinary Differential Equations
PC	Pressure Correction
PEM	Primary Electric Motor
PFO	Progressive Flooding Opening
RF	Random Forests

Abbreviation	Description
S	Sink
SOLAS	International Convention for the Safety of Life at Sea
SRtP	Safe Return to Port
SVM	Support Vector Machine
SVM1	Linear Support Vector Machine
SVM2	Quadratic Support Vector Machine
SVM3	Cubic Support Vector Machine
SVMG	Gaussian Support Vector Machine
TE	Time Exceeded

List of Figures

List of Tables

Appendices

Appendix A

Sensitivity Study on Progressive Flooding Simulation

In this appendage, the effects of a systematic variation of parameters on progressive flooding simulations are studied for the Helsinki barge's test cases A, B, and C presented in Chapter 4. Part of this analysis has been already published in [7]. All simulations have been carried out applying the LODEG method with a constant time step 0.25 s. The parameters considered in the sensitivity study are related to damage geometry, ship geometry and initial condition. Different damage geometries have been tested varying the damage dimension and its location. The ship geometry has been altered in terms of permeabilities and discharge coefficients of the openings. Moreover, different initial conditions have been tested changing the height of the centre of mass KG and the displacement considering also the effect of the water density. For each parameter, the trends of heel, trim, sinkage, and all levels inside flooded compartments have been compared. Furthermore, the effect on several overall quantities characterizing the progressive flooding process has been also studied. In detail, these variables are the time-to-flood t_f, which is of the utmost importance for decision support purposes, the minimum value of metacentric height GM_{min} measured during the process, considered representative of the stability of the ship, the maximum heeling angle ϕ_{max} and the final floating position of the barge in terms of trim θ_e and sinkage s_e. The error e compared to original values has been evaluated for each variable in each case studied.

A.1 Damage geometry

The geometry of the damage has been studied in terms of two parameters: the damage area A_d and its location. The shape of the damage has not been considered since it is usually taken into account by means of a proper discharge coefficient. Concerning A_d, it has been increased and reduced by 5.0% and 10.0% with a constant position of the centre of damage. The location of damage was changed according to the specific case studied, assuming a constant damage area.

The damage centre (LCD, TCD, VCD) in case A was moved longitudinally and vertically along the barge side, while in cases B and C was moved longitudinally and transversally along the flat part of the bottom. The adopted positions are provided in Table A.1. The results of simulations in function of damage area are provided in Table A.2 and then in Figures A.1, A.2, and A.3 for cases A, B, and C respectively. The results of simulations in function of the damage location are provided in Table A.3 and in Figures A.4, A.5, and A.6 for cases A, B, and C respectively.

Table A.1 Applied damage position for cases A, B and C

case	location	LCD (m)	TCD (m)	VCD (m)
A	0	2.675	0.395	0.315
A	1	2.375	0.395	0.455
A	2	2.975	0.395	0.455
A	3	2.375	0.395	0.175
A	4	2.975	0.395	0.175
B	0	2.675	0.000	0.000
B	1	2.375	0.250	0.000
B	2	2.375	-0.250	0.000
B	3	2.975	0.250	0.000
B	4	2.975	-0.250	0.000
C	0	2.325	0.000	0.000
C	1	2.225	0.250	0.000
C	2	2.225	-0.250	0.000
C	3	2.425	0.250	0.000
C	4	2.425	-0.250	0.000

The geometry of damage does not affect the final floating position of the ship. Only some small differences have been experienced in case C but are caused by the delay in detection of an air pocket inside room DB1. Concerning the position, the only relevant parameter, which leads to significant differences, is the vertical position of damage while longitudinal and transversal translations do not result in notable differences. In case A, the vertical position has a small impact on t_f which increases with the VCD and a greater effect on ϕ_{max} which decreases as the VCD increases. Both the trends appear non-linear. The effects on the progressive flooding process induced by the A_d differ substantially due to the internal subdivision. The t_f increases with the damage area but the increment is also dependent on the area of the other openings of the first flooded room: the effect becomes even negligible in case C where the opening DB1-DB2 is very small compared with the damage size. Thus, greater damage size accelerates only the flooding of room DB1. In case A, the heeling moment during the transient phase increases with damage size, leading to a greater ϕ_{max} value.

Table A.2 Overall effect of damage size for cases A, B and C

case	ΔA_d	e_{t_f}	$e_{GM_{min}}$	$e_{\phi_{max}}$	e_{s_e}	e_{θ_e}
A	-10.00%	0.61%	0.11%	-7.30%	0.00%	0.00%
A	-5.00%	0.53%	0.10%	-2.86%	0.00%	0.00%
A	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
A	5.00%	-0.38%	0.01%	3.18%	0.00%	0.00%
A	10.00%	-0.83%	0.02%	12.30%	0.00%	0.00%
B	-10.00%	10.63%	0.00%	0.00%	0.03%	0.02%
B	5.00%	4.85%	0.00%	0.00%	0.02%	0.01%
B	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
B	5.00%	-4.54%	0.00%	0.00%	-0.01%	0.00%
B	10.00%	-7.96%	0.00%	0.00%	-0.02%	-0.02%
C	-10.00%	0.03%	0.02%	0.00%	-0.22%	-0.15%
C	-5.00%	-0.29%	0.02%	0.00%	-0.37%	-0.25%
C	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
C	5.00%	0.00%	0.02%	0.00%	-0.23%	-0.16%
C	10.00%	0.00%	-0.01%	0.00%	0.11%	0.07%

Table A.3 Overall effect of damage location for cases A, B and C

case	ΔA_d	e_{t_f}	$e_{GM_{min}}$	$e_{\phi_{max}}$	e_{s_e}	e_{θ_e}
A	1	3.10%	0.07%	-42.05%	0.00%	0.00%
A	2	2.80%	0.09%	-42.37%	0.00%	0.00%
A	0	0.00%	0.00%	0.00%	0.00%	0.00%
A	3	-0.15%	-0.03%	2.52%	0.00%	0.00%
A	4	-0.15%	-0.03%	2.52%	0.00%	0.00%
B	1	0.00%	0.00%	0.00%	0.00%	0.00%
B	2	0.00%	0.00%	0.00%	0.00%	0.00%
B	0	0.00%	0.00%	0.00%	0.00%	0.00%
B	3	0.00%	0.00%	0.00%	0.00%	0.00%
B	4	0.00%	0.00%	0.00%	0.00%	0.00%
C	1	0.00%	0.00%	0.00%	0.00%	0.00%
C	2	0.00%	0.00%	0.00%	0.00%	0.00%
C	0	0.00%	0.00%	0.00%	0.00%	0.00%
C	3	0.00%	0.00%	0.00%	0.00%	0.00%
C	4	0.00%	0.00%	0.00%	0.00%	0.00%

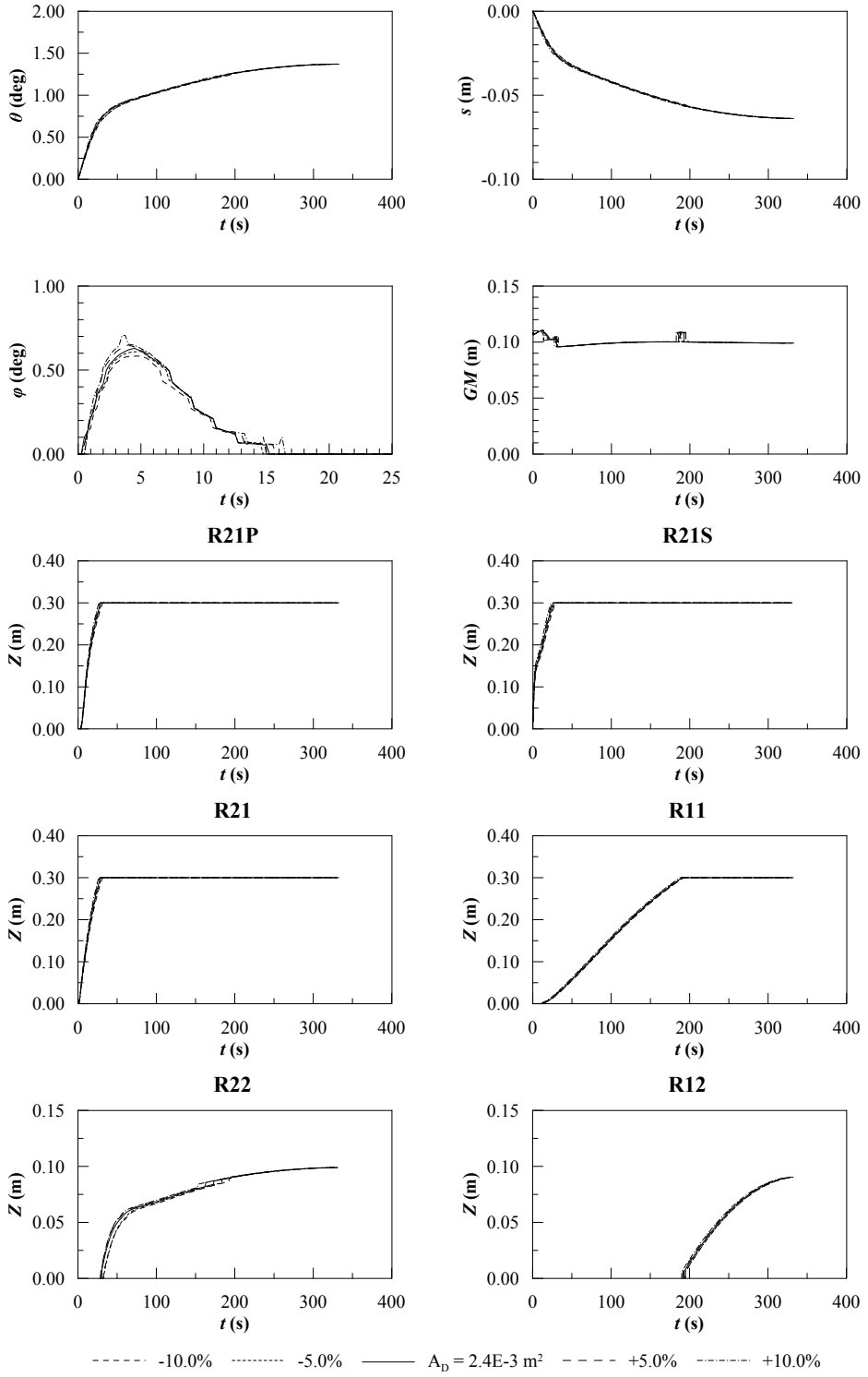

Figure A.1 Overall effect of damage size for cases A

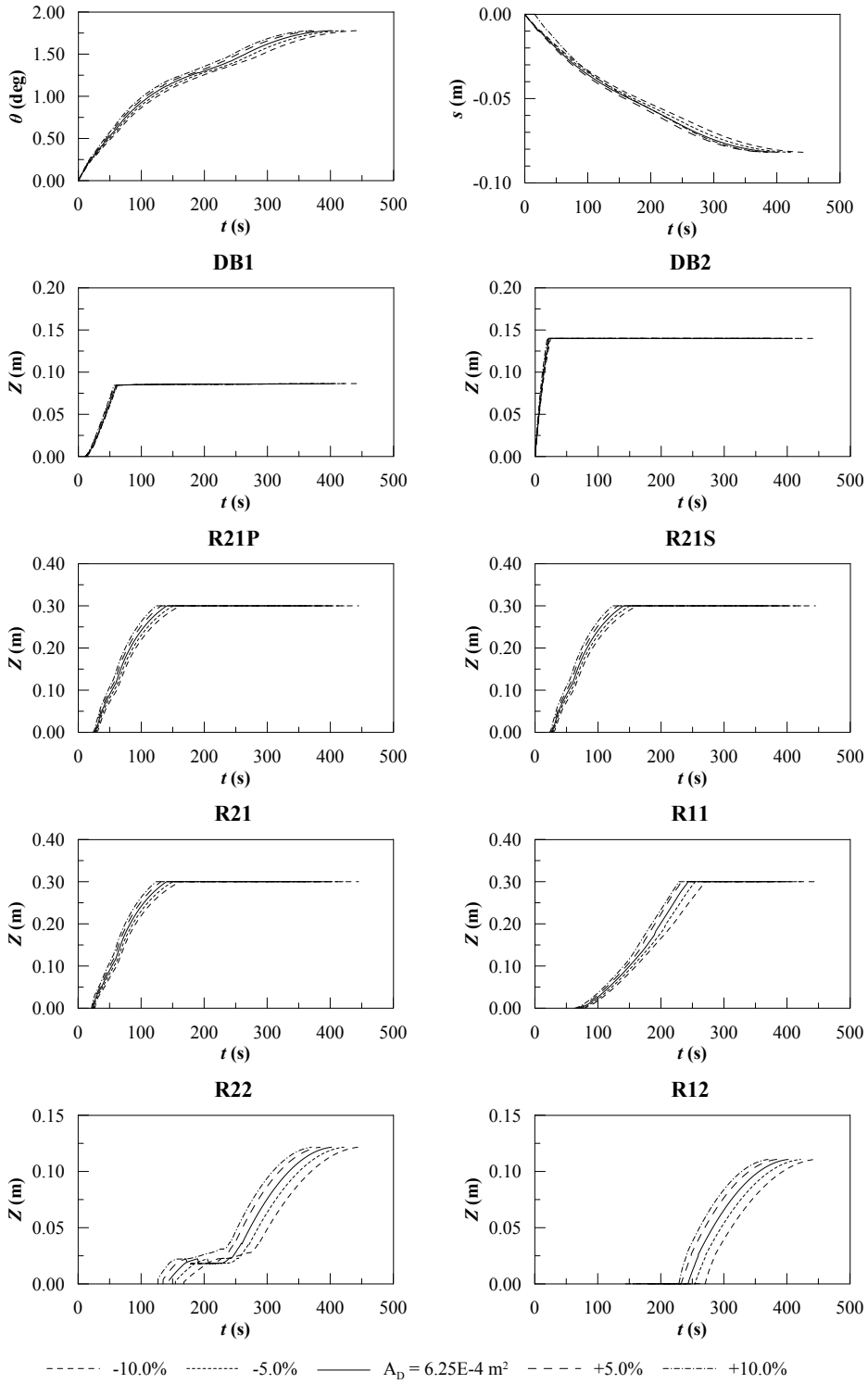

Figure A.2 Overall effect of damage size for cases B

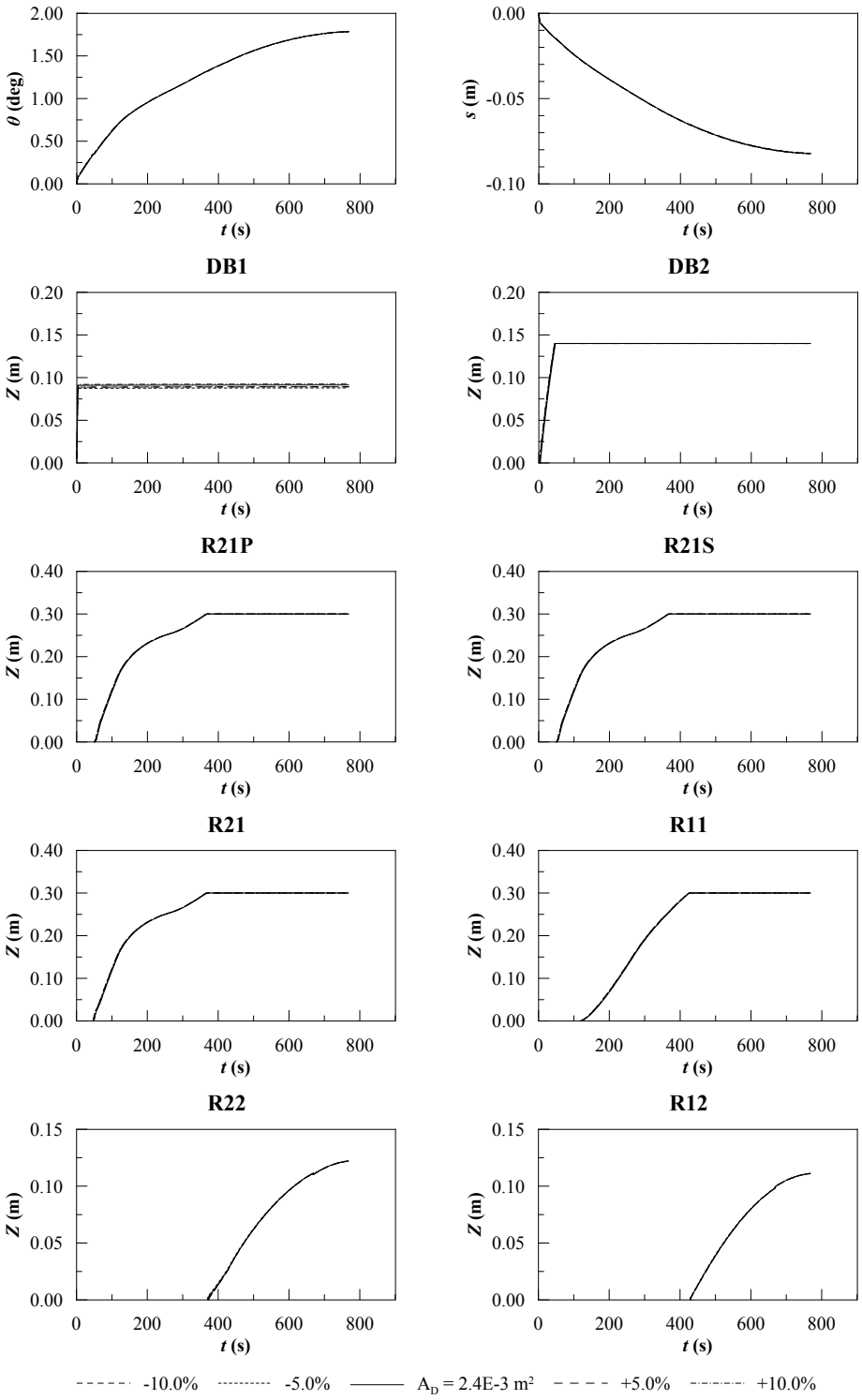

Figure A.3 Overall effect of damage size for cases C

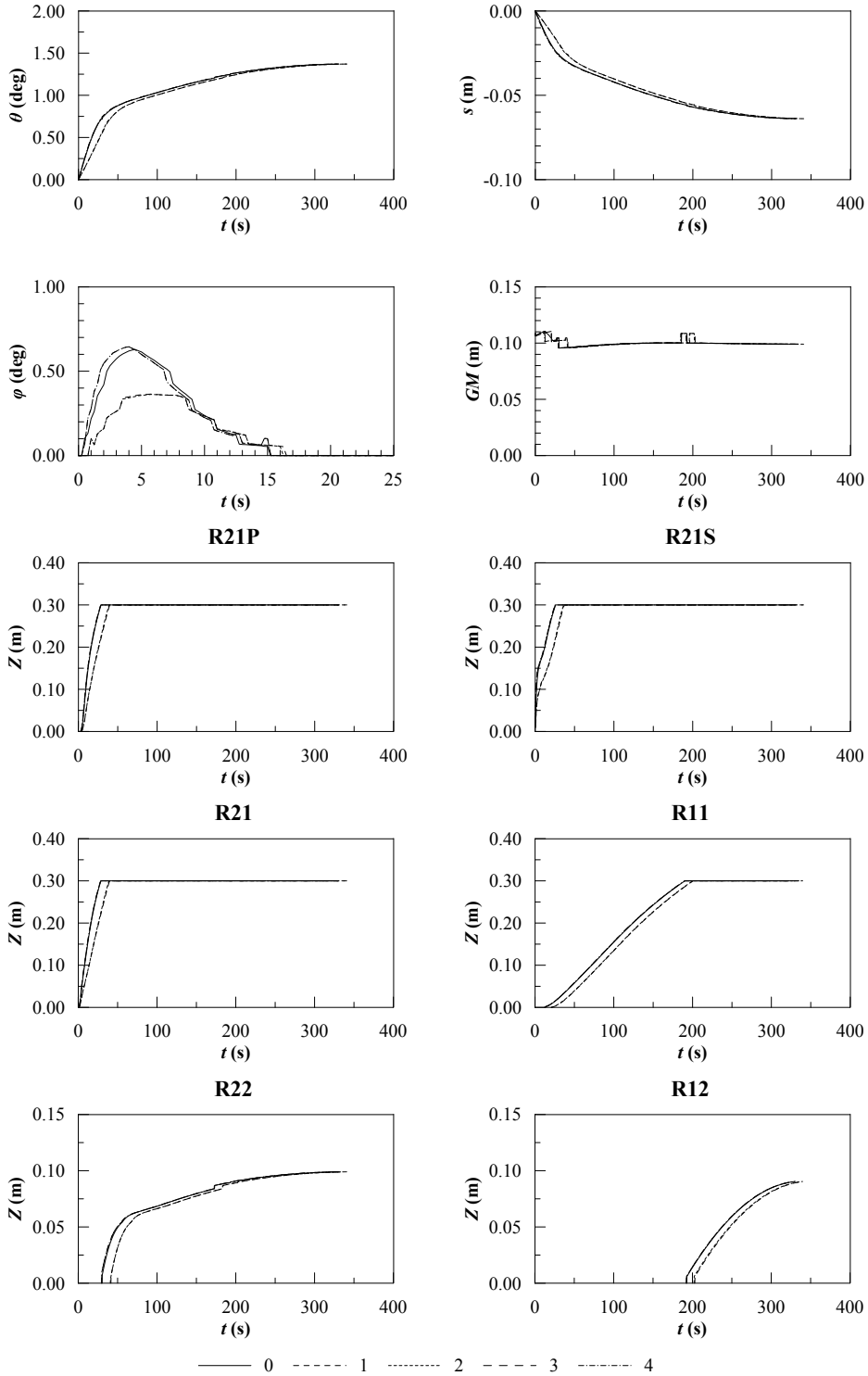

Figure A.4 Overall effect of damage location for cases A

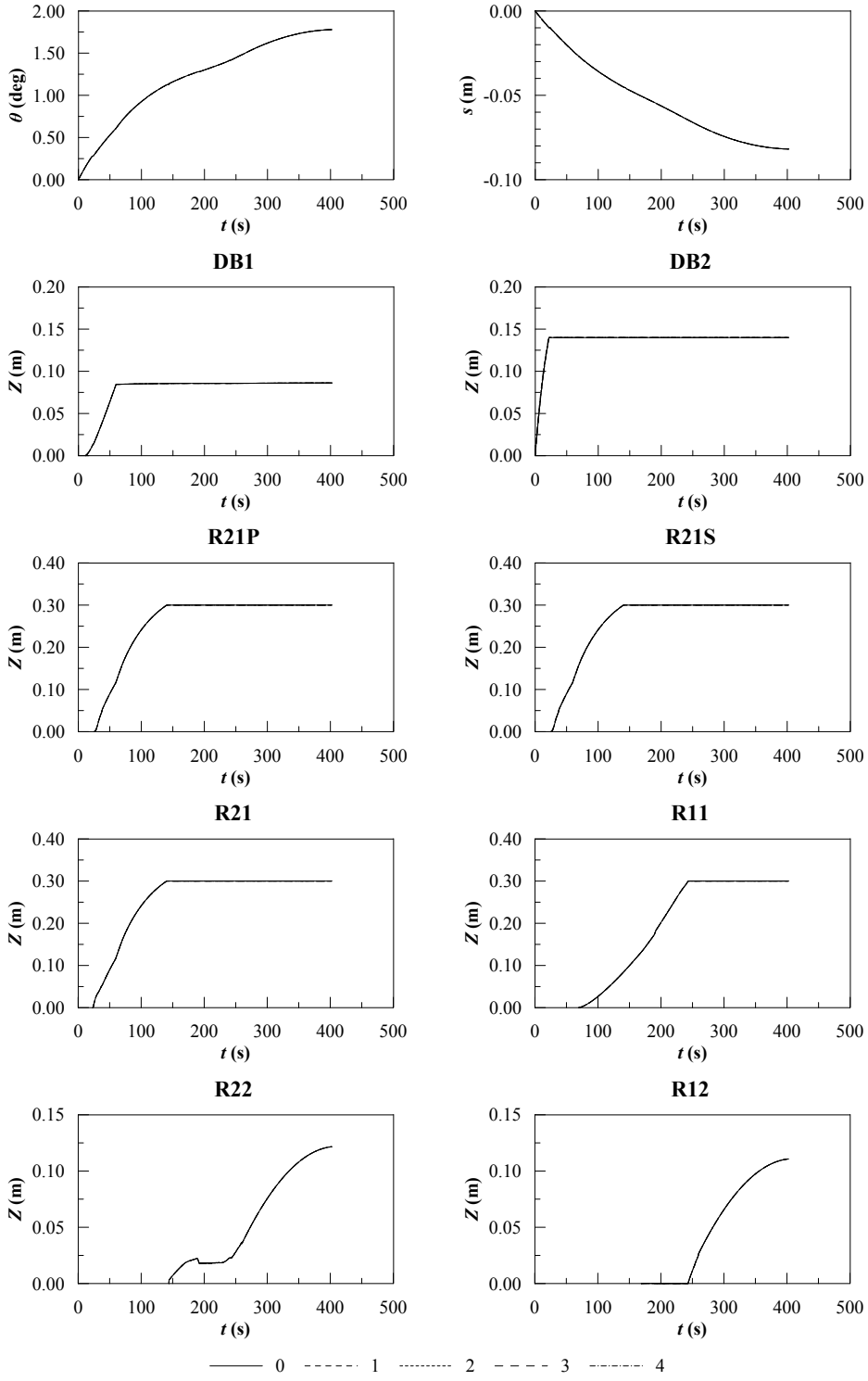

Figure A.5 Overall effect of damage location for cases B

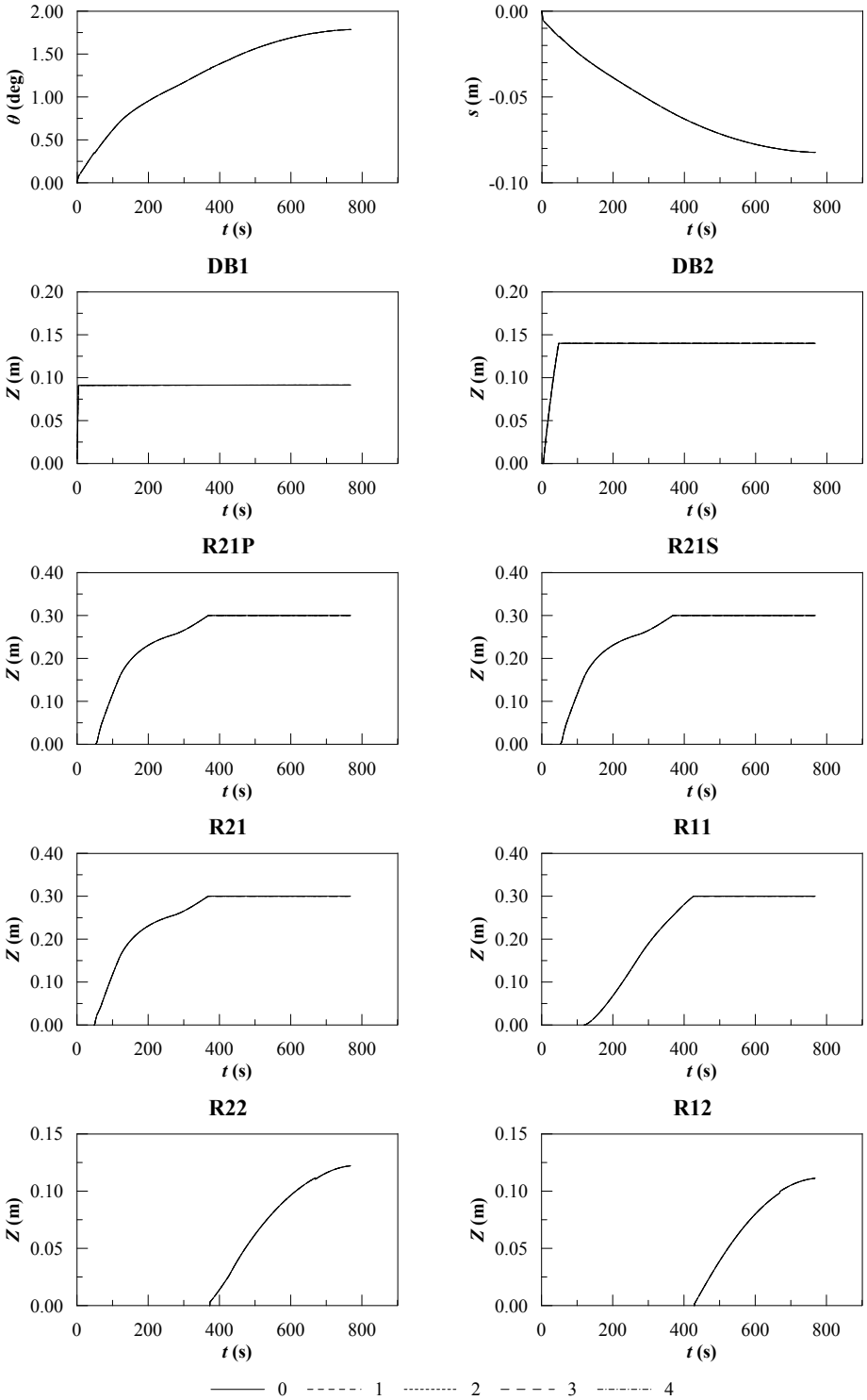

Figure A.6 Overall effect of damage location for cases C

A.2 Ship geometry

The internal layout of bulkheads and decks was considered constant, while different permeabilities and the discharge coefficients have been applied to all the compartments and all the openings respectively. The compartments of the test barge were modelled taking into account the thickness of all boundaries. Therefore, a unit permeability was applied during validation. Later, simulations were carried out applying lower permeabilities to all rooms, ranging between 1.00 and 0.85. The results are provided in Table A.4 and in Figures A.8, A.9 and A.10 for cases A, B and C respectively. The discharge coefficients for barge internal openings were assessed experimentally. Several works highlighted that in real scenarios the value of these coefficients is uncertain and usually change within the range 0.6 and 0.8 [111]. Nevertheless, it is common

Table A.4 Overall effect of permeabilities for cases A, B and C

case	ΔA_d	e_{t_f}	$e_{GM_{min}}$	$e_{\phi_{max}}$	e_{s_e}	e_{θ_e}
A	1.00	0.00%	0.00%	0.00%	0.00%	0.00%
A	0.95	-0.91%	-0.21%	-2.85%	-4.45%	-4.43%
A	0.90	-1.59%	-0.53%	-7.94%	-10.56%	-10.51%
A	0.85	-1.89%	-0.92%	-13.08%	-16.53%	-16.45%
B	1.00	0.00%	0.00%	0.00%	0.00%	0.00%
B	0.95	-0.44%	-0.15%	0.00%	-4.43%	-4.37%
B	0.90	-1.12%	-0.35%	0.00%	-10.57%	-10.43%
B	0.85	-1.62%	0.83%	0.00%	-16.52%	-16.31%
C	1.00	0.00%	0.00%	0.00%	0.00%	0.00%
C	0.95	-0.65%	-0.15%	0.00%	-4.46%	-4.40%
C	0.90	-1.86%	-0.37%	0.00%	-10.57%	-10.43%
C	0.85	-2.87%	-0.58%	0.00%	-16.54%	-16.33%

Table A.5 Overall effect of discharge coefficients for cases A, B and C

case	ΔA_d	e_{t_f}	$e_{GM_{min}}$	$e_{\phi_{max}}$	e_{s_e}	e_{θ_e}
A	-5.00%	5.52%	0.03%	-1.89%	0.00%	0.00%
A	-2.50%	2.65%	0.08%	-0.75%	0.00%	0.00%
A	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
A	2.50%	-4.01%	0.42%	0.39%	0.00%	0.00%
A	5.00%	-5.37%	0.10%	0.03%	0.00%	0.00%
B	-5.00%	4.97%	0.10%	0.00%	-0.01%	0.00%
B	-2.50%	2.49%	0.00%	0.00%	0.02%	0.01%
B	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
B	2.50%	-2.73%	0.00%	0.00%	0.00%	0.00%
B	5.00%	-4.85%	0.16%	0.00%	0.00%	0.00%
A	-5.00%	4.56%	3.41%	0.00%	-0.37%	-0.25%
A	-2.50%	1.95%	3.49%	0.00%	-0.18%	-0.13%

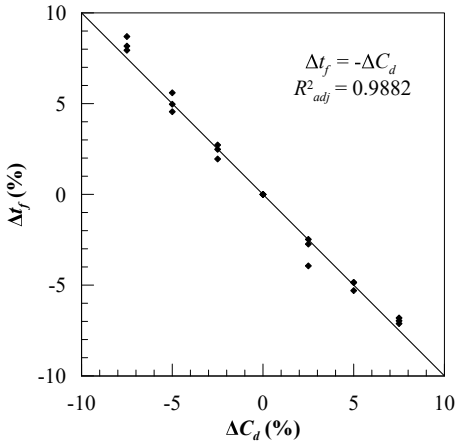

Figure A.7 Overall effect of permeabilities for cases A

practice to assume for full-scale simulations a value equal to 0.6 [83]. For such a reason, the effect of discharge coefficients on the progressive flooding process has been studied. To this end, all the coefficients associated with barge openings (including damages) have been multiplied by a coefficient reducing and increasing the experimental values by 2.5% and 5.0%. The results are provided in Table A.5 and in Figures A.11, A.12 and A.13 for cases A, B and C respectively.

The permeability has a heavy impact on the ship floating position during the whole progressive flooding process, since it acts on the volume of floodwater loaded onboard. For the simple geometry of the test barge, the percentage reduction of trim, heel and sinkage is almost equal to the one connected to permeability. It is worth noticing that the effect on $\overline{GM}_{min}$ is not heavy as well as the one on t_f. In fact, although a lower volume has to be filled, the resulting lower draught at openings drives to lower velocities that approximately compensate the time-to-flood reduction (a reduction of 2% of t_f is caused by a reduction of 15% of μ).

The discharge coefficient has no effects on progressive flooding except for the time evolution. As a matter of fact, the floating position, the stability and the levels of floodwater assume the same values, but a delay or an advance is observed proportional to the reduction or increase of discharge coefficients values. In detail, the percentage error on time-to-flood Δt_f is almost equal and opposite to the percentage difference in discharge coefficients ΔC_d applied to all the openings. Figure A.7 shows that this trend is confirmed from a ΔC_d ranging from -7.5% up to 7.5%, which is representative of a realistic uncertainty level for discharge coefficients.

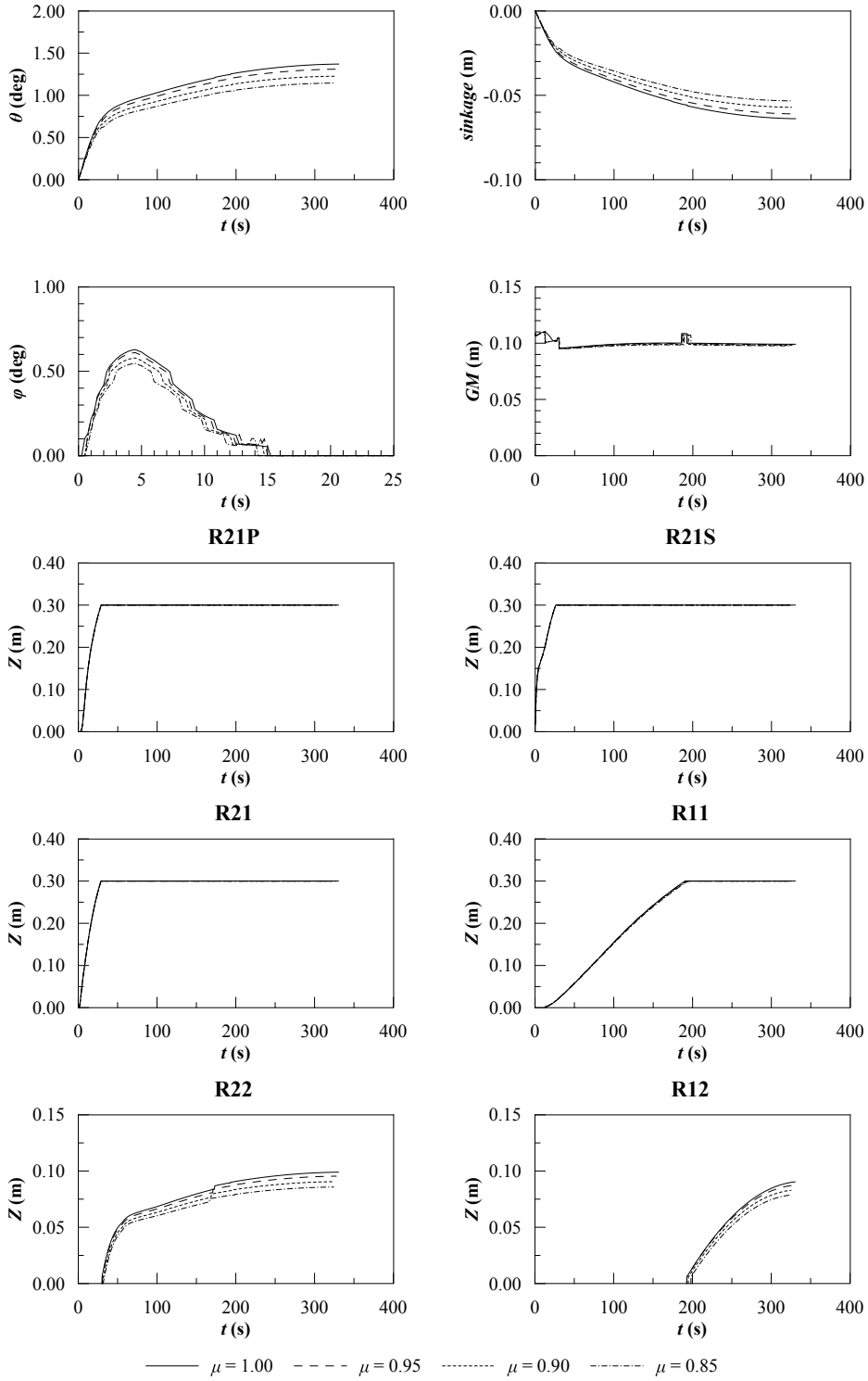

Figure A.8 Overall effect of permeabilities for cases A

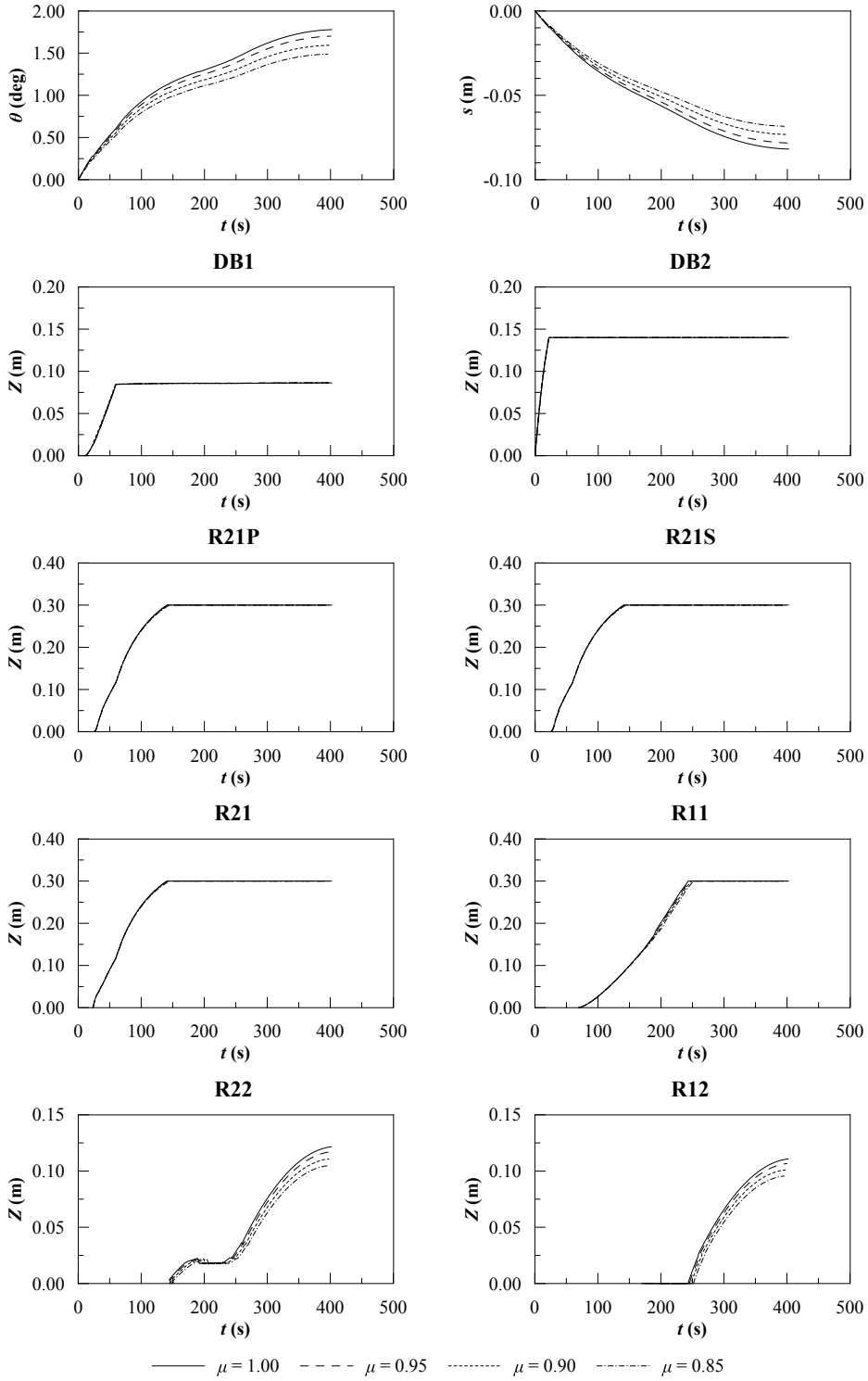

Figure A.9 Overall effect of permeabilities for cases B

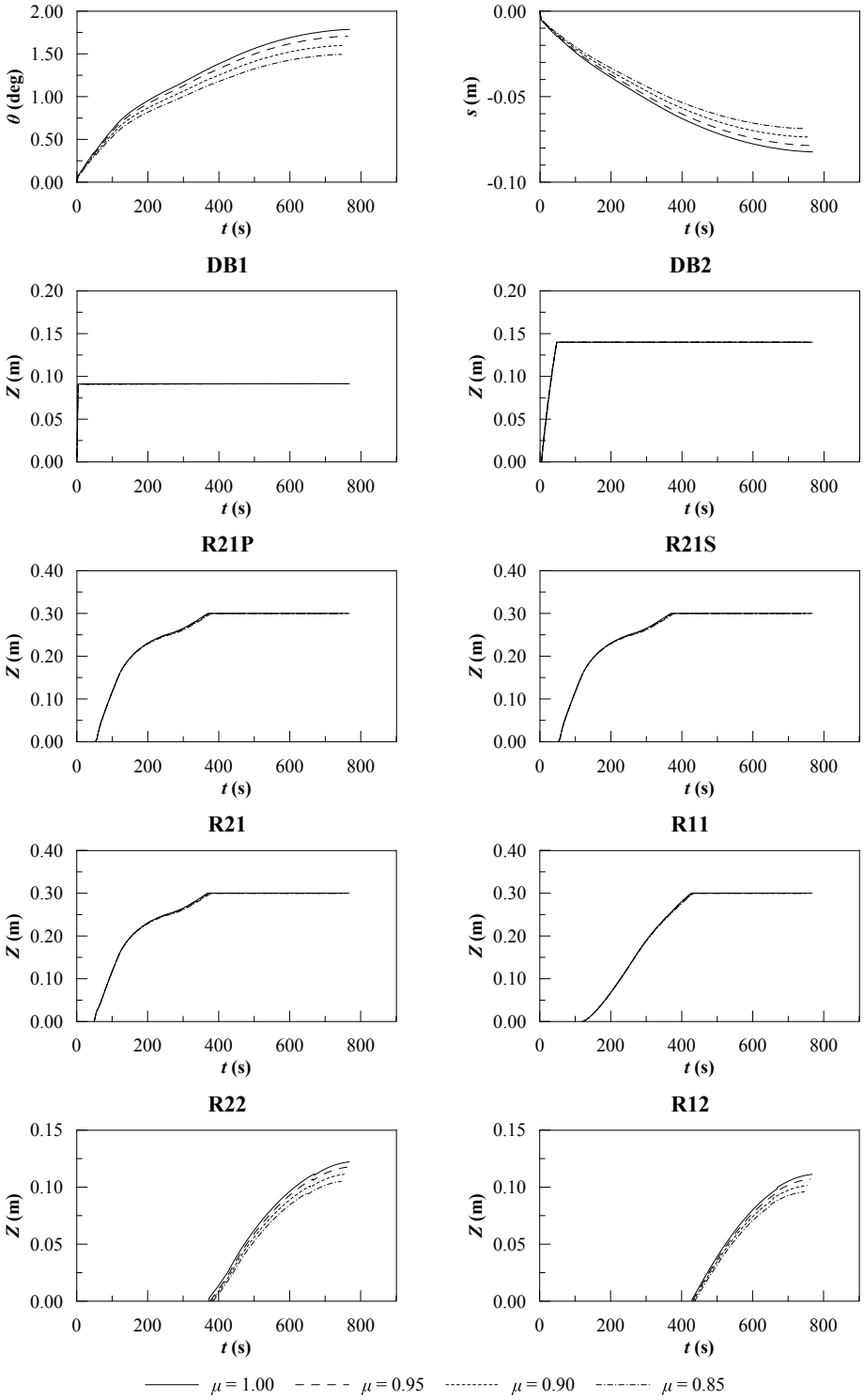

Figure A.10 Overall effect of permeabilities for cases C

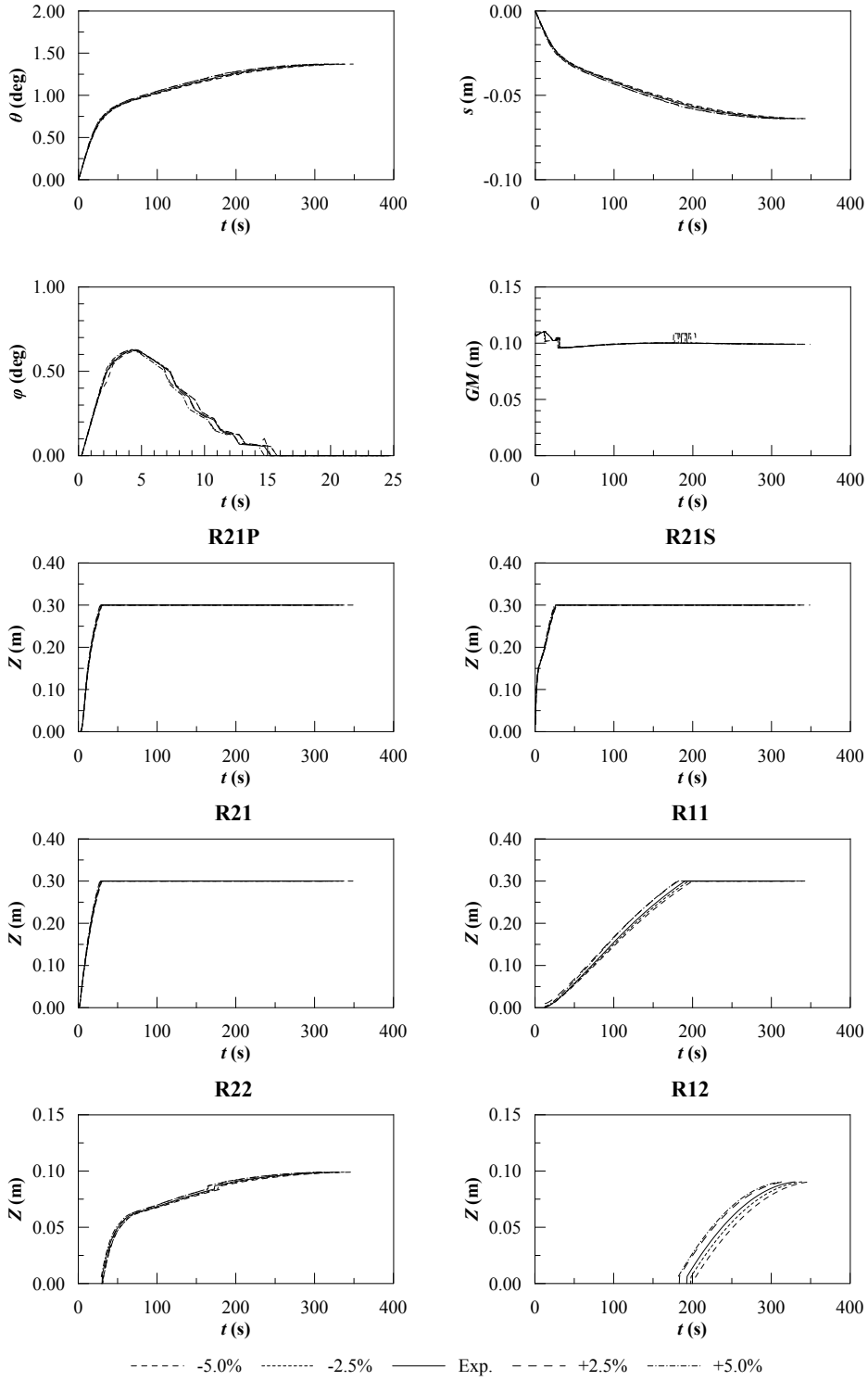

Figure A.11 Overall effect of discharge coefficients for cases A

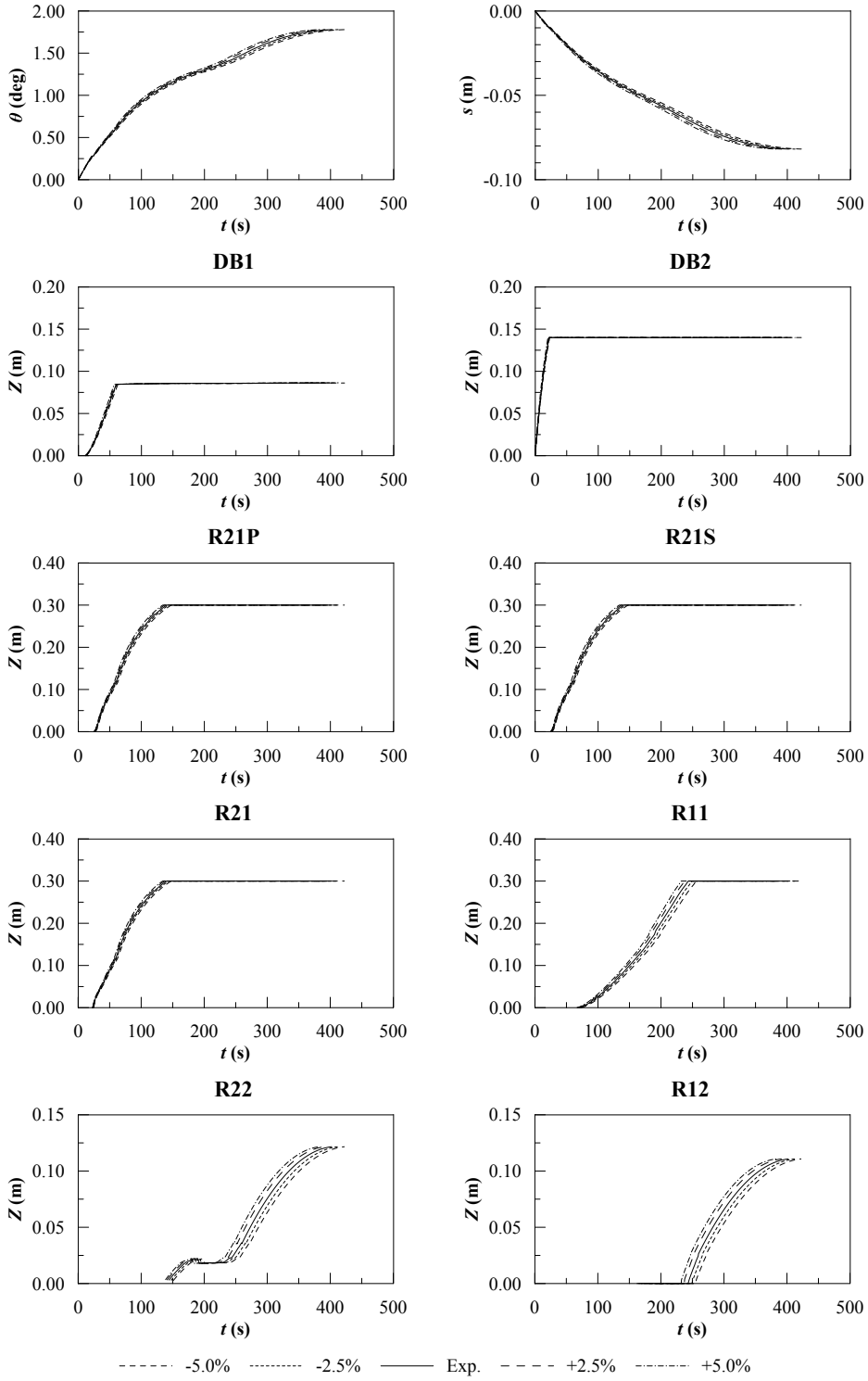

Figure A.12 Overall effect of discharge coefficients for cases B

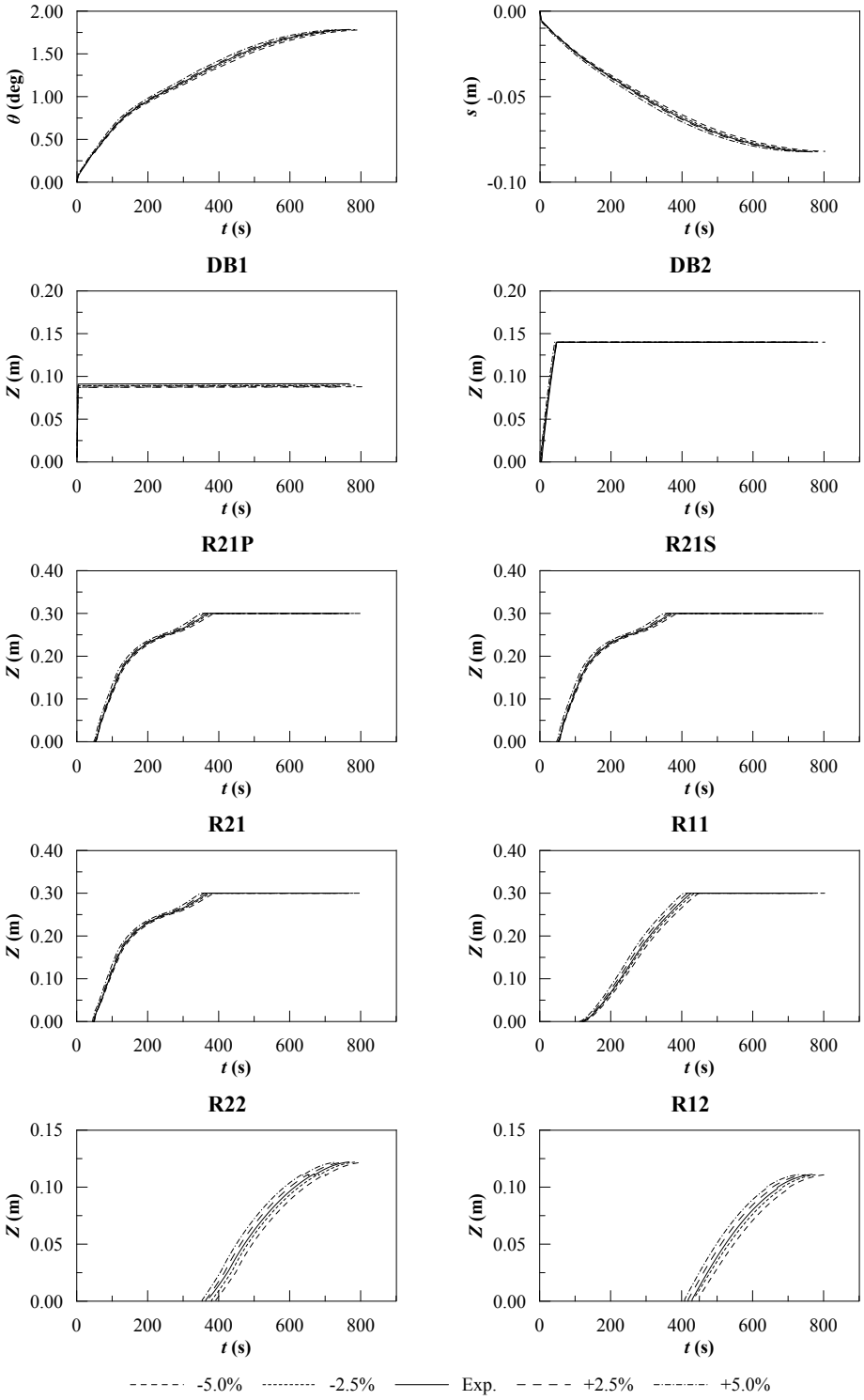

Figure A.13 Overall effect of discharge coefficients for cases C

A.3 Initial Condition

In an operative condition, often the actual displacement, the position of mass centre and the water density are not accurately defined. An uncertainty within 5% on ship weight assessed via loading instrument compared with the displacement corresponding to the measured draughts is considered acceptable. This is why the impact of the initial condition on the progressive flooding process has been studied.

To this end, different values of displacement Δ and height of the centre of mass VCG have been studied. The water density ρ was not analysed, since it does not influence the progressive flooding. In fact, from several simulations carried out with constant initial draughts and variable water density no differences were spotted on floating position, water levels in compartments, nor ship stability during the whole progressive flooding process. Concerning the displacement, it has been increased and reduced by a 2.5% and 5.0% and the results are provided in Table A.6 and in Figures A.14, A.15 and A.16 for cases A, B and C, respectively. The height of mass centre has been increased and reduced by a 5.0% and 10.0% and the results are provided in Table A.7 and in Figures A.17, A.18 and A.19 for cases A, B and C respectively.

It has been proved that the initial condition has a heavy impact on progressive flooding. In particular, an increased displacement causes an increment of time-to-flood, of initial stability, on final draught and trim, whereas a reduction of healing angle during the transient phase. The vertical position of the mass centre has a relevant effect only on ship stability and, thus, also on the heeling angle during the transient phase. Only a modest effect on the trim was observed but was not sufficient to cause a relevant effect on the water level inside compartments. Nevertheless, the barge has a large value of initial $\overline{GM}$; in case of asymmetric flooding scenario or for less stable ships (e.g. large passenger ships), VCG is likely to have even a greater impact. Moreover, a low value of VCG could lead to a reduction of time-to-flood in a capsize scenario.

Table A.6 Overall effect of displacement for cases A, B and C

case	ΔA_d	e_{t_f}	$e_{GM_{min}}$	$e_{\phi_{max}}$	e_{s_e}	e_{θ_e}
A	1.378	-2.80%	-6.54%	15.55%	-6.98%	-6.60%
A	1.414	-1.59%	-3.35%	4.16%	-3.49%	-3.29%
A	1.450	0.00%	0.00%	0.00%	0.00%	0.00%
A	1.486	1.13%	3.56%	-3.09%	3.49%	3.25%
A	1.523	3.10%	7.15%	-9.04%	6.99%	6.47%
B	1.378	0.37%	-5.49%	0.00%	0.02%	0.01%
B	1.414	0.75%	-2.71%	0.00%	-2.72%	-2.44%
B	1.450	0.00%	0.00%	0.00%	0.00%	0.00%
B	1.486	-0.12%	2.95%	0.00%	2.73%	2.42%
B	1.523	0.75%	6.02%	0.00%	5.48%	4.85%
C	1.378	-3.09%	-5.51%	0.00%	-5.62%	-5.04%
C	1.414	-1.66%	1.38%	0.00%	-2.81%	-2.50%
C	1.450	0.00%	0.00%	0.00%	0.00%	0.00%
C	1.486	1.37%	2.95%	0.00%	2.81%	2.48%
C	1.523	3.06%	6.03%	0.00%	5.02%	4.54%

Table A.7 Overall effect of mass centre height for cases A, B and C

case	ΔA_d	e_{t_f}	$e_{GM_{min}}$	$e_{\phi_{max}}$	e_{s_e}	e_{θ_e}
A	0.264	0.08%	27.49%	-20.46%	-0.03%	-0.94%
A	0.250	0.00%	13.78%	-11.27%	-0.01%	-0.47%
A	0.278	0.00%	0.00%	0.00%	0.00%	0.00%
A	0.292	0.00%	-13.66%	23.87%	0.02%	0.48%
A	0.306	-0.61%	-27.20%	46.07%	0.04%	0.97%
B	0.264	0.00%	27.53%	0.00%	-0.06%	-0.96%
B	0.250	0.00%	13.77%	0.00%	-0.02%	-0.47%
B	0.278	0.00%	0.00%	0.00%	0.00%	0.00%
B	0.292	0.00%	-13.77%	0.00%	-0.01%	0.46%
B	0.306	0.81%	-27.53%	0.00%	0.04%	0.96%
C	0.264	-0.20%	27.49%	0.00%	-0.04%	-0.94%
C	0.250	-0.20%	16.60%	0.00%	-0.02%	-0.47%
C	0.278	0.00%	0.00%	0.00%	0.00%	0.00%
C	0.292	0.52%	-13.79%	0.00%	0.02%	0.48%
C	0.306	0.20%	-27.55%	0.00%	0.04%	0.96%

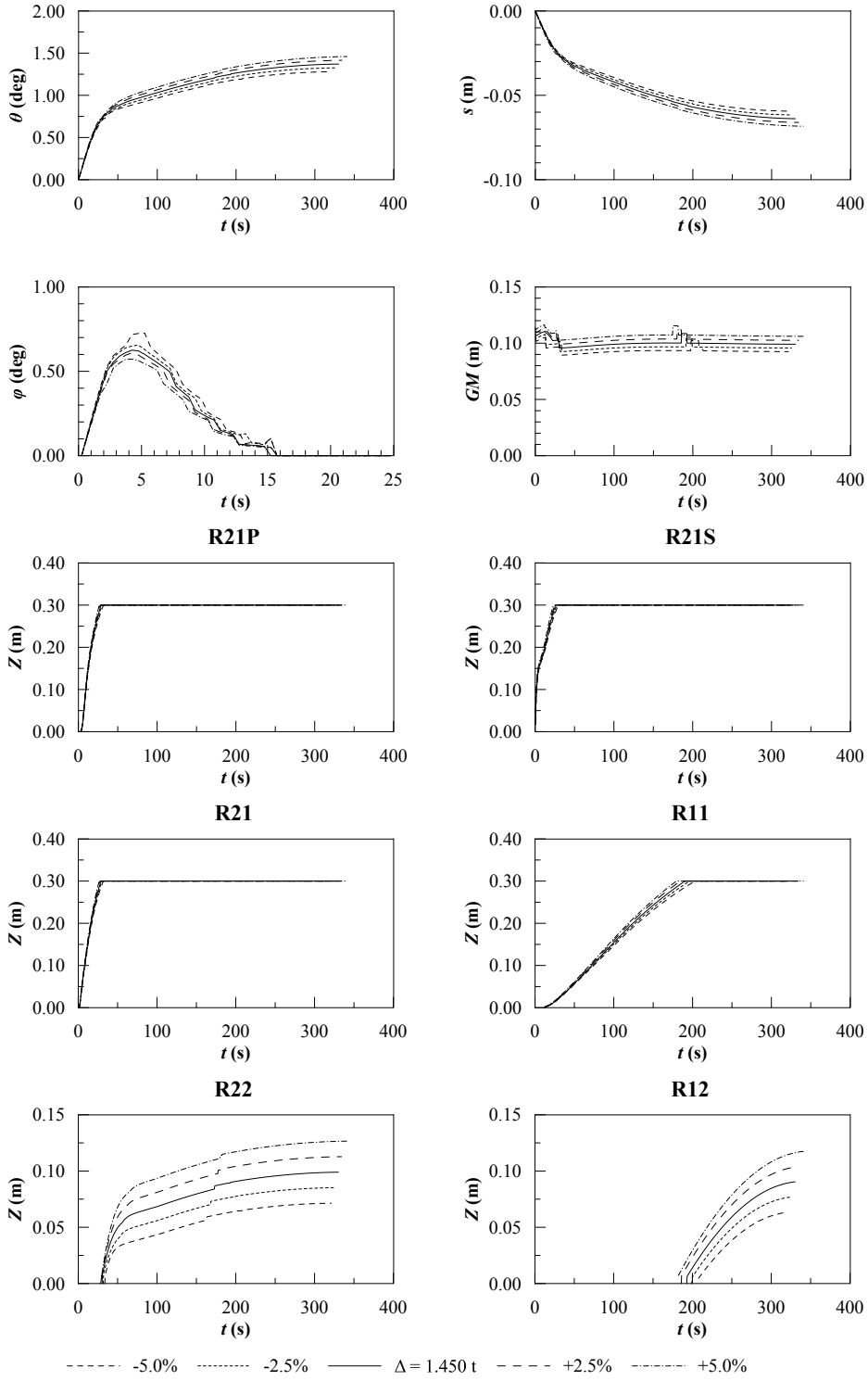

Figure A.14 Overall effect of displacement for cases A

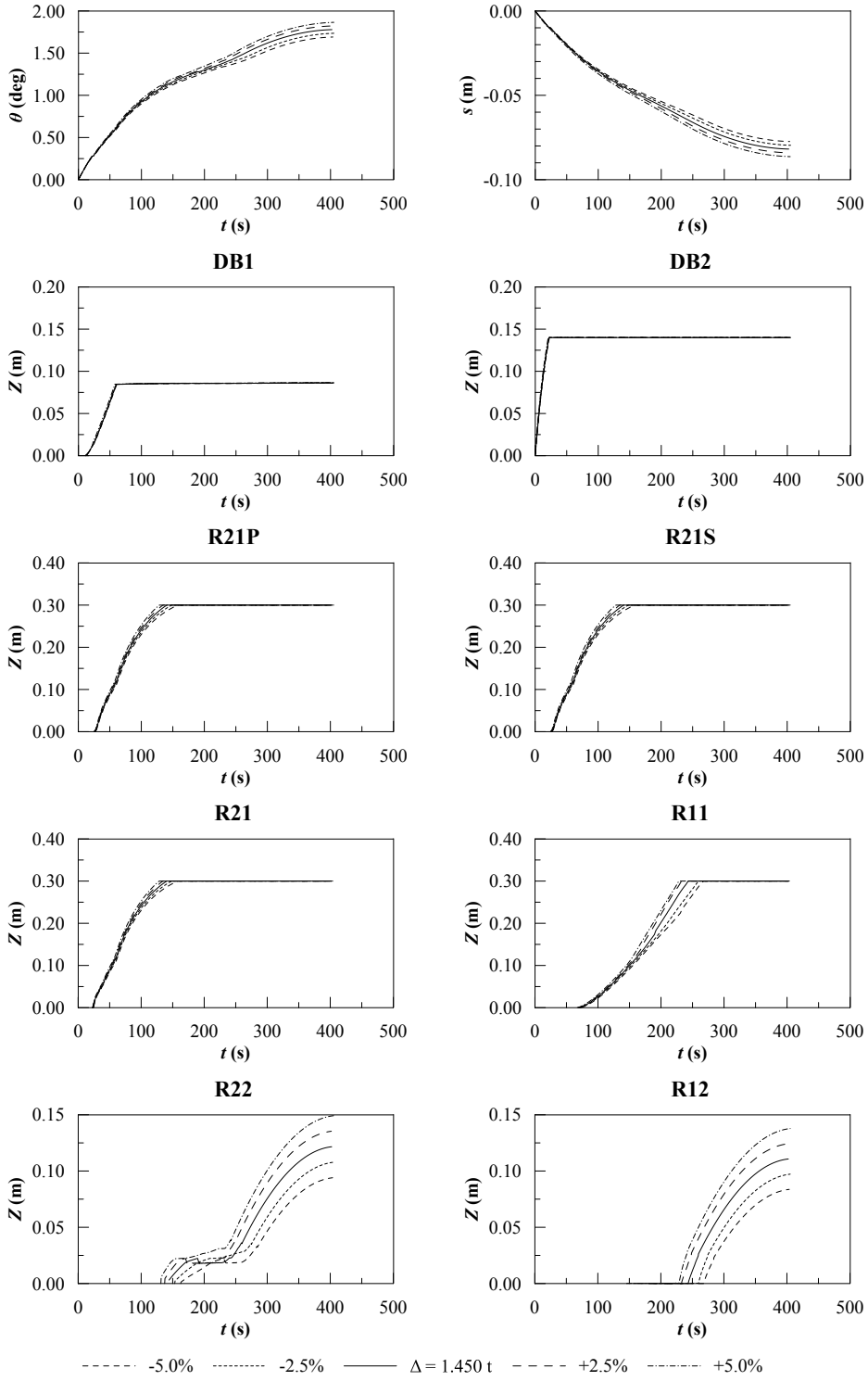

Figure A.15 Overall effect of displacement for cases B

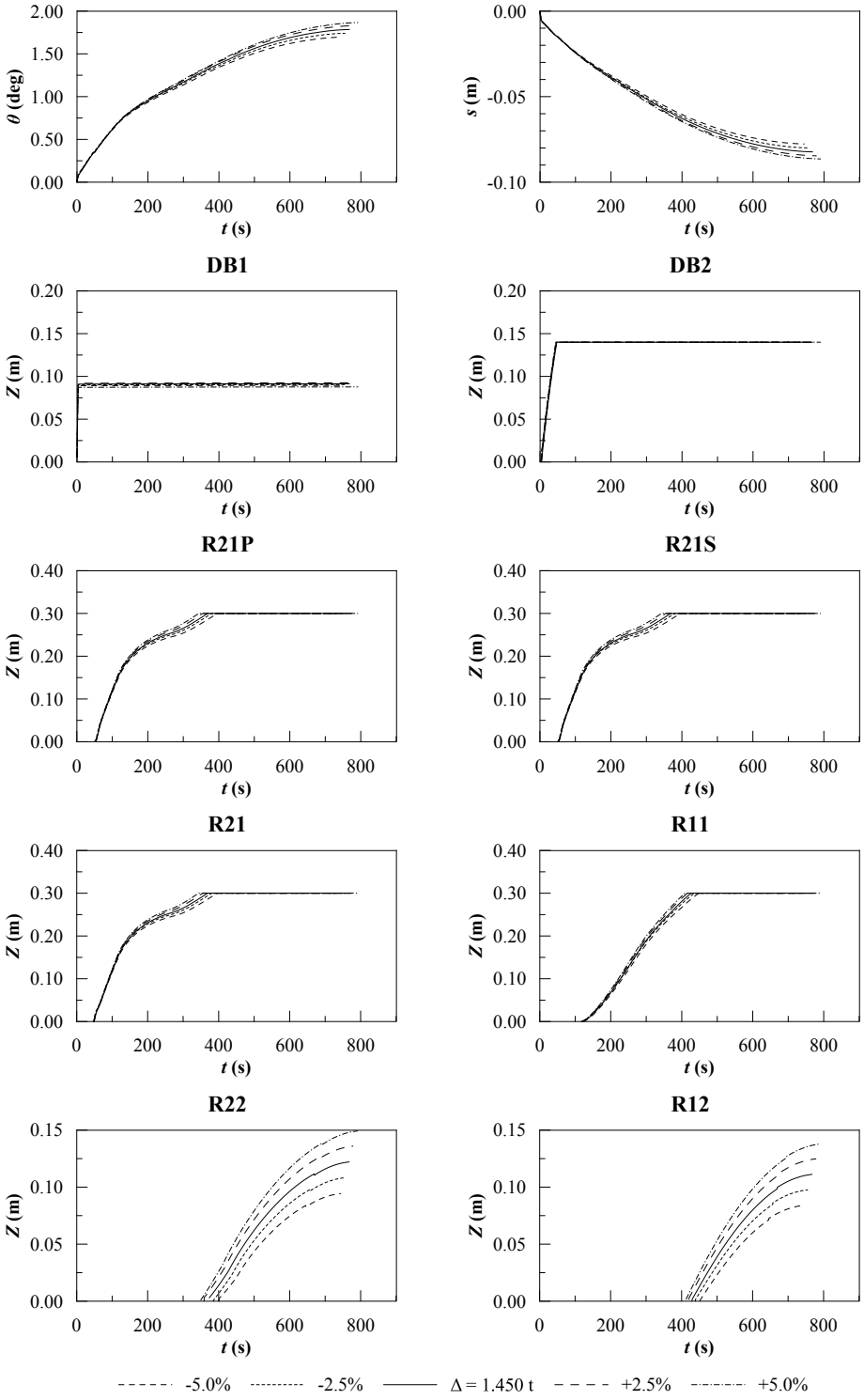

Figure A.16 Overall effect of displacement for cases C

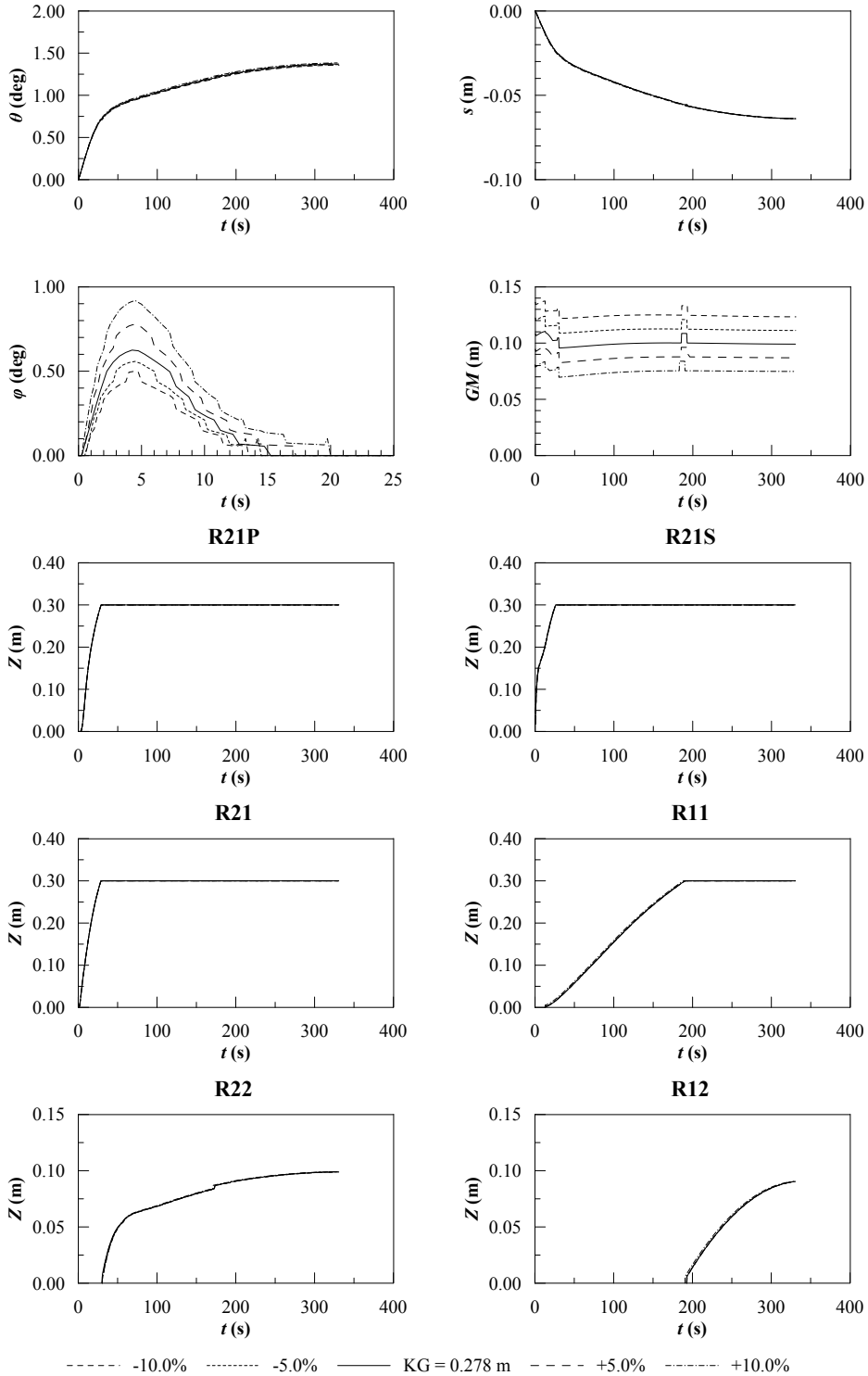

Figure A.17 Overall effect of mass centre height for cases A

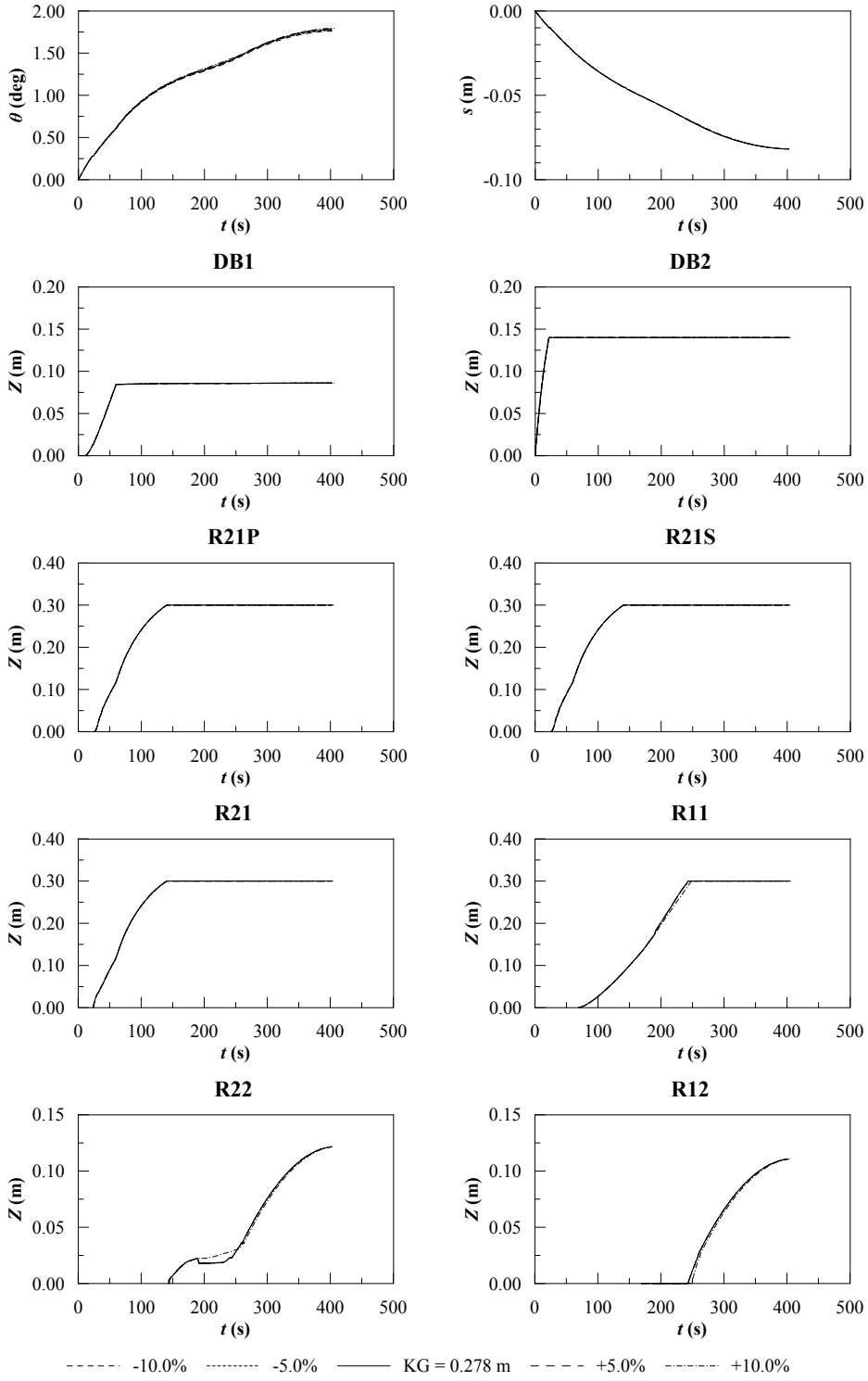

Figure A.18 Overall effect of mass centre height for cases C

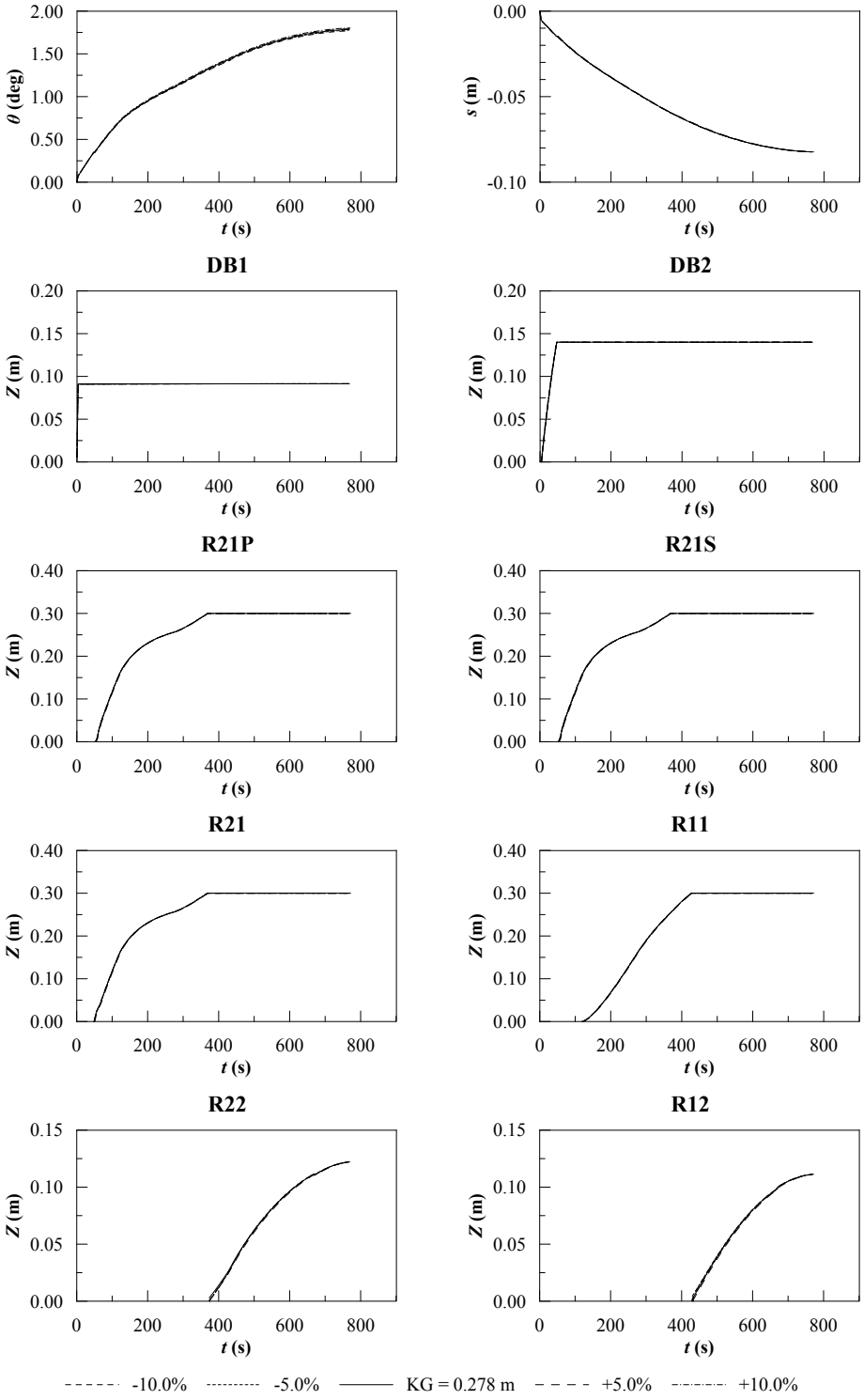

Figure A.19 Overall effect of mass centre height for cases C

A.4 Concluding Remarks

A sensitivity study has been carried out acting on several parameters connected to damage geometry, ship geometry and initial condition at damage occurrence. This analysis provided interesting results, helping to define the minimum set of parameters required to build a complete database of damage scenarios for a single ship.

The damage location appears to be relevant only in terms of the vertical position of the damage centre. Therefore, it is not necessary to investigate all the possible longitudinal and transversal positions for each damaged room. On the other hand, the damage dimension requires more attention, since it was observed that its effect on the progressive flooding process depends upon internal subdivision geometry. In particular, as the damage size increases the effect of internal openings connecting the damaged room to other spaces becomes relevant. This is why it could be reasonable to define a non-dimensional damage size leading to relevant differences in the progressive flooding process. Further study is advisable devoted to assessing such a maximum size as a function of relevant geometrical coefficients describing the damaged room.

Concerning ship geometry, the permeability has not a significant effect on ship initial stability and neither on time-to-flood. These are the most important characteristics of the progressive flooding process required to support decision after damage. Therefore, the application of standard values from classification societies or ITTC could not lead to critical errors in the estimation of the damage scenario's outcome. However, the permeability has an impact on the evolution of the floating position and thus, for more complex hull geometries, could reduce the ship stability too. This is why, whenever it is possible, a detailed analysis of internal volumes is recommended. Concerning discharge coefficients, they only influence the time-to-flood applying a delay/advance equal to their uncertainty. The discharge coefficients are quite difficult to predict a-priori, but a parametric study of the effect of different values could lead to a huge number of different cases even with a limited number of internal openings and thus is actually not feasible for real applications on large passenger ships.

The seawater density was found to be indifferent to progressive flooding simulations provided that constant draughts (and the related different displacement) are applied in the initial condition. Therefore, to assure the accuracy of onboard simulations, it is always advisable to use the displacement assessed from measured draught, assuming even a standard value of water density. This assumption might lead to computational errors in the case of floodwater mixtures with other liquids carried onboard. However, mixtures are not usually taken into account by quasi-static techniques nowadays applied onboard. Concerning displacement and height of mass centre, they have a relevant influence on progressive flooding process leading to relevant differences on studied variables. Thus, to study all the possible damage scenarios of a vessel, not only the standard loading conditions included in the stability booklet should be analysed. To this end, a parametric analysis within all the operative range of the vessel is recommended.

Appendix B

Time-to-flood estimation

In this appendix, a regression analysis on time-to-flood is presented considering one and two parallelepiped rooms in calm water. The methods can be used to estimate a-priori the time-to-flood to define the parameters for database generation.

B.1 Single room geometry

Considering an parallelepiped room in upright position and assuming that its position does not change during flooding process, it can be defined a non-dimensional time-to-flood as:

$$t_a = t_f \sqrt{\frac{g}{z_{rb}}} = f\left(\frac{z}{z_{rb}}, \frac{A}{S}\right) \tag{B.1}$$

where t_f is the time-to-flood in seconds, g is the gravity constant, z_{rb} is the draught of the room bottom measured in earth fixed reference system. The non-dimensional time-to-flood is function of two non-dimensional quantities: z/z_{rb} and A/S, where z and A are the draught of damage centre and its effective area (including the reduction due to discharge coefficient C_d) respectively and S is the waterplane area of the flooded room. The relation can be written as (coefficients provided in Table B.1):

$$\begin{aligned}\frac{1}{t_a} =& a_0 + a_1 \ln\frac{z}{z_{rb}} + a_2 \left(\ln\frac{z}{z_{rb}}\right)^2 + a_3 \left(\ln\frac{z}{z_{rb}}\right)^3 + a_4 \left(\ln\frac{z}{z_{rb}}\right)^4 \\ &+ \frac{A}{S}\left[a_5 + a_6 \ln\frac{z}{z_{rb}} + a_7 \left(\ln\frac{z}{z_{rb}}\right)^2 + a_8 \left(\ln\frac{z}{z_{rb}}\right)^3\right]\end{aligned} \tag{B.2}$$

Table B.1 Coefficients of multivariate regression for single room geometry

a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
-0.003213	-0.024535	-0.030957	-0.012832	-0.001641	0.787854	0.037498	-0.083947	-0.01397

10000 simulations, z/z_{rb} and A/S ranging in $]0,1[$, $R^2 = 0.9999$

B.2 Two rooms geometry

Considering a simple two rooms geometry where the first flooded room is connected with another parallelepiped room by means of an opening having area A_1 and located on the bottom of the first flooded room. Moreover, It is assumed that the second room has a constant free surface area equal to S_1 and has the same bottom draught of the first flooded room.

It is possible to evaluate by means of Equation (B.2) the time-to-flood of the first room only ($t_{a\,min}$) and the one connected to the sum of he volumes of the two rooms ($t_{a\,max}$):

$$\begin{aligned} t_{a\,min} &= f\left(\frac{z}{z_{rb}}, \frac{A}{S}\right) \\ t_{a\,max} &= f\left(\frac{z}{z_{rb}}, \frac{A}{S+S_1}\right) \end{aligned} \tag{B.3}$$

Defining a correction factor as $c_t = t_{a\,min}/t_a$, where t_a is related to the time-to-flood of the first room connected to the second room. The correction factor can be evaluated as a function of two main non-dimensional parameters: A_1/A and S_1/S. The correction factor is equal to one if one of the two parameters is null, otherwise it assumes values within $[1,0[$ range. At a generic S_1/S value, the factor c_t decreases fast reaching a local minimum, than it shows an horizontal asymptote reaching, for $A_1/A \to \infty$, the value:

$$c_{t_\infty} = t_{a\,min}/t_{a\,max} \tag{B.4}$$

It can be noticed that the position of the minimum is independent by the value of S_1/S and it is located at:

$$\left(\frac{A_1}{A}\right)_{min} = 0.202422549 \tag{B.5}$$

Moreover, the value of the minimum can be estimated with the following regression equation (coefficients provided in Table B.3):

$$c_{t_{min}} = 1 + a_1 \exp\left[-b_1\left(\frac{S_1}{S}\right)^{c_1}\right] + a_2 \exp\left[-b_2\left(\frac{S_1}{S}\right)^{c_2}\right] - a_1 - a_2 \tag{B.6}$$

Table B.2 Coefficients of multivariate regression for $c_{t_{min}}$

a_1	b_1	c_1	a_2	b_2	c_2
0.538418736	2.019498758	1.273836007	0.446287131	0.680113795	0.687334693
20 values, S_1/S ranging in $]0,20]$, $SSE = 2.39E-04$, $R^2 = 0.9998$					

Considering the values defined in Equations (B.4), (B.5) and (B.6), the corrective factor can be estimated with the following model for $A_1/A < (A_1/A)_{min}$:

$$
\begin{aligned}
c_t\left(\frac{A_1}{A},\frac{S_1}{S}\right) &= 1-(1-c_{t_{min}})\left[a_1\sin\left(\omega\frac{A_1}{A}\right)+a_2\sin\left(3\omega\frac{A_1}{A}\right)+\right.\\
&\qquad \left.+a_3\sin\left(5\omega\frac{A_1}{A}\right)+a_4\sin\left(7\omega\frac{A_1}{A}\right)\right]\\
a_1\left(\frac{S_1}{S}\right) &= a_{11}\exp\left[-a_{12}\left(\frac{S_1}{S}\right)\right]+1-a_{11}+a_{13}\left(\frac{S_1}{S}\right)^{a_{14}-1}\exp\left[-a_{15}\left(\frac{S_1}{S}\right)^{a_{14}}\right]\\
a_2\left(\frac{S_1}{S}\right) &= a_{21}\exp\left[-a_{22}\left(\frac{S_1}{S}\right)^{a_{23}}\right]-a_{21}\\
a_3\left(\frac{S_1}{S}\right) &= \min\left(0,\, a_{31}\exp\left[-a_{32}\left(\frac{S_1}{S}-a_{33}\right)\right]-a_{31}\right)\\
a_4\left(\frac{S_1}{S}\right) &= a_1\left(\frac{S_1}{S}\right)-a_2\left(\frac{S_1}{S}\right)+a_3\left(\frac{S_1}{S}\right)-1\\
\omega &= \frac{\pi}{2\left(\frac{A_1}{A}\right)_{min}}
\end{aligned}
\tag{B.7}
$$

and for $A_1/A > (A_1/A)_{min}$:

$$
\begin{aligned}
c_t\left(\frac{A_1}{A},\frac{S_1}{S}\right) &= (c_{t_\infty}-c_{t_{min}})\arctan\left(\frac{2a_5}{\pi}\frac{A_1}{A}\right)-c_{t_{min}}\\
a_5\left(\frac{S_1}{S}\right) &= a_{51}\left(\frac{S_1}{S}\right)^{-a_{52}}+a_{53}\left(\frac{S_1}{S}\right)^{-a_{54}}+a_{55}
\end{aligned}
\tag{B.8}
$$

Table B.3 Coefficients of multivariate regression up to $(A_1/A)_{min}$

i	a_{i1}	a_{i2}	a_{i3}	a_{i4}	a_{i5}	SSE	R^2
1	0.034601445	2.760871221	-0.164917431	1.323666382	0.390427418	0.0018	0.9709
2	0.207390573	2.237779056	1.451066079	-	-	0.0007	0.9850
3	0.115200923	0.370945666	0.901206923	-	-	0.0013	0.9311
5	1.34E-06	12.18022137	9.78825515	0.957288337	1.668287856	0.8644	0.9998

20 values, S_1/S ranging in $]0,40]$, A_1/A ranging in $]0,2]$

Appendix C

Comparison of different learners

In the present appendix additional data to compare the performances of the tested learners are provided. In detail, all the learners are compared grouped by classification problem and considered time instant t^* as follows:

- Final fate: $t^* = 250$ s: Tables C.1-C.11; $t^* = 500$ s: Tables C.12-C.22
- Flooded compartments: $t^* = 250$ s: Tables C.23-C.33; $t^* = 500$ s: Tables C.34-C.44
- Time-to-flood: Figures C.1-C.11

Table C.1 Confusion matrix, ship final fate; $t^* = 250$ s; training: MC20; validation: MC50b; method: DT

True Class	**Predicted Class (%)**			
	Capsize	**Equilibrium**	**Ex. Heeling**	**Time Exceeded**
Capsize	12.73	0.22	0.04	0.01
Equilibrium	0.19	82.71	0.00	0.27
Ex. Heeling	0.05	0.02	3.04	0.00
Time Exceeded	0.01	0.30	0.00	0.398

Table C.2 Confusion matrix, ship final fate; $t^* = 250$ s; training: MC20; validation: MC50b; method: DTB

True Class	**Predicted Class (%)**			
	Capsize	**Equilibrium**	**Ex. Heeling**	**Time Exceeded**
Capsize	12.75	0.18	0.01	0.06
Equilibrium	0.35	80.17	0.02	2.63
Ex. Heeling	0.01	0.03	3.06	0.01
Time Exceeded	0.00	0.02	0.00	0.69

Table C.3 Confusion matrix, ship final fate; $t^* = 250$ s; training: MC20; validation: MC50b; method: RF

True Class	Predicted Class (%)			
	Capsize	**Equilibrium**	**Ex. Heeling**	**Time Exceeded**
Capsize	12.79	0.17	0.01	0.02
Equilibrium	0.07	82.97	0.00	0.12
Ex. Heeling	0.04	0.03	3.05	0.00
Time Exceeded	0.02	0.32	0.00	0.37

Table C.4 Confusion matrix, ship final fate; $t^* = 250$ s; training: MC20; validation: MC50b; method: KNN2

True Class	Predicted Class (%)			
	Capsize	**Equilibrium**	**Ex. Heeling**	**Time Exceeded**
Capsize	12.65	0.29	0.04	0.01
Equilibrium	0.08	82.95	0.00	0.14
Ex. Heeling	0.11	0.04	2.97	0.00
Time Exceeded	0.00	0.50	0.00	0.21

Table C.5 Confusion matrix, ship final fate; $t^* = 250$ s; training: MC20; validation: MC50b; method: KNN3

True Class	Predicted Class (%)			
	Capsize	**Equilibrium**	**Ex. Heeling**	**Time Exceeded**
Capsize	12.62	0.32	0.05	0.01
Equilibrium	0.09	82.94	0.01	0.14
Ex. Heeling	0.12	0.04	2.96	0.00
Time Exceeded	0.00	0.50	0.00	0.21

Table C.6 Confusion matrix, ship final fate; $t^* = 250$ s; training: MC20; validation: MC50b; method: KNNC

True Class	Predicted Class (%)			
	Capsize	**Equilibrium**	**Ex. Heeling**	**Time Exceeded**
Capsize	12.64	0.28	0.07	0.01
Equilibrium	0.10	82.92	0.01	0.14
Ex. Heeling	0.12	0.04	2.96	0.00
Time Exceeded	0.01	0.48	0.00	0.23

Table C.7 Confusion matrix, ship final fate; $t^* = 250$ s; training: MC20; validation: MC50b; method: KNNW

True Class	Predicted Class (%)			
	Capsize	**Equilibrium**	**Ex. Heeling**	**Time Exceeded**
Capsize	12.70	0.25	0.04	0.01
Equilibrium	0.06	82.90	0.01	0.21
Ex. Heeling	0.06	0.04	3.02	0.00
Time Exceeded	0.01	0.34	0.00	0.36

Table C.8 Confusion matrix, ship final fate; $t^* = 250$ s; training: MC20; validation: MC50b; method: SVM1

True Class	Predicted Class (%)			
	Capsize	**Equilibrium**	**Ex. Heeling**	**Time Exceeded**
Capsize	12.20	0.65	0.15	0.00
Equilibrium	0.30	82.86	0.01	0.00
Ex. Heeling	0.39	0.05	2.68	0.00
Time Exceeded	0.01	0.71	0.00	0.00

Table C.9 Confusion matrix, ship final fate; $t^* = 250$ s; training: MC20; validation: MC50b; method: SVM2

True Class	Predicted Class (%)			
	Capsize	**Equilibrium**	**Ex. Heeling**	**Time Exceeded**
Capsize	12.58	0.34	0.05	0.02
Equilibrium	0.08	83.00	0.01	0.09
Ex. Heeling	0.02	0.04	3.06	0.00
Time Exceeded	0.01	0.65	0.00	0.05

Table C.10 Confusion matrix, ship final fate; $t^* = 250$ s; training: MC20; validation: MC50b; method: SVM3

True Class	Predicted Class (%)			
	Capsize	**Equilibrium**	**Ex. Heeling**	**Time Exceeded**
Capsize	12.61	0.33	0.05	0.02
Equilibrium	0.05	82.66	0.01	0.46
Ex. Heeling	0.02	0.04	3.06	0.00
Time Exceeded	0.00	0.46	0.00	0.25

Table C.11 Confusion matrix, ship final fate; $t^* = 250$ s; training: MC20; validation: MC50b; method: SVMG

True Class	Predicted Class (%)			
	Capsize	**Equilibrium**	**Ex. Heeling**	**Time Exceeded**
Capsize	12.58	0.40	0.02	0.00
Equilibrium	0.09	83.08	0.00	0.00
Ex. Heeling	0.12	0.04	2.95	0.00
Time Exceeded	0.00	0.71	0.00	0.00

Table C.12 Confusion matrix, ship final fate; $t^* = 500$ s; training: MC20; validation: MC50b; method: DT

True Class	Predicted Class (%)			
	Capsize	**Equilibrium**	**Ex. Heeling**	**Time Exceeded**
Capsize	12.76	0.17	0.06	0.01
Equilibrium	0.08	82.92	0.00	0.17
Ex. Heeling	0.02	0.02	3.05	0.00
Time Exceeded	0.03	0.13	0.00	0.54

Table C.13 Confusion matrix, ship final fate; $t^* = 500$ s; training: MC20; validation: MC50b; method: DTB

True Class	Predicted Class (%)			
	Capsize	**Equilibrium**	**Ex. Heeling**	**Time Exceeded**
Capsize	12.88	0.08	0.00	0.03
Equilibrium	0.26	81.86	0.02	1.03
Ex. Heeling	0.01	0.01	3.09	0.00
Time Exceeded	0.01	0.04	0.00	0.66

Table C.14 Confusion matrix, ship final fate; $t^* = 500$ s; training: MC20; validation: MC50b; method: RF

True Class	Predicted Class (%)			
	Capsize	**Equilibrium**	**Ex. Heeling**	**Time Exceeded**
Capsize	12.89	0.09	0.00	0.01
Equilibrium	0.05	83.00	0.00	0.12
Ex. Heeling	0.01	0.02	3.05	0.00
Time Exceeded	0.02	0.17	0.00	0.52

Table C.15 Confusion matrix, ship final fate; $t^* = 500$ s; training: MC20; validation: MC50b; method: KNN2

True Class	Predicted Class (%)			
	Capsize	**Equilibrium**	**Ex. Heeling**	**Time Exceeded**
Capsize	12.76	0.19	0.03	0.02
Equilibrium	0.08	82.98	0.00	0.11
Ex. Heeling	0.06	0.02	3.03	0.01
Time Exceeded	0.01	0.40	0.00	0.30

Table C.16 Confusion matrix, ship final fate; $t^* = 500$ s; training: MC20; validation: MC50b; method: KNN3

True Class	Predicted Class (%)			
	Capsize	**Equilibrium**	**Ex. Heeling**	**Time Exceeded**
Capsize	12.70	0.22	0.06	0.02
Equilibrium	0.07	82.99	0.00	0.10
Ex. Heeling	0.10	0.01	3.00	0.01
Time Exceeded	0.01	0.41	0.00	0.30

Table C.17 Confusion matrix, ship final fate; $t^* = 500$ s; training: MC20; validation: MC50b; method: KNNC

True Class	Predicted Class (%)			
	Capsize	**Equilibrium**	**Ex. Heeling**	**Time Exceeded**
Capsize	12.75	0.14	0.09	0.02
Equilibrium	0.09	82.97	0.01	0.10
Ex. Heeling	0.12	0.02	2.98	0.00
Time Exceeded	0.02	0.37	0.00	0.32

Table C.18 Confusion matrix, ship final fate; $t^* = 500$ s; training: MC20; validation: MC50b; method: KNNW

True Class	Predicted Class (%)			
	Capsize	**Equilibrium**	**Ex. Heeling**	**Time Exceeded**
Capsize	12.79	0.15	0.04	0.02
Equilibrium	0.04	83.01	0.00	0.11
Ex. Heeling	0.03	0.01	3.07	0.00
Time Exceeded	0.01	0.29	0.00	0.42

Table C.19 Confusion matrix, ship final fate; $t^* = 500$ s; training: MC20; validation: MC50b; method: SVM1

True Class	Predicted Class (%)			
	Capsize	**Equilibrium**	**Ex. Heeling**	**Time Exceeded**
Capsize	12.35	0.50	0.15	0.00
Equilibrium	0.36	82.80	0.01	0.00
Ex. Heeling	0.37	0.03	2.73	0.00
Time Exceeded	0.00	0.71	0.00	0.00

Table C.20 Confusion matrix, ship final fate; $t^* = 500$ s; training: MC20; validation: MC50b; method: SVM2

True Class	Predicted Class (%)			
	Capsize	**Equilibrium**	**Ex. Heeling**	**Time Exceeded**
Capsize	12.72	0.21	0.04	0.03
Equilibrium	0.06	83.02	0.00	0.08
Ex. Heeling	0.01	0.02	3.09	0.00
Time Exceeded	0.02	0.46	0.00	0.23

Table C.21 Confusion matrix, ship final fate; $t^* = 500$ s; training: MC20; validation: MC50b; method: SVM3

True Class	Predicted Class (%)			
	Capsize	**Equilibrium**	**Ex. Heeling**	**Time Exceeded**
Capsize	12.73	0.19	0.05	0.02
Equilibrium	0.03	83.01	0.01	0.12
Ex. Heeling	0.01	0.02	3.09	0.00
Time Exceeded	0.01	0.31	0.00	0.40

Table C.22 Confusion matrix, ship final fate; $t^* = 500$ s; training: MC20; validation: MC50b; method: SVMG

True Class	Predicted Class (%)			
	Capsize	**Equilibrium**	**Ex. Heeling**	**Time Exceeded**
Capsize	12.79	0.19	0.00	0.02
Equilibrium	0.05	83.09	0.00	0.03
Ex. Heeling	0.06	0.02	3.05	0.00
Time Exceeded	0.01	0.52	0.00	0.18

Table C.23 Confusion matrix, flooded compartments; $t^* = 250$ s; training: MC20; validation: MC50b; method: DT

True Class	Predicted Class (%)												
	1	**2**	**3**	**4**	**5**	**1,2**	**2,3**	**3,4**	**4,5**	**1,2,3**	**2,3,4**	**3,4,5**	**all**
1	17.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
2	0.02	13.31	0.00	0.00	0.00	0.00	0.07	0.00	0.00	0.00	0.00	0.00	0.01
3	0.00	0.00	13.61	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	13.37	0.02	0.00	0.00	0.04	0.01	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.01	16.51	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00
1,2	0.01	0.00	0.00	0.00	0.00	1.52	0.00	0.00	0.00	0.00	0.00	0.00	0.28
2,3	0.00	0.13	0.00	0.00	0.00	0.00	5.61	0.00	0.00	0.00	0.01	0.00	0.03
3,4	0.00	0.01	0.00	0.08	0.02	0.00	0.01	5.75	0.10	0.00	0.01	0.01	0.12
4,5	0.00	0.00	0.00	0.02	0.05	0.00	0.00	0.09	4.16	0.00	0.00	0.04	0.48
1,2,3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.01
2,3,4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.06	0.00	0.02
3,4,5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.02	0.00	0.00	0.01	0.01
all	0.02	0.05	0.00	0.01	0.04	0.37	0.05	0.20	0.36	0.02	0.01	0.02	6.01

Table C.24 Confusion matrix, flooded compartments; $t^* = 250$ s; training: MC20; validation: MC50b; method: DTB

True Class	Predicted Class (%)												
	1	**2**	**3**	**4**	**5**	**1,2**	**2,3**	**3,4**	**4,5**	**1,2,3**	**2,3,4**	**3,4,5**	**all**
1	17.08	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.44	12.79	0.00	0.00	0.00	0.00	0.11	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	13.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.10	13.27	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00
5	0.00	0.02	0.00	1.44	14.99	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00
1,2	0.06	0.00	0.00	0.00	0.00	1.64	0.00	0.01	0.00	0.09	0.00	0.00	0.02
2,3	0.03	0.14	0.01	0.00	0.00	0.00	5.35	0.00	0.00	0.04	0.17	0.00	0.05
3,4	0.00	0.00	0.05	0.22	0.04	0.00	0.00	4.83	0.13	0.00	0.02	0.81	0.00
4,5	0.00	0.01	0.00	0.02	0.15	0.00	0.00	0.07	3.72	0.00	0.00	0.78	0.08
1,2,3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00
2,3,4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.09	0.00	0.00
3,4,5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.06	0.00
all	0.24	0.04	0.01	0.01	0.31	0.70	0.11	0.09	0.29	0.46	0.14	0.30	4.50

Table C.25 Confusion matrix, flooded compartments; $t^* = 250$ s; training: MC20; validation: MC50b; method: RF

True Class	Predicted Class (%)												
	1	**2**	**3**	**4**	**5**	**1,2**	**2,3**	**3,4**	**4,5**	**1,2,3**	**2,3,4**	**3,4,5**	**all**
1	17.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	13.37	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.03
3	0.00	0.00	13.62	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	13.43	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	16.53	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1,2	0.01	0.00	0.00	0.00	0.00	1.53	0.00	0.00	0.00	0.00	0.00	0.00	0.26
2,3	0.00	0.09	0.00	0.00	0.00	0.00	5.67	0.00	0.00	0.00	0.00	0.00	0.01
3,4	0.00	0.01	0.00	0.06	0.02	0.00	0.00	5.92	0.04	0.00	0.00	0.01	0.05
4,5	0.00	0.00	0.00	0.00	0.06	0.00	0.00	0.01	4.56	0.00	0.00	0.01	0.21
1,2,3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.02
2,3,4	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.00	0.00	0.07	0.00	0.00
3,4,5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.03	0.00	0.00	0.01	0.00
all	0.02	0.04	0.00	0.01	0.04	0.28	0.02	0.20	0.36	0.01	0.01	0.02	6.18

Table C.26 Confusion matrix, flooded compartments; $t^* = 250$ s; training: MC20; validation: MC50b; method: KNN2

True	Predicted Class (%)												
Class	**1**	**2**	**3**	**4**	**5**	**1,2**	**2,3**	**3,4**	**4,5**	**1,2,3**	**2,3,4**	**3,4,5**	**all**
1	17.03	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.01	13.38	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.01
3	0.00	0.00	13.62	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.01	13.42	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.01	0.02	16.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1,2	0.01	0.00	0.00	0.00	0.00	1.60	0.00	0.00	0.00	0.00	0.00	0.00	0.19
2,3	0.01	0.20	0.13	0.00	0.00	0.00	5.43	0.00	0.00	0.00	0.00	0.00	0.01
3,4	0.00	0.04	0.18	0.12	0.00	0.00	0.02	5.47	0.21	0.00	0.00	0.00	0.07
4,5	0.00	0.00	0.00	0.02	0.06	0.00	0.00	0.06	4.65	0.00	0.00	0.00	0.05
1,2,3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.02
2,3,4	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.04	0.00	0.00	0.04	0.00	0.01
3,4,5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.04	0.00	0.00	0.00	0.01
all	0.06	0.06	0.02	0.03	0.05	0.33	0.03	0.24	0.45	0.00	0.00	0.00	5.90

Table C.27 Confusion matrix, flooded compartments; $t^* = 250$ s; training: MC20; validation: MC50b; method: KNN3

True	Predicted Class (%)												
Class	**1**	**2**	**3**	**4**	**5**	**1,2**	**2,3**	**3,4**	**4,5**	**1,2,3**	**2,3,4**	**3,4,5**	**all**
1	17.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.01	13.37	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.01
3	0.00	0.00	13.62	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.01	13.43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.01	0.02	16.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1,2	0.01	0.00	0.00	0.00	0.00	1.61	0.00	0.01	0.00	0.00	0.00	0.00	0.18
2,3	0.01	0.19	0.13	0.00	0.00	0.00	5.43	0.00	0.00	0.00	0.00	0.00	0.02
3,4	0.01	0.06	0.19	0.13	0.00	0.00	0.01	5.41	0.23	0.00	0.00	0.00	0.07
4,5	0.00	0.02	0.00	0.02	0.05	0.00	0.00	0.07	4.63	0.00	0.00	0.00	0.05
1,2,3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.02
2,3,4	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.04	0.00	0.00	0.03	0.00	0.01
3,4,5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.04	0.00	0.00	0.00	0.01
all	0.07	0.06	0.01	0.03	0.06	0.36	0.04	0.25	0.44	0.01	0.00	0.00	5.86

Table C.28 Confusion matrix, flooded compartments; $t^* = 250$ s; training: MC20; validation: MC50b; method: KNNC

True	Predicted Class (%)												
Class	**1**	**2**	**3**	**4**	**5**	**1,2**	**2,3**	**3,4**	**4,5**	**1,2,3**	**2,3,4**	**3,4,5**	**all**
1	17.03	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.01	13.37	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.02
3	0.00	0.00	13.62	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.01	0.00	0.01	13.42	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.04	0.01	0.02	16.46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1,2	0.01	0.00	0.00	0.00	0.00	1.62	0.00	0.00	0.00	0.00	0.00	0.00	0.18
2,3	0.01	0.19	0.12	0.00	0.00	0.00	5.44	0.00	0.00	0.00	0.00	0.00	0.01
3,4	0.01	0.04	0.17	0.11	0.00	0.00	0.01	5.46	0.24	0.00	0.00	0.00	0.07
4,5	0.00	0.01	0.00	0.02	0.05	0.00	0.00	0.04	4.65	0.00	0.00	0.00	0.07
1,2,3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.02
2,3,4	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.04	0.00	0.00	0.04	0.00	0.01
3,4,5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.04	0.00	0.00	0.00	0.01
all	0.06	0.05	0.01	0.03	0.04	0.36	0.04	0.22	0.45	0.01	0.00	0.00	5.91

Table C.29 Confusion matrix, flooded compartments; $t^* = 250$ s; training: MC20; validation: MC50b; method: KNNW

True	Predicted Class (%)												
Class	**1**	**2**	**3**	**4**	**5**	**1,2**	**2,3**	**3,4**	**4,5**	**1,2,3**	**2,3,4**	**3,4,5**	**all**
1	17.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	13.38	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.01
3	0.00	0.00	13.61	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	13.43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.01	0.01	16.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1,2	0.01	0.00	0.00	0.00	0.00	1.57	0.00	0.00	0.00	0.00	0.00	0.00	0.23
2,3	0.00	0.15	0.09	0.00	0.00	0.00	5.53	0.00	0.00	0.00	0.00	0.00	0.01
3,4	0.00	0.04	0.11	0.11	0.00	0.00	0.02	5.59	0.16	0.00	0.00	0.00	0.08
4,5	0.00	0.00	0.00	0.02	0.05	0.00	0.00	0.03	4.54	0.00	0.00	0.01	0.19
1,2,3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.02
2,3,4	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.03	0.00	0.00	0.06	0.00	0.00
3,4,5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.03	0.00	0.00	0.01	0.01
all	0.04	0.05	0.01	0.02	0.05	0.31	0.02	0.20	0.39	0.01	0.01	0.00	6.07

Table C.30 Confusion matrix, flooded compartments; $t^* = 250$ s; training: MC20; validation: MC50b; method: SVM1

True	Predicted Class (%)												
Class	**1**	**2**	**3**	**4**	**5**	**1,2**	**2,3**	**3,4**	**4,5**	**1,2,3**	**2,3,4**	**3,4,5**	**all**
1	16.93	0.00	0.00	0.01	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.05	13.31	0.00	0.01	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.02
3	0.00	0.00	13.38	0.24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.01	13.43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.06	0.00	0.04	0.32	16.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1,2	0.01	0.00	0.00	0.00	0.00	1.48	0.00	0.01	0.00	0.00	0.00	0.00	0.31
2,3	0.02	0.11	0.00	0.04	0.00	0.00	5.53	0.00	0.00	0.00	0.01	0.00	0.07
3,4	0.01	0.00	0.05	0.06	0.01	0.00	0.00	5.84	0.05	0.00	0.00	0.00	0.09
4,5	0.02	0.00	0.01	0.03	0.02	0.00	0.00	0.00	4.73	0.00	0.00	0.00	0.04
1,2,3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04
2,3,4	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.01	0.00	0.00	0.00	0.00	0.07
3,4,5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.03	0.00	0.00	0.00	0.00
all	0.07	0.08	0.00	0.05	0.06	0.37	0.02	0.29	0.55	0.00	0.00	0.00	5.68

Table C.31 Confusion matrix, flooded compartments; $t^* = 250$ s; training: MC20; validation: MC50b; method: SVM2

True	Predicted Class (%)												
Class	**1**	**2**	**3**	**4**	**5**	**1,2**	**2,3**	**3,4**	**4,5**	**1,2,3**	**2,3,4**	**3,4,5**	**all**
1	16.89	0.00	0.00	0.15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.04	13.33	0.00	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.03
3	0.00	0.00	13.62	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.01	13.42	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.06	0.01	0.00	0.20	16.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1,2	0.01	0.00	0.00	0.00	0.00	1.49	0.00	0.00	0.00	0.00	0.00	0.00	0.31
2,3	0.01	0.09	0.00	0.01	0.00	0.00	5.65	0.00	0.00	0.00	0.00	0.00	0.02
3,4	0.00	0.00	0.03	0.06	0.00	0.00	0.01	5.93	0.04	0.00	0.01	0.00	0.03
4,5	0.01	0.00	0.00	0.04	0.01	0.00	0.00	0.01	4.68	0.00	0.00	0.01	0.08
1,2,3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.02
2,3,4	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.00	0.00	0.07	0.00	0.00
3,4,5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.03	0.00	0.00	0.00	0.01
all	0.02	0.03	0.00	0.05	0.02	0.31	0.01	0.24	0.39	0.02	0.02	0.00	6.06

Table C.32 Confusion matrix, flooded compartments; $t^* = 250$ s; training: MC20; validation: MC50b; method: SVM3

True	Predicted Class (%)												
Class	1	2	3	4	5	1,2	2,3	3,4	4,5	1,2,3	2,3,4	3,4,5	all
1	16.92	0.00	0.12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.04	0.13	0.00	0.00	0.00	0.00	13.14	0.00	0.00	0.00	0.00	0.00	0.10
3	0.00	0.00	13.62	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.01	13.42	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00
5	0.05	0.00	0.00	0.08	16.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1,2	0.01	0.00	0.00	0.00	0.00	1.52	0.00	0.00	0.00	0.00	0.00	0.00	0.28
2,3	0.01	0.01	0.00	0.01	0.00	0.00	5.69	0.00	0.00	0.01	0.01	0.00	0.04
3,4	0.00	0.01	0.01	0.06	0.00	0.00	0.01	5.95	0.03	0.01	0.01	0.00	0.03
4,5	0.00	0.00	0.00	0.01	0.05	0.00	0.00	0.03	2.15	0.00	0.00	0.00	2.59
1,2,3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.02
2,3,4	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.07	0.00	0.01
3,4,5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.01	0.00	0.00	0.01	0.02
all	0.01	0.01	0.01	0.04	0.02	0.34	0.04	0.25	0.16	0.12	0.01	0.01	6.17

Table C.33 Confusion matrix, flooded compartments; $t^* = 250$ s; training: MC20; validation: MC50b; method: SVMG

True	Predicted Class (%)												
Class	1	2	3	4	5	1,2	2,3	3,4	4,5	1,2,3	2,3,4	3,4,5	all
1	16.88	0.00	0.00	0.16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.04	13.32	0.00	0.01	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.03
3	0.00	0.00	13.62	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.02	13.42	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.02	0.03	0.00	0.72	15.76	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1,2	0.01	0.00	0.00	0.00	0.00	1.50	0.00	0.00	0.00	0.00	0.00	0.00	0.30
2,3	0.02	0.19	0.00	0.02	0.01	0.00	5.50	0.00	0.00	0.00	0.00	0.00	0.04
3,4	0.00	0.00	0.05	0.08	0.00	0.00	0.01	5.82	0.10	0.00	0.00	0.00	0.05
4,5	0.01	0.01	0.00	0.05	0.00	0.00	0.00	0.01	4.72	0.00	0.00	0.00	0.05
1,2,3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04
2,3,4	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.03	0.00	0.00	0.03	0.00	0.03
3,4,5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.04	0.00	0.00	0.00	0.00
all	0.02	0.11	0.00	0.06	0.02	0.39	0.03	0.25	0.50	0.00	0.00	0.00	5.79

Table C.34 Confusion matrix, flooded compartments; $t^* = 500$ s; training: MC20; validation: MC50b; method: DT

True	Predicted Class (%)												
Class	1	2	3	4	5	1,2	2,3	3,4	4,5	1,2,3	2,3,4	3,4,5	all
1	17.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.01	13.33	0.00	0.00	0.01	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.03
3	0.00	0.00	13.61	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	13.39	0.02	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.01
5	0.00	0.00	0.00	0.06	16.46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1,2	0.00	0.00	0.00	0.00	0.00	1.52	0.00	0.00	0.00	0.00	0.00	0.00	0.28
2,3	0.00	0.06	0.00	0.00	0.00	0.00	5.66	0.00	0.00	0.00	0.01	0.00	0.05
3,4	0.00	0.01	0.00	0.05	0.01	0.00	0.01	5.83	0.06	0.00	0.00	0.01	0.14
4,5	0.00	0.00	0.00	0.01	0.03	0.00	0.00	0.07	4.30	0.00	0.00	0.01	0.43
1,2,3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.02
2,3,4	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.03	0.00	0.00	0.05	0.00	0.02
3,4,5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.02	0.00	0.00	0.00	0.01
all	0.01	0.03	0.00	0.01	0.02	0.34	0.02	0.20	0.37	0.02	0.02	0.00	6.14

Table C.35 Confusion matrix, flooded compartments; $t^* = 500$ s; training: MC20; validation: MC50b; method: DTB

True Class	Predicted Class (%)												
	1	**2**	**3**	**4**	**5**	**1,2**	**2,3**	**3,4**	**4,5**	**1,2,3**	**2,3,4**	**3,4,5**	**all**
1	17.07	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.31	12.27	0.00	0.00	0.00	0.00	0.74	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	13.69	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.09	13.28	0.01	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00
5	0.00	0.01	0.01	0.72	15.75	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00
1,2	0.04	0.00	0.00	0.00	0.00	1.64	0.00	0.00	0.01	0.09	0.00	0.00	0.04
2,3	0.01	0.08	0.01	0.00	0.00	0.00	5.56	0.00	0.00	0.04	0.06	0.00	0.04
3,4	0.00	0.00	0.03	0.19	0.02	0.00	0.00	4.99	0.19	0.00	0.01	0.69	0.00
4,5	0.00	0.01	0.01	0.01	0.11	0.00	0.00	0.04	3.83	0.00	0.00	0.75	0.08
1,2,3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00
2,3,4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.09	0.00	0.00
3,4,5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.05	0.00
all	0.15	0.01	0.03	0.00	0.72	0.75	0.10	0.10	0.34	0.45	0.09	0.25	4.21

Table C.36 Confusion matrix, flooded compartments; $t^* = 500$ s; training: MC20; validation: MC50b; method: RF

True Class	Predicted Class (%)												
	1	**2**	**3**	**4**	**5**	**1,2**	**2,3**	**3,4**	**4,5**	**1,2,3**	**2,3,4**	**3,4,5**	**all**
1	17.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.01	13.39	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
3	0.00	0.00	13.62	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	13.42	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.01	16.52	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1,2	0.00	0.00	0.00	0.00	0.00	1.52	0.00	0.00	0.00	0.00	0.00	0.00	0.28
2,3	0.00	0.06	0.00	0.00	0.00	0.00	5.70	0.00	0.00	0.00	0.00	0.00	0.02
3,4	0.00	0.00	0.00	0.04	0.01	0.00	0.00	5.96	0.03	0.00	0.00	0.01	0.06
4,5	0.00	0.01	0.00	0.00	0.03	0.00	0.00	0.01	4.63	0.00	0.00	0.01	0.16
1,2,3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.02
2,3,4	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.00	0.00	0.07	0.00	0.01
3,4,5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.03	0.00	0.00	**0.01**	0.00
all	0.01	0.04	0.00	0.01	0.03	0.26	0.01	0.19	0.37	0.01	0.01	0.02	6.23

Table C.37 Confusion matrix, flooded compartments; $t^* = 500$ s; training: MC20; validation: MC50b; method: KNN2

True Class	Predicted Class (%)												
	1	**2**	**3**	**4**	**5**	**1,2**	**2,3**	**3,4**	**4,5**	**1,2,3**	**2,3,4**	**3,4,5**	**all**
1	17.02	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.02	13.37	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01
3	0.00	0.00	13.62	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	13.43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.02	0.01	0.05	16.45	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1,2	0.00	0.00	0.00	0.00	0.00	1.61	0.00	0.00	0.00	0.00	0.00	0.00	0.19
2,3	0.01	0.16	0.10	0.00	0.00	0.00	5.50	0.00	0.00	0.00	0.00	0.00	0.01
3,4	0.00	0.03	0.11	0.14	0.00	0.00	0.01	5.54	0.21	0.00	0.00	0.00	0.07
4,5	0.00	0.01	0.00	0.00	0.03	0.00	0.00	0.06	4.68	0.00	0.00	0.00	0.05
1,2,3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.02
2,3,4	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.04	0.00	0.00	0.04	0.00	0.01
3,4,5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.04	0.00	0.00	0.00	0.01
all	0.03	0.06	0.01	0.02	0.05	0.35	0.02	0.25	0.45	0.01	0.01	0.00	5.93

Table C.38 Confusion matrix, flooded compartments; $t^* = 500$ s; training: MC20; validation: MC50b; method: KNN3

True Class	Predicted Class (%)												
	1	**2**	**3**	**4**	**5**	**1,2**	**2,3**	**3,4**	**4,5**	**1,2,3**	**2,3,4**	**3,4,5**	**all**
1	17.02	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.03	13.36	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.01
3	0.00	0.00	13.62	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	13.43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.03	0.01	0.05	16.43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1,2	0.00	0.00	0.00	0.00	0.00	1.62	0.00	0.01	0.00	0.00	0.00	0.00	0.17
2,3	0.01	0.17	0.11	0.00	0.00	0.00	5.47	0.00	0.00	0.00	0.00	0.00	0.02
3,4	0.00	0.06	0.13	0.13	0.00	0.00	0.01	5.46	0.24	0.00	0.00	0.00	0.08
4,5	0.00	0.02	0.00	0.00	0.03	0.00	0.00	0.07	4.66	0.00	0.00	0.00	0.06
1,2,3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.02
2,3,4	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.04	0.00	0.00	0.04	0.00	0.01
3,4,5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.04	0.00	0.00	0.00	0.01
all	0.03	0.10	0.00	0.02	0.05	0.37	0.02	0.25	0.47	0.01	0.01	0.00	5.87

Table C.39 Confusion matrix, flooded compartments; $t^* = 500$ s; training: MC20; validation: MC50b; method: KNNC

True Class	Predicted Class (%)												
	1	**2**	**3**	**4**	**5**	**1,2**	**2,3**	**3,4**	**4,5**	**1,2,3**	**2,3,4**	**3,4,5**	**all**
1	17.03	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.02	13.36	0.00	0.00	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.02
3	0.00	0.00	13.62	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.01	0.00	0.00	13.42	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.03	0.01	0.06	16.43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1,2	0.00	0.00	0.00	0.00	0.00	1.62	0.00	0.00	0.01	0.00	0.00	0.00	0.17
2,3	0.01	0.16	0.09	0.00	0.00	0.00	5.51	0.00	0.00	0.00	0.00	0.00	0.01
3,4	0.00	0.02	0.11	0.13	0.01	0.00	0.01	5.52	0.24	0.00	0.00	0.00	0.07
4,5	0.00	0.02	0.00	0.00	0.01	0.00	0.00	0.05	4.68	0.00	0.00	0.00	0.08
1,2,3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.02
2,3,4	0.00	0.00	0.01	0.00	0.00	0.00	0.01	0.04	0.00	0.00	0.04	0.00	0.01
3,4,5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.04	0.00	0.00	0.00	0.01
all	0.02	0.05	0.01	0.02	0.04	0.35	0.02	0.22	0.45	0.01	0.00	0.00	5.99

Table C.40 Confusion matrix, flooded compartments; $t^* = 500$ s; training: MC20; validation: MC50b; method: KNNW

True Class	Predicted Class (%)												
	1	**2**	**3**	**4**	**5**	**1,2**	**2,3**	**3,4**	**4,5**	**1,2,3**	**2,3,4**	**3,4,5**	**all**
1	17.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.01	13.38	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.02
3	0.00	0.00	13.61	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	13.44	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.01	0.03	16.49	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1,2	0.00	0.00	0.00	0.00	0.00	1.57	0.00	0.00	0.00	0.00	0.00	0.00	0.23
2,3	0.00	0.11	0.06	0.00	0.00	0.00	5.60	0.00	0.00	0.00	0.00	0.00	0.01
3,4	0.00	0.02	0.05	0.11	0.00	0.00	0.01	5.65	0.17	0.00	0.00	0.00	0.08
4,5	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.03	4.60	0.00	0.00	0.01	0.16
1,2,3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.02
2,3,4	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.03	0.00	0.00	0.07	0.00	0.00
3,4,5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.03	0.00	0.00	0.01	0.01
all	0.03	0.04	0.01	0.01	0.05	0.32	0.01	0.20	0.39	0.01	0.01	0.00	6.10

Table C.41 Confusion matrix, flooded compartments; $t^* = 500$ s; training: MC20; validation: MC50b; method: SVM1

True	Predicted Class (%)												
Class	**1**	**2**	**3**	**4**	**5**	**1,2**	**2,3**	**3,4**	**4,5**	**1,2,3**	**2,3,4**	**3,4,5**	**all**
1	16.93	0.00	0.00	0.11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.03	13.32	0.01	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.05
3	0.00	0.00	13.62	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.05	13.38	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.04	0.00	0.01	0.28	16.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1,2	0.01	0.00	0.00	0.00	0.00	1.51	0.00	0.00	0.00	0.00	0.00	0.00	0.29
2,3	0.01	0.10	0.00	0.03	0.00	0.00	5.58	0.00	0.00	0.00	0.01	0.00	0.06
3,4	0.00	0.00	0.04	0.04	0.01	0.00	0.00	5.89	0.03	0.00	0.00	0.00	0.10
4,5	0.02	0.00	0.00	0.01	0.00	0.00	0.00	0.02	4.75	0.00	0.00	0.00	0.05
1,2,3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04
2,3,4	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.01	0.00	0.00	0.00	0.00	0.08
3,4,5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.03	0.00	0.00	0.00	0.00
all	0.03	0.07	0.00	0.03	0.03	0.41	0.02	0.28	0.54	0.00	0.00	0.00	5.76

Table C.42 Confusion matrix, flooded compartments; $t^* = 500$ s; training: MC20; validation: MC50b; method: SVM2

True	Predicted Class (%)												
Class	**1**	**2**	**3**	**4**	**5**	**1,2**	**2,3**	**3,4**	**4,5**	**1,2,3**	**2,3,4**	**3,4,5**	**all**
1	16.94	0.00	0.00	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.02	13.34	0.00	0.01	0.00	0.00	0.02	0.00	0.00	0.00	0.01	0.00	0.02
3	0.00	0.00	13.62	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.01	13.42	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00
5	0.02	0.01	0.00	0.11	16.38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1,2	0.00	0.00	0.00	0.00	0.00	1.51	0.00	0.00	0.00	0.00	0.00	0.00	0.30
2,3	0.00	0.06	0.00	0.01	0.00	0.00	5.66	0.00	0.00	0.00	0.01	0.00	0.03
3,4	0.00	0.01	0.02	0.05	0.00	0.00	0.01	5.97	0.04	0.00	0.00	0.00	0.01
4,5	0.01	0.00	0.00	0.01	0.00	0.00	0.00	0.02	4.73	0.00	0.00	0.00	0.07
1,2,3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.02
2,3,4	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.03	0.00	0.00	0.06	0.00	0.01
3,4,5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.03	0.00	0.00	0.00	0.01
all	0.01	0.03	0.00	0.03	0.02	0.30	0.01	0.25	0.40	0.02	0.01	0.00	6.09

Table C.43 Confusion matrix, flooded compartments; $t^* = 500$ s; training: MC20; validation: MC50b; method: SVM3

True	Predicted Class (%)												
Class	**1**	**2**	**3**	**4**	**5**	**1,2**	**2,3**	**3,4**	**4,5**	**1,2,3**	**2,3,4**	**3,4,5**	**all**
1	16.95	0.00	0.03	0.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.01	13.37	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01
3	0.00	0.00	13.62	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.01	13.40	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.00
5	0.01	0.00	0.00	0.08	16.43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1,2	0.00	0.00	0.00	0.00	0.00	1.55	0.00	0.00	0.00	0.00	0.00	0.00	0.26
2,3	0.01	0.06	0.01	0.01	0.00	0.00	5.63	0.00	0.00	0.00	0.00	0.00	0.07
3,4	0.00	0.02	0.01	0.04	0.00	0.00	0.01	5.96	0.00	0.00	0.00	0.01	0.06
4,5	0.01	0.01	0.00	0.01	0.01	0.00	0.00	0.17	0.21	0.00	0.00	0.00	4.43
1,2,3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.02
2,3,4	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.07	0.00	0.02
3,4,5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.04
all	0.01	0.04	0.00	0.02	0.02	0.30	0.01	0.22	0.16	0.04	0.01	0.00	6.34

Table C.44 Confusion matrix, flooded compartments; $t^* = 500$ s; training: MC20; validation: MC50b; method: SVMG

True Class	Predicted Class (%)												
	1	**2**	**3**	**4**	**5**	**1,2**	**2,3**	**3,4**	**4,5**	**1,2,3**	**2,3,4**	**3,4,5**	**all**
1	16.93	0.00	0.00	0.07	0.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.02	13.33	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05
3	0.00	0.00	13.62	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	13.43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.04	0.00	0.40	16.09	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1,2	0.00	0.00	0.00	0.00	0.00	1.59	0.00	0.00	0.00	0.00	0.00	0.00	0.22
2,3	0.00	0.11	0.00	0.01	0.01	0.00	5.62	0.00	0.00	0.00	0.00	0.00	0.04
3,4	0.00	0.01	0.03	0.05	0.01	0.00	0.00	5.89	0.08	0.00	0.00	0.00	0.05
4,5	0.00	0.02	0.00	0.01	0.00	0.00	0.00	0.00	4.73	0.00	0.00	0.00	0.09
1,2,3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04
2,3,4	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.02	0.00	0.00	0.03	0.00	0.04
3,4,5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.04	0.00	0.00	0.00	0.00
all	0.00	0.07	0.00	0.03	0.02	0.43	0.01	0.26	0.45	0.00	0.00	0.00	5.91

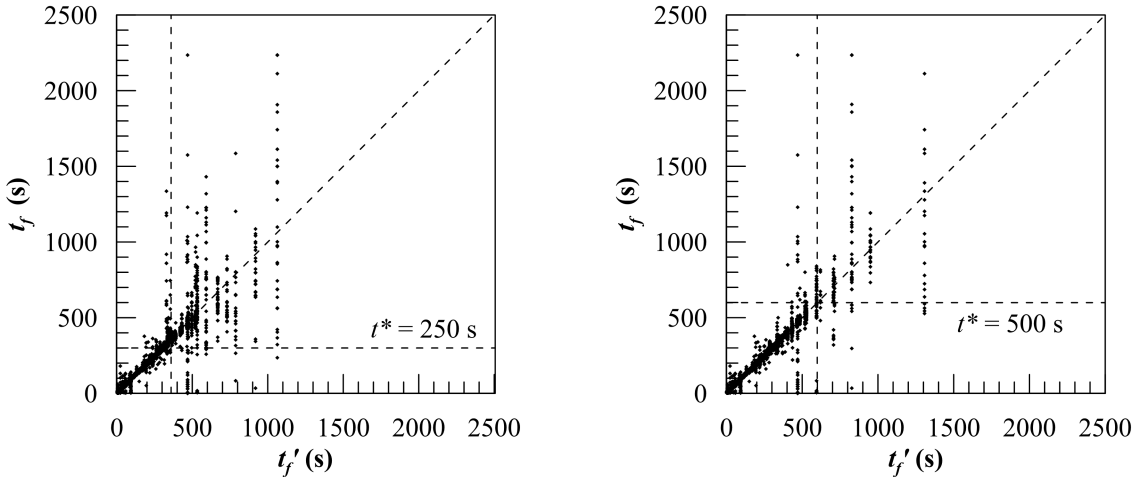

Figure C.1 Predicted over obseved values, t_f; $t^* = 250$ s and $t^* = 500$ s; training: MC20; validation: MC50b; method: DT

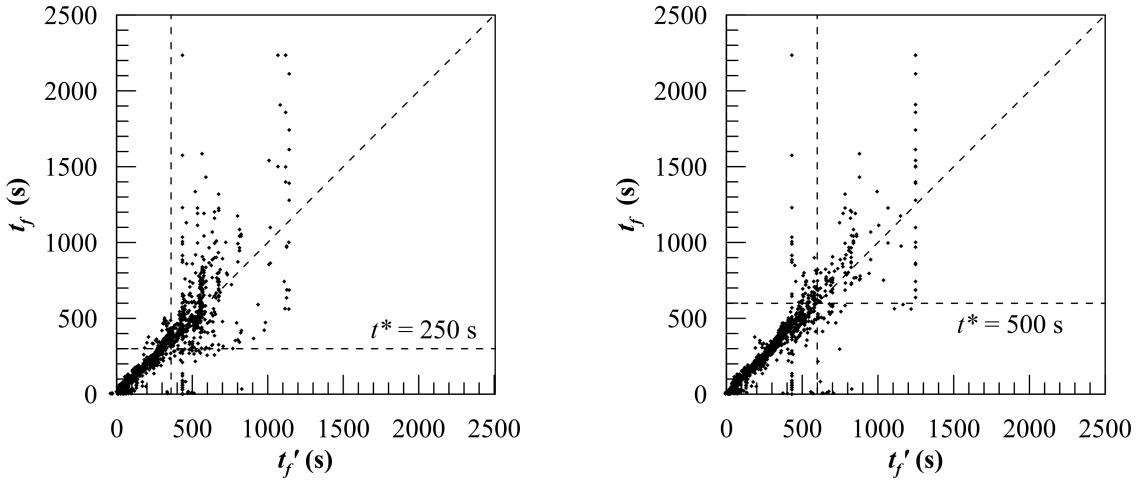

Figure C.2 Predicted over obseved values, t_f; $t^* = 250$ s and $t^* = 500$ s; training: MC20; validation: MC50b; method: DTB

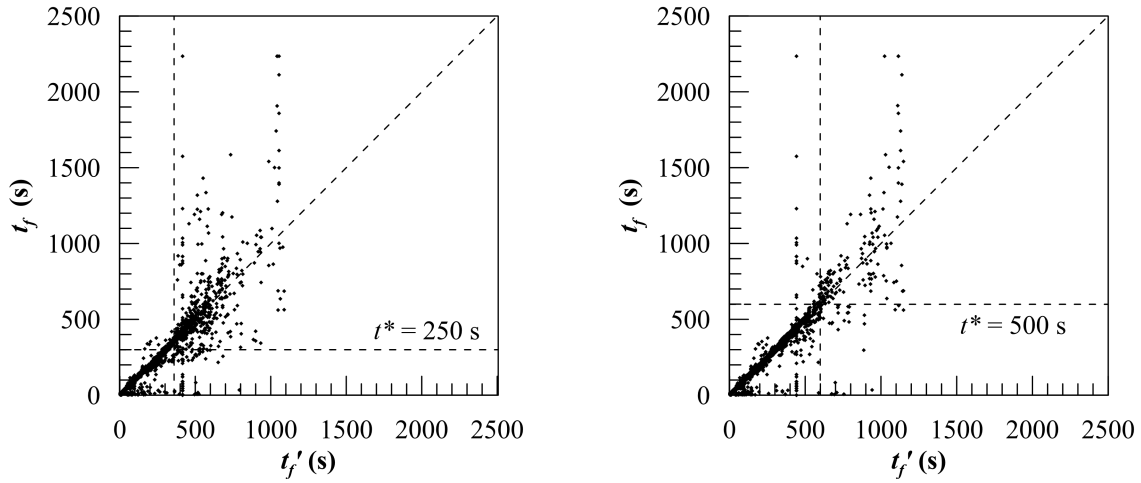

Figure C.3 Predicted over obseved values, t_f; $t^* = 250$ s and $t^* = 500$ s; training: MC20; validation: MC50b; method: RF

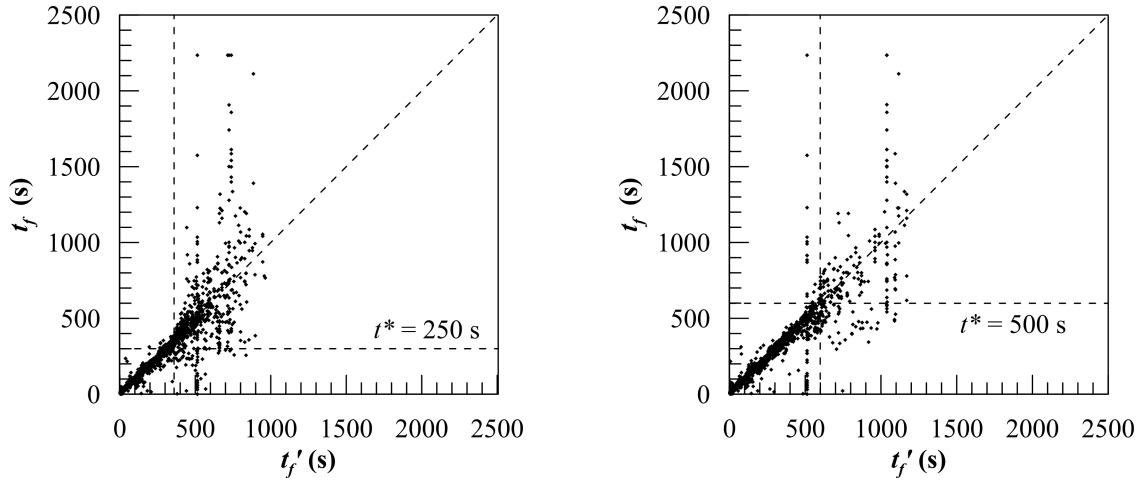

Figure C.4 Predicted over obseved values, t_f; $t^* = 250$ s and $t^* = 500$ s; training: MC20; validation: MC50b; method: KNN2

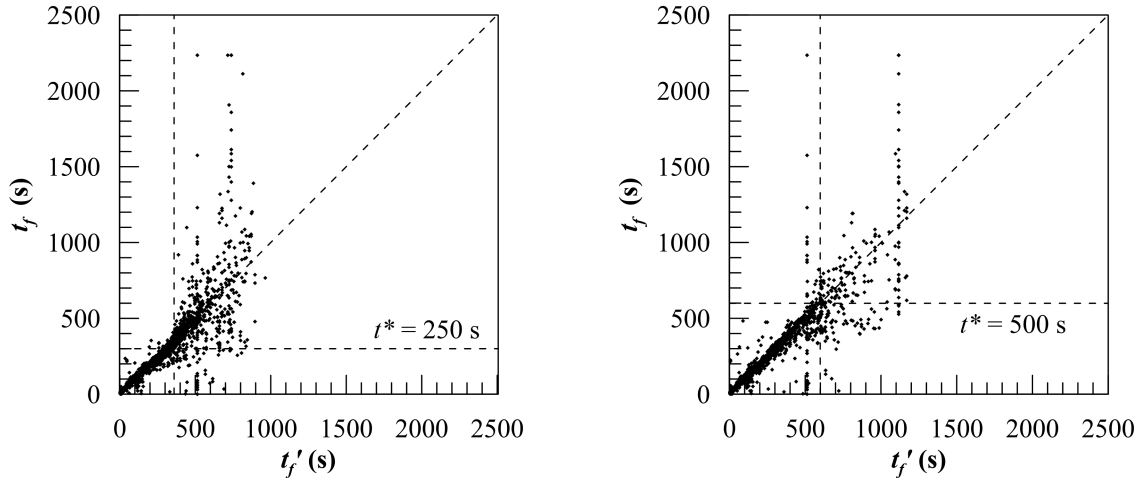

Figure C.5 Predicted over obseved values, t_f; $t^* = 250$ s and $t^* = 500$ s; training: MC20; validation: MC50b; method: KNN3

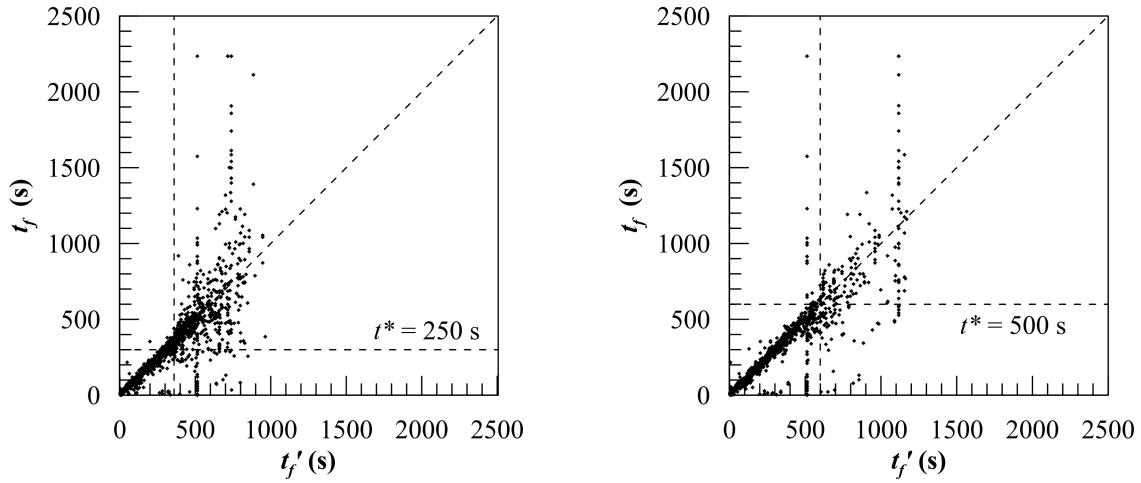

Figure C.6 Predicted over obseved values, t_f; $t^* = 250$ s and $t^* = 500$ s; training: MC20; validation: MC50b; method: KNNC

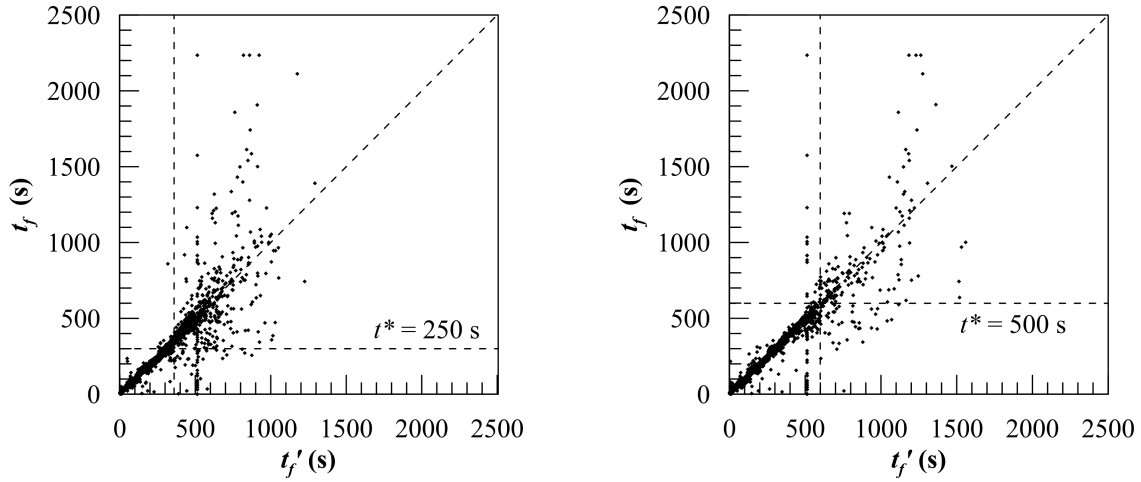

Figure C.7 Predicted over obseved values, t_f; $t^* = 250$ s and $t^* = 500$ s; training: MC20; validation: MC50b; method: KNN2

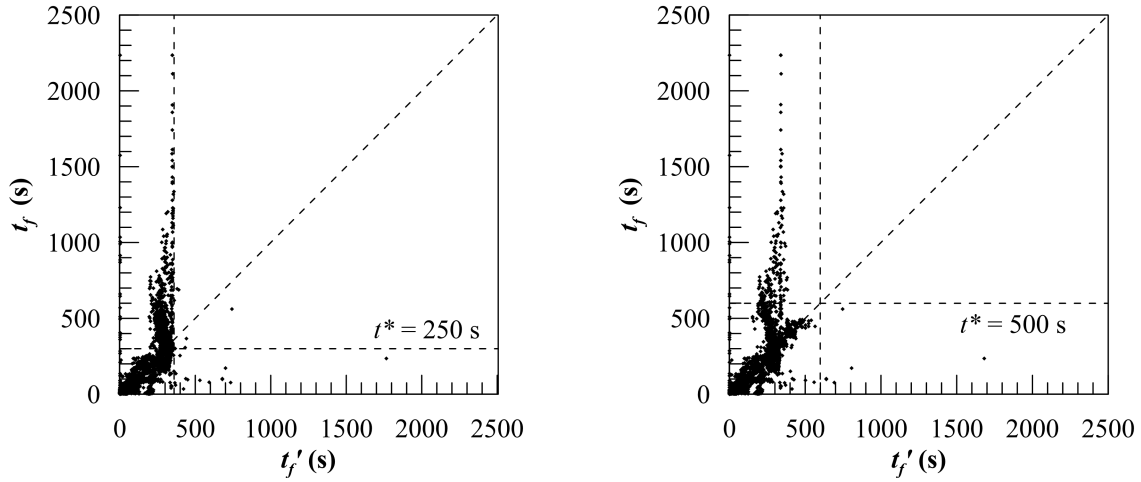

Figure C.8 Predicted over obseved values, t_f; $t^* = 250$ s and $t^* = 500$ s; training: MC20; validation: MC50b; method: SVM1

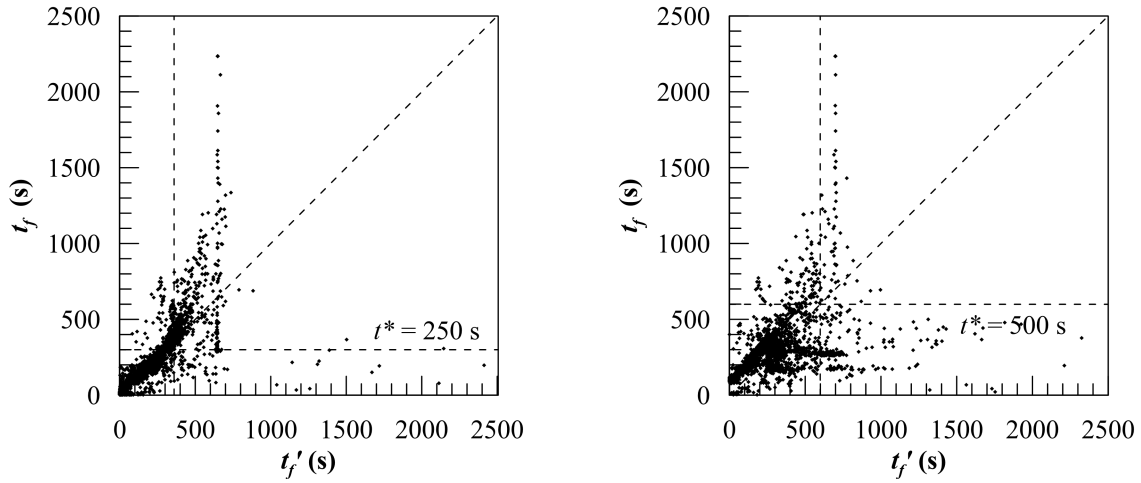

Figure C.9 Predicted over obseved values, t_f; $t^* = 250$ s and $t^* = 500$ s; training: MC20; validation: MC50b; method: SVM2

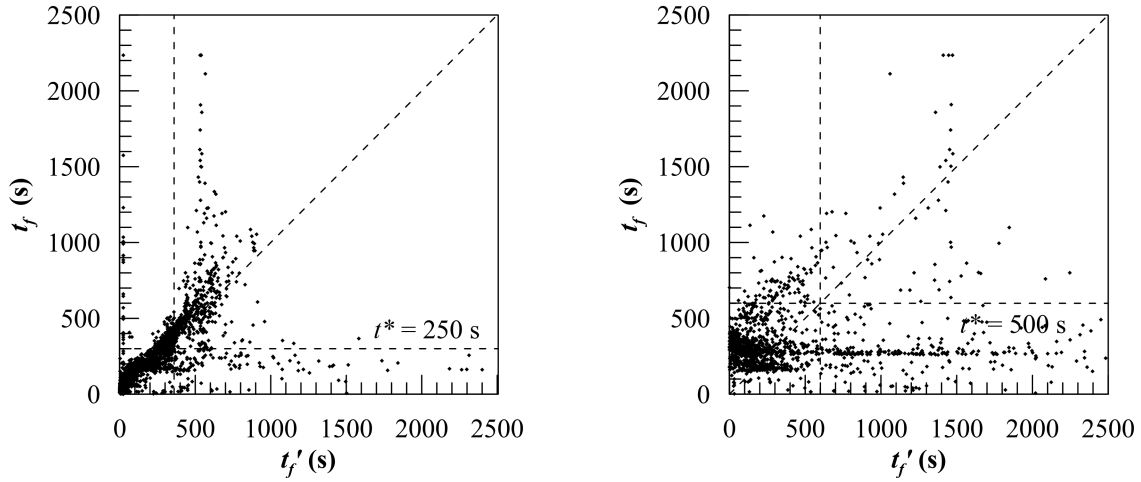

Figure C.10 Predicted over obseved values, t_f; $t^* = 250$ s and $t^* = 500$ s; training: MC20; validation: MC50b; method: SVM3

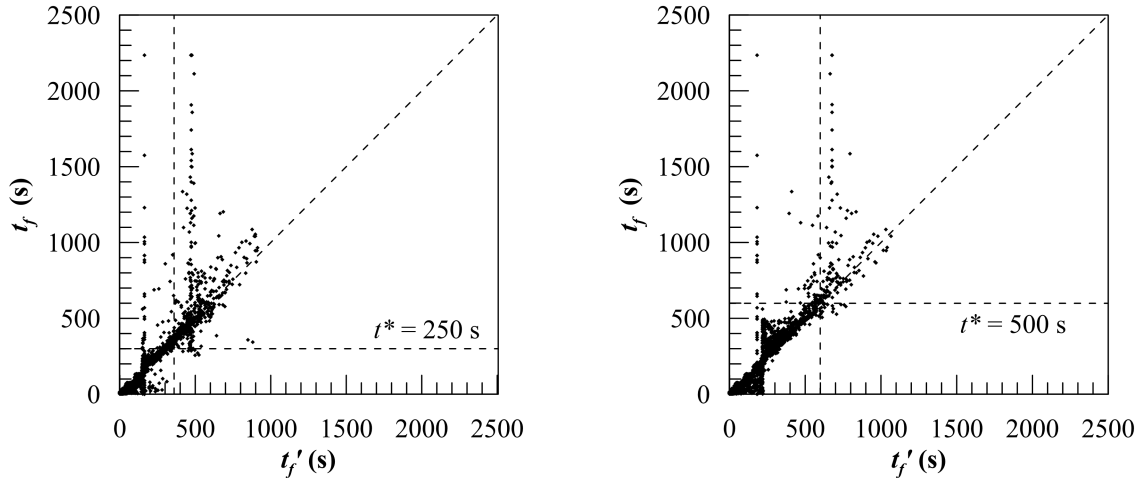

Figure C.11 Predicted over obseved values, t_f; $t^* = 250$ s and $t^* = 500$ s; training: MC20; validation: MC50b; method: SVMG

Author's short biography

Luca Braidotti was born in Udine (Italy) on July 3rd 1990. He received the High School Diploma in 2009 from the Liceo Scientifico Statale G. Marinelli (Udine) with the highest honours. Then, he enrolled in the course of Naval Architecture and Marine Engineering at the University of Trieste, where he received the Bachelor and Master Degree with the highest honours in 2012 and 2014, respectively. Later, he worked as a consultant and, in 2016, he founded the company Navium S.r.l.s. where he acted as managing partner and was responsible for R&D and software development. In this period his work covered multiple fields including ship stability, emergency decision support, progressive flooding simulation, onboard risk evaluation in collaboration with primary actors in the Italian maritime sector, including Seastema S.p.a. (Fincantieri company). From 2017 up to now, Luca Braidotti is a research fellow at the University of Trieste within the Integrated Ship Design Laboratory where his research is focused on marine decision support systems, the development of metamodels for ship concept design, the development of new vessel designs. From 2017, he is a PhD student at the University of Rijeka, Faculty of Engineering. During the doctoral study, his research was focused on fast progressive flooding simulation tools and flooding-sensor-agnostic decision support systems based on machine learning. From 2020 he is a research assistant at the Venice International University within the TeDIS program where his research is mainly focused on transport sustainability. From 2020 Luca Braidotti is also appointed Assistant Professor at the University of Trieste, holding the course "Laboratory of Naval Architecture and Ship Technology", part of the Bachelor Degree in Naval Architecture and Marine Engineering. Luca Braidotti is a member of the ATENA (Italian association of naval architects and marine engineers) since 2017, being a member of the board of the Friuli Venezia Giulia section from 2019. In the latest years, he was involved in several international conferences being member of the scientific committee of NAV 2018, CNM 2019, HSMV 2020 and of the organizing committee of NAV 2018, CNM 2019, HSMV 2020 and the forthcoming NAV 2022. Luca Braidotti is author of more than 25 publications in scientific journals or included in conference proceedings.