Securing harbor by combining probabilistic approach with event-based approach

Salma Zouaoui-Elloumi

Center for Applied Mathematics,Rue Claude Daunesse - BP 207 - 06904 - Sophia Antipolis France

Valérie Roy

Center for Applied Mathematics,Rue Claude Daunesse - BP 207 - 06904 - Sophia Antipolis France

Nadia Maïzi

Center for Applied Mathematics,Rue Claude Daunesse - BP 207 - 06904 - Sophia Antipolis France

Abstract

As a part of SECuring MARitime (SECMAR) project that aimed to improve security at Marseilles harbor, we developed a decision support system that helps port staff to better monitor ship behavior. It consists of two complementary sub-systems. The first system is based on the probabilistic Hidden Markov Model (HMM) approach and deals with nominal (regular and recurrent) behavior of large to medium size of commercial ships. The second system is based on the reactive synchronous language Esterel and concerns aggressive and transgressive behavior of small ships that may navigate freely in the harbor. Real-time evaluations showed that the proposed decision support system efficiently captured and evaluated ship behavior.

Keywords: Hidden Markov Models, Reactive synchronous language, decision support, Harbor security.

1. Introduction

Nowadays, overseas transport, in addition to its touristic aspect, plays a major role in international trade since 90% of raw materials such as oil, gas, iron ore and coal are exchanged over the sea. As a consequence, it is worth paying a particular attention to the maritime space, which can be exposed to significant risks, such as accidents (for example, the recent sinking of the cruise ship Costa Concordia in 2012 in Italy) and terrorist attacks (such as the attack of the American ship USS Cole in 2000). For these reasons, several measures and security systems were introduced by European and American countries:

- The Customs-Trade Partnership Against Terrorism (CTPAT): a government business-program established by the U.S in 2001 that aims to strengthen the overall supply chain and U.S. border security.
- The Seveso directive of 1996: a European law aimed at identifying and preventing industrial sites with high level of risk (containing dangerous substances).
- The Vigipirate system: a national security alert system created in 1978 by the French government that aims to prevent from threats and terrorism.

Email addresses: salma.zouaoui@gmail.com (Salma Zouaoui-Elloumi), valerie.roy@mines-paristech.fr (Valérie Roy), nadia.maizi@mines-paristech.fr (Nadia Maïzi)

- The Automated Identification System (AIS): an automatic tracking system imposed by the International Maritime Organization (IMO) to all the ships of more than 300 tons. The military and small ships don't have this obligation. The AIS is formed by an on-ship transmitter that sends the ship realtime information (type, position, speed, etc) to a receiver located in the harbor.

However, in spite of the above measures to improve security in the maritime space especially after the events of September 11th, harbors remained unsecure for many reasons:

- First, the AIS, which is considered as the best ship tracking system used by all countries, is unable to track all ships in the sea. In fact, many small size ships such as jet skis or sailboats are not equipped with this system. Moreover, the AIS can be deliberately stopped or its data can be modified.
- Secondly, several aspects of maritime surveillance such as underwater surveillance or video surveillance have hardly ever been established in a harbor.
- Finally, monitoring harbor currently performed by a security guard is a tedious task since the guard must often monitor several vessels at the same time. Besides he needs to have the ability to integrate simultaneously several data sources from the various sensors and from the harbor environment.

In this context, SECMAR project (Figure 1) aimed at developing a decision support system for the security team of Marseilles' harbor. It comes in three folds:

1. An advanced detection system based on new adjusted *sensors*, namely *radar*, *sonar* and *optronic cameras*. This system is installed on the harbor to avoid any dependency on the ships equipment.
2. A data fusion system that merges all the sensors data to the so called "track system" which is a set of exploitable data such as speed and position.
3. A behavior analysis modulus which is able to transform the track system into a valuable decision support information. This last point is the focus of this current paper.

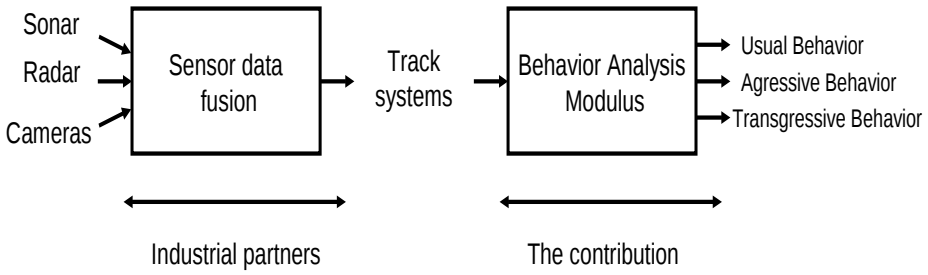


Figure 1: Steps of SECMAR project

The industrial partner of SECMAR project, Thales Underwater System, has developped an innovative algorithm to merge sensor data recorded by radar, sonar and cameras in order to obtain the track systems. A track system is a single information about every vessel in the harbor at each time step. It is composed of the following information: position in longitude and attitude, speed, heading, time and type (small or commercial ship). To conceive the Behavior Analysis Modulus (BAM), ships' positions (meter), speeds(m/s), times(seconds) and types are used as input of the system. We have started by observing the ships trajectories (Figure 2 represents some ships trajectories in Marseilles harbor) reconstructed from the track systems and their motion through the well-marked areas

established by the legislation (for example area A, area B, area Mooring N in Figure 2). Thus, we have established two kinds of ship behavior:

- Commercial ships such as oil tankers or container ships displayed regular, recurrent behavior. They have a well-established route that they have to follow to reach their corresponding platform (Figure 2 - (a)).

- Small boats, like sailboats and jet skis moved freely in the harbor without any restriction (Figure 2- (b)).

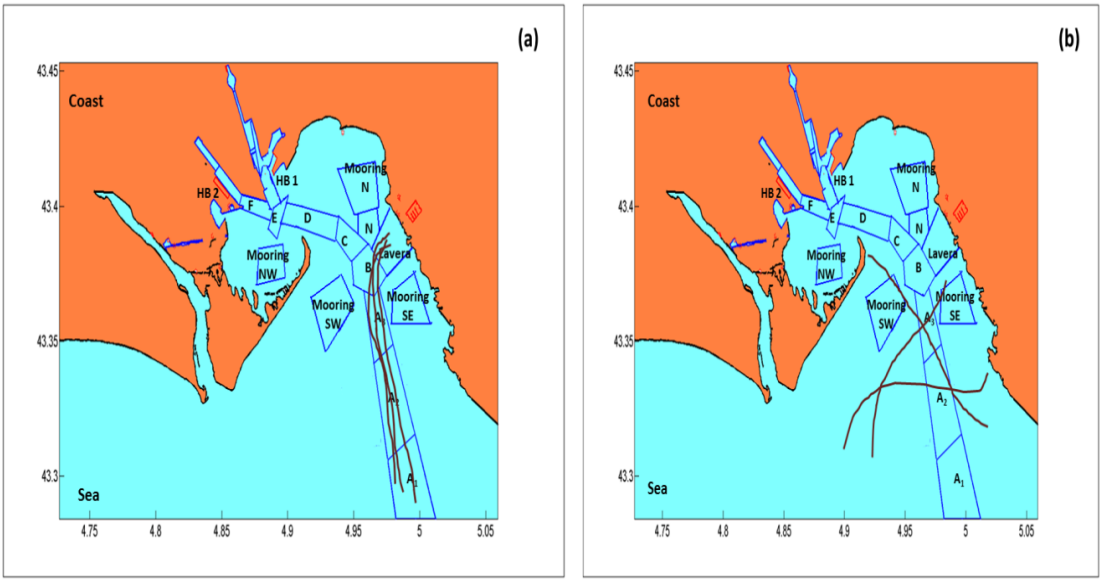


Figure 2: Examples of trajectories of commercial ships (a) and small boats (b)

In this paper, we propose a Behavior Analysis Modulus (BAM) formed by two parallel and complementary sub-systems (Figure 3) that tackle this duality of the observed ships:

- The first system is based on the probabilistic Hidden Markov Model (HMM) approach and deals with nominal behavior of commercial ships showing regular and recurrent behavior.

- The second system is based on an event-based approach with the Esterel synchronous reactive language and concerns mainly the aggressive and transgressive behavior of small ships.

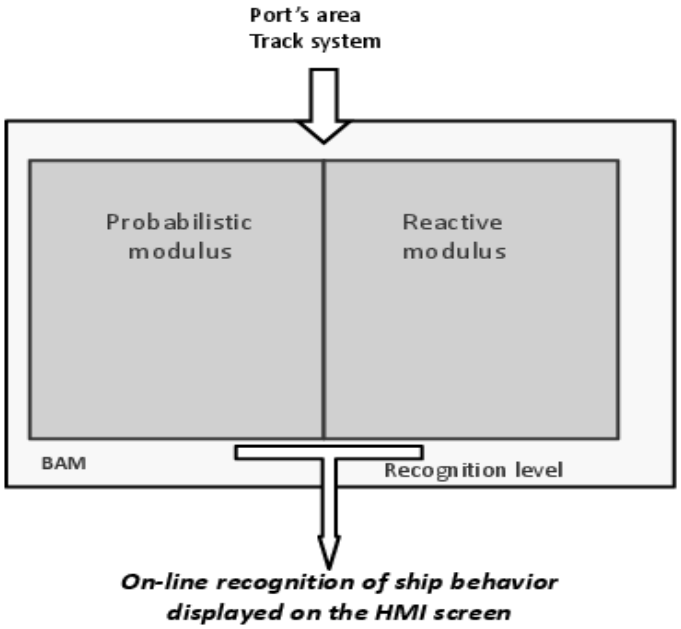


Figure 3: Behavior Analysis Modulus (BAM)

We will start, in the next section, by presenting some related works in the field of behavior recognition. We introduce in Section 3 an overview of the proposed approach. The used approaches for the BAM conception are presented in Section 4: the HMM is presented in Section 4.1, then the synchronous reactive approach and the Esterel language are introduced in Section 4.2. We describe in Section 5 the Behavior Analysis Modulus for ship behavior recognition. This system is formed by two systems: the probabilistic modulus that we introduce at first in Section 5.1 and the reactive modulus that we present thereafter in Section 5.2. The parallelism and the complementary of both systems are illustrated in Section 5.3 through the description of two examples of ship behavior: a nominal and a dangerous one. Finally, Section 6 ends with a comparison and analysis of both systems.

2. Related works on Behavior recognition and analysis

Behavior recognition is an important field of Artificial Intelligence (AI). Techniques such as statistical or logical methods were widely used to conceive autonomous systems used for security or for decision support. We can classify works on behavior recognition and analysis into two categories: works based on probabilistic approaches and works based on event-based approaches. In this section, we will focus briefly and separately on both of them.

2.1. A probabilistic-based approach for behavior recognition

In robotics and according to [1] , the conception of an autonomous robot goes through modeling its interaction with its indoor or outdoor environment. For example, authors proposed in [2] to use the probabilistic approach Hidden Markov Models (HMM) to learn many human skills that were transferred to a space station robot system for teleoperation control. In the same manner, authors in [3] used HMM approach in order to develop many autonomous robots playing soccer. The behavior recognition was also used in order to integrate intelligent systems

in vehicles to driver support. In [4], the HMM approach was used in order to learn driver's behavior that through a model of "intention-action". Seeing that human behavior is based on unknown intentions, we were able to infer them from the driver actions. In this case, human behavior is well described by an HMM where model's variables were constructed from the hidden states (intentions) and the observations (actions). In [5] and [6], authors also used HMM in order to model many drivers' behavior since each driver has its own manner of driving. This idea proved to be a very good method in preventing against vehicle theft.

2.2. An event-based approach for behavior recognition

The event-based approach is a logical and a deterministic method. For example, in robotics, modeling robot's interaction with its environment goes through programs based on geometric terms, analytical expression and symbols [1]. In this context, authors in [7] used logical methods, programming languages and automaton theory in order to resolve plan and control problems associated to robots. Authors also proposed in [8] to model ants' behavior by robots programmed in an imperative language (Esterel).

The next section justifies the use of the two previsouly described approach for the conception of our framework.

3. Overview of the proposed approach

We propose a complete framework for recognizing usual and unusual ship behavior by combining a probabilistic HMM approach and a reactive synchronous system developped in Esterel language. By analyzing ships' behavior in the harbor from a previously recorded track systems, we noted that commercial ships' trajectories are recurrent and regular since they are respecting maritime rules. Thus, the systems can constitute a set of data that can be handled in a statistical way. We have chosen to use the track systems as well as the well-marked areas to construct and learn, in a probabilistic manner, the set of all models of usual ship behavior in the harbor. This set of models enables to recognize on-line the behavior of every new ship entering the harbor and to distinguish between usual and unusual ship's behavior. Since the HMM is a well-known approach for recognition issues (see Section 4.1), we have pitched on this approach. However, in the case of small ships moving freely in the harbor, the previously described representation of ship behavior is inappropriate. Indeed, the non-regular ships behavior makes it difficult to ascertain their behavior using a probabilistic approach. For this reason, we have decided to analyze and interpret the track systems and the well-marked areas in order to obtain significant events. For example, the increasing speed of a vessel is represented by the event "the ship is accelerating". That is, a physical information (track system and well-marked area) is transformed to a logical one described by an event. An event that we called *fact* is computed at each time step. The constructed sequences of facts received during the time enable the description of the ongoing behavior. The set of facts is manipulated by a set of instructions in order to construct a behavior that we called *scenario*. This way, we obtain several types of scenarios and each scenario is a reactive system that instantaneously sends out a recognition level signal indicating the current type of the ship behavior. To develop the described scenarios, we selected the Esterel synchronous reactive language that is well-known for its easy modular description of reactive systems. Moreover, in comparaison to other synchronous reactive languages (see Section 4.2.1) Esterel is a simple language offering an extensive set of tools. It also produces executable programs that are small and efficient.

4. Used approaches: HMM and reactive synchronous language

4.1. Hidden Markov Model

During the past decade, the Hidden Markov Model (HMM) has been considered as the most popular probabilistic approach in the field of speech recognition [9, 10]. The use of HMM has also been extended to many other

applications, such as face recognition [11] and handwriting recognition [12]. This approach has also performed well in modeling human behavior. Many authors in [13],[14] and [5] used HMM to analyze and recognize driver behavior patterns. For example, in [6], authors proposed recognizing drivers' behavior based on their driving performances. Starting from a set of information data associated with the driver (acceleration, braking and steering value), seven individual HMM models for seven drivers were trained by using the iterative Baum-Welch algorithm. These models were used as a classifier to identify the drivers from a new set of observation sequences.

4.1.1. Definition of HMM

A Hidden Markov Model is an extension of the standard Markov model [10], in which the system to be modeled depends on unobservable (hidden) states. Hidden states as well as states and observations are connected to each other. A model of HMM λ is defined as follow $\lambda = (\pi, A, B)$ where:

- $A = \{a_{ij}\}_{1 \leq i,j \leq N}$ is the transition matrix that illustrates connections between states where N is the number of states and $a_{ij} = P(X_{t+1} = S_j / X_t = S_i)$ is the transition probability from state i to state j. Note that the states of the HMM model are always discrete.
- $B = \{b_j(O)\}_{1 \leq j \leq N}$ is the output probability distribution where $b_j(O)$ is the random function associated with state j which is in general a mixture of Gaussian distributions.
- $\pi = \{\pi_i\}_{1 \leq i \leq N}$ is the initial state distribution and $\pi_i = P(X_1 = S_i)$.

Given the definition of HMM, there are three basic problems of interest that must be solved for real world applications: the evaluation problem, the decoding problem, and the learning problem. The solutions to these three problems are the Forward-Backward algorithm, the Viterbi algorithm, and the Baum-Welch algorithm [10].

4.2. Synchronous reactive approach

4.2.1. Reactive appraoch

The term "reactive system" was introduced in [15] to characterize systems that permanently interacts with their environment (e.g. the program controlling a lift or the undercarriage of an airplane, etc.). Reactive systems' behavior is equivalent to that of finite state machines (or FSM). Their execution is very time efficient (by state transition). They allow tools such as model checkers [16] to be used to formally verify their behavior. In aiming for expression, a language for designing reactive systems must support: modularity, communication and concurrency between modules, and preemption. Moreover, behavior must be driven by its environment and must be deterministic (i.e. if you give a reactive system exactly the same sequence of information, it must react the same way). The synchronous reactive approach provides a neat solution to deal with reactive systems: combining concurrency and determinism.

4.2.2. Synchronous approach

Every time it is run, a synchronous reactive program reacts instantaneously to a set of external events (its input signals) by emitting a set of events towards its environment (its output signals) and is immediately ready to react again. This instantaneous run is called a reaction. To meet the instantaneous requirement, restrictions have been applied to the language: there is no memory allocation during the execution (all needed space is statically allocated); manipulation of data is reduced to simple operations on integral data types such as comparison, subtraction, addition, etc; the implementation of operators such as sequence, signal status testing, signal emission etc. takes no time.

The run of a reactive synchronous system is an infinite loop of reactions. To meet the instantaneous hypothesis, the reactive synchronous approach makes some assumptions presented in [17].

Synchronous reactive languages have been designed in the nineties to specify critical reactive sub-parts of systems: they offer high-level and expressive mathematical formalisms. The best-know reactive languages are: the imperative control-dominated language Esterel [18] that has been designed by our research center, the declarative dataflow language Lustre [19] and the declarative multi-flow language Signal [20].
The choice of language depends on the application. The reactive languages previously referred were used intensively in industry to program the reactive sub-systems of major systems in aeronautics, car industry, space, telecommunications, etc. We have chosen to specify the behaviors of our scenarios in the Esterel language. In the next section we will introduce the main operators of this language.

4.2.3. Set of operators of a synchronous reactive language: ESTEREL

Users of programming language such as Esterel need a complete and accurate understanding of the intended meaning of the language (its semantics [21]) as well as its form [18]. We will briefly present here the main operators as well as their syntax in Esterel which is a synchronous reactive language.

- p1; p2

In a sequence of two statements, the execution of the second statement begins at the instant the first one ends;

- p1// p2

The two statements p1 and p2 are executed concurrently. The parallel finishes as soon as the two statements are completely finished whatever the order of their termination;

- *await S*

As soon as it is executed, whatever the status of S, this statement blocks. At the following instants, as soon as the status of S is present, the statement ends, otherwise it stays blocked;

- *emit S*

This statement instantaneously changes the status of the S signal to present and ends; present S then p1 else p2 The first instant this statement is executed, if the status of S is present, program p1 starts to be executed, otherwise program p2 does;

- *weak/strong abort p when S*

The instant this statement is executed, the statement p starts to be executed and if p ends instantaneously, so does the abort statement. During the following instants, if the status of S is absent, the execution of the statement p continues, otherwise the abort statement ends: we say that p is preempted by S. In case of a strong abortion, the instant signal S is present, program p is not executed at all, in case of a weak abortion, the program p ends its current instant.

5. Behavior Analysis Modulus for ship behavior recognition

In this section, we will explain how we used the described approaches of Section 4 (HMM and reactive language) in order to conceive a system for ships' behavior recognition. The overall system is formed of two parallel subsystems. The first one is a probabilistic modulus that supervises efficiently the nominal behavior of ships. The second one is a reactive modulus whose first mission is to detect abnormal ship's behavior. Before introducing the operating mode of the overall modulus, we will start in the next sub-sections (Section 5.1 and Section 5.2) by separately presenting and validating each one of them.

5.1. Probabilistic modulus

We modeled a Nominal ship behavior λ by an HMM whose parameters π, A et B need to be initially identified. After studying maritime rules and observing all possible ship trajectories in the harbor, we determined a list of all possible and nominal ships' behavior.

Commercial ships have to respect legally established maritime rules by taking one of the corresponding fairways of Figure 5-(a) to reach their corresponding target (i.e. oil tankers must head towards Harbor Bassin 1, etc). Here, is a list of 7 possible and acceptable ship behavior (nominal behavior) set down in the legislation:

- *Ships in the fairway heading towards South East Mooring Area (Mooring SE).*
- *Ships in the fairway heading towards South West Mooring Area (Mooring SW).*
- *Ships in the fairway heading towards Lavera.*
- *Ships in the fairway heading towards North Mooring Area (Mooring N).*
- *Ships in the fairway heading towards North West Mooring Area (Mooring NW).*
- *Ships in the fairway heading towards harbor bassin 1 (HB1).*
- *Ships in the fairway heading towards harbor bassin 2 (HB2).*

The Probabilistic modulus recognizes every ship behavior using probabilistic computing and classifies it into common or uncommon. Any ship behavior pattern that does not come under the list above will be considered as unusual.

The idea is to construct the set Λ of all HMM models $\Lambda = (\lambda_1, ..., \lambda_n)$ associated to all ship behavior defined in the legislation where λ_i, $1 \leq i \leq n$, represents the model that we have obtained after initialization and optimization phase and n (n=7) is the number of all behavior patterns. Now, let us consider that a new ship is entering the port. At each time slice t, we extract from the recorded track system the ships' positions that we used as observation sequences $O = (O_1, ..., O_T)$. Ships' positions are initially recorded in geographic coordinates and then converted into Cartisian coordinates.

The recognition modulus will try to classify and recognize the behavior of this ship by computing at every t:
$\{P(O_1...O_t/\lambda_i\}_{1\leq i\leq n}$
Then, the appropriate behavior of the ship will be obtained by computing at each time-slice:
$max_{\lambda_i}\{P(O_1...O_t/\lambda_i\}_{1\leq i\leq n}$
In practice, the probabilistic modulus (Figure 4) is made up of three parts:

- Initialization phase, where the HMM models of all ship behavior defined in legislation were initialized (Figure 4 - (a));
- Learning (reestimation) phase where sensor data (track systems) as well as the initialized models were used to reestimate HMM models in order to obtain more accurate behavior models (Figure 4-(b));
- Recognition phase, where the behavior of every new ship entering the harbor will be recognized (Figure 4-(c)).

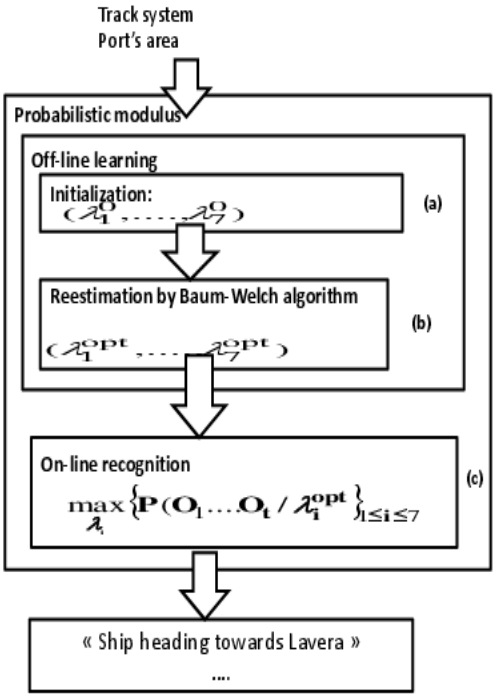


Figure 4: Probabilistic modulus

To initialize HMM models associated to ship behavior, our idea was to link each small area of the harbor (areas defined by the legislation) in Figure 5 - (a) to an ellipse of Figure 5 - (b) whose parameters would be introduced into the HMM models.

Thereafter, we used track system (ship trajectories represented from track system are represented in Figure 6)) to reestimate initial parameters of HMM Models associated to ship behavior.

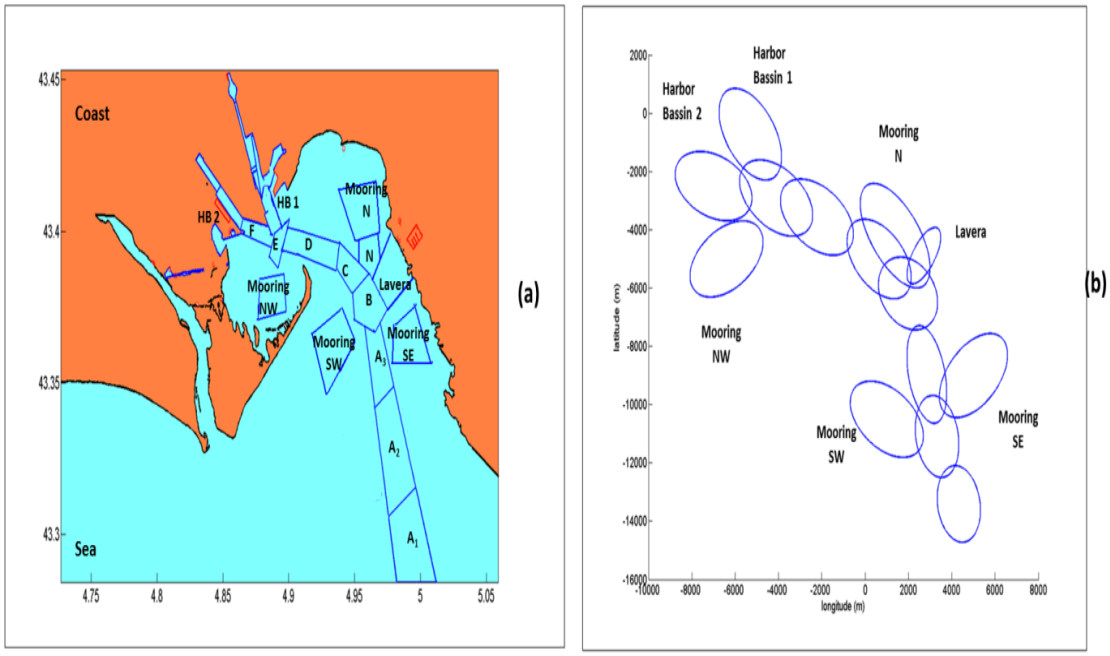


Figure 5: (a) areas constituting the access channels of ships to the harbor (b) ellipses representing areas of the harbor

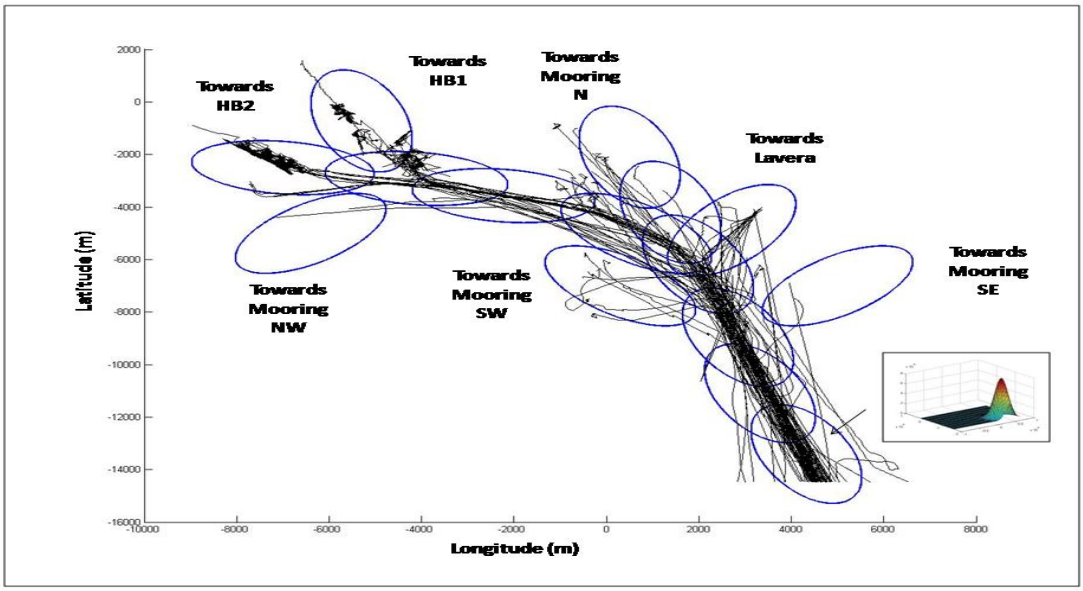


Figure 6: ships'trajectories given by track systems

5.1.1. Initial models

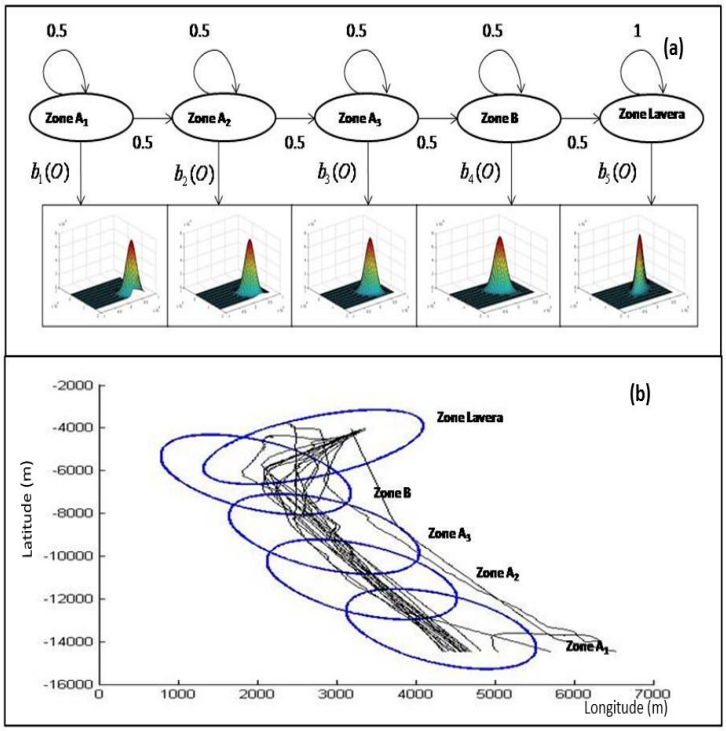


Figure 7: Behavior of an oil tanker heading towards Lavera. The ship goes through area A1, area A2, area A3, area B and Lavera

To introduce all steps of models initialization, we will take the example of the behavior of ships that are heading towards Lavera (Figure 7).

Our system will be performed with a left-right HMM with no-skip state transitions (Figure 7 – (a)). We assumed that only transition from state i to state i and from state i to state $i+1$ are allowed. This means that we are considering that for every time-slice, the ship stays in the ith area or goes to the $i+1th$ area. We will consider states' probabilities to be equiprobable so that the transition matrix A is initialized as follows:

- $\{a_{ij}\}_{1\leq i,j\leq N} = 0.5$ if $i = j$ or $j = i + 1$
- $a_{NN} = 1$

The output probability distribution $B = \{b_j(O)\}_{1\leq j\leq N}$ is a set of 2D Gaussian function where:

$$b_j(O) = \frac{1}{(2\pi)^{\frac{n}{2}}} exp\left\{-\frac{1}{2}(O-\mu_i)\Sigma_j^{-1}(O-\mu_i)^T\right\} \quad (1)$$

As described previously from Figure 5, each area is linked to an ellipse centered in $\mu_i = (x_i, y_i)$ where a_i and b_i are respectively the semi major axis and the semi minor axis of each ellipse. The covariance of the Gaussian function depends on a_i and b_i as follows:

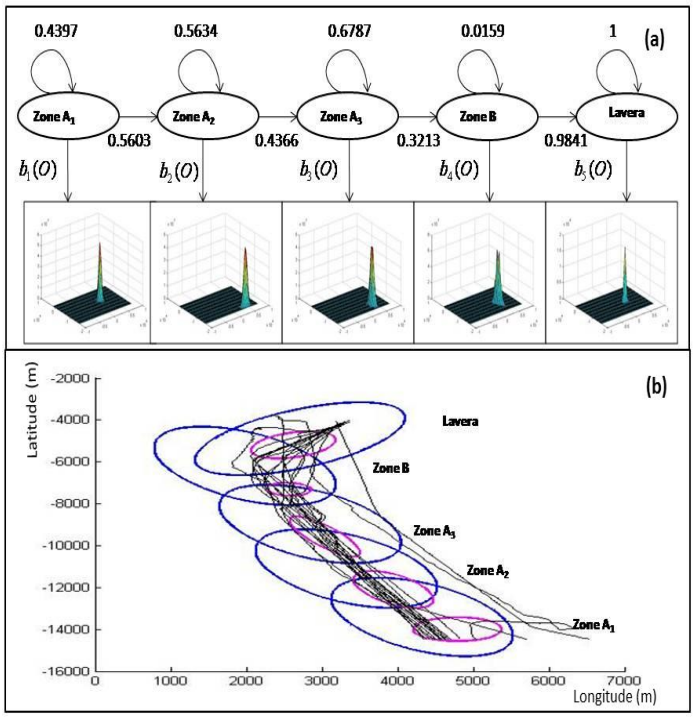


Figure 8: Obtained areas after reestimation procedure for behavior of Figure 7

$\Sigma_i = U_i A_i U_i^T$ where $A_i = \begin{pmatrix} a_i^2 & 0 \\ 0 & b_i^2 \end{pmatrix}$, $U_i = \begin{pmatrix} cos\,(\theta_i) & -sin\,(\theta_i) \\ sin\,(\theta_i) & cos\,(\theta_i) \end{pmatrix}$ and θ_i is the angle between the ellipse and the orthonormal basis. Ellipses are obtained from the covariance parameters of each output Gaussian probability (equation (1)).

5.1.2. Training models

All HMM models of all behavior patterns were initialized as described previously in Section 5.1.1. Then, sensor data (ship's positions) as well as the recursive Baum-Welch algorithm [10] were used to train and reestimate models. For this step, we extracted observation sequences from the set of 12 trajectories. More details on this phase are presented in [22].

After re-estimation procedure, except for the initial vector , all coefficients of A (state graph at (Figure 8 – (a)) and B are modified. The property of left-right HMM with no-skip states transition is conserved after re-estimation procedure and the matrix A is no longer represented by equiprobable values. The values μ_i and Σ_i and are also modified.

Figure 8– (b) also illustrates the changing means and covariance values. After the re-estimation procedure, the centers of ellipses $\mu_i = (x_i, y_i)$ previously chosen were adjusted with a set of observation sequences (ship positions). The result was that Gaussians were more defined and ellipse parameters (semi minor and semi major axis) were closer to the centers.

5.1.3. Recognition phase

The constructed models of ships behavior $\Lambda = (\lambda_1, ..., \lambda_7)$ will be used for the recognition phase by computing at each time-slice t, the probability of an observation (ship's position) given all models: $\left\{P\left(O_1...O_t/\lambda_i\right)_{1\leq i\leq 7}\right\}$

For example, we take the example of Figure 9 where we know formerly that the ship is heading towards Mooring North's area.

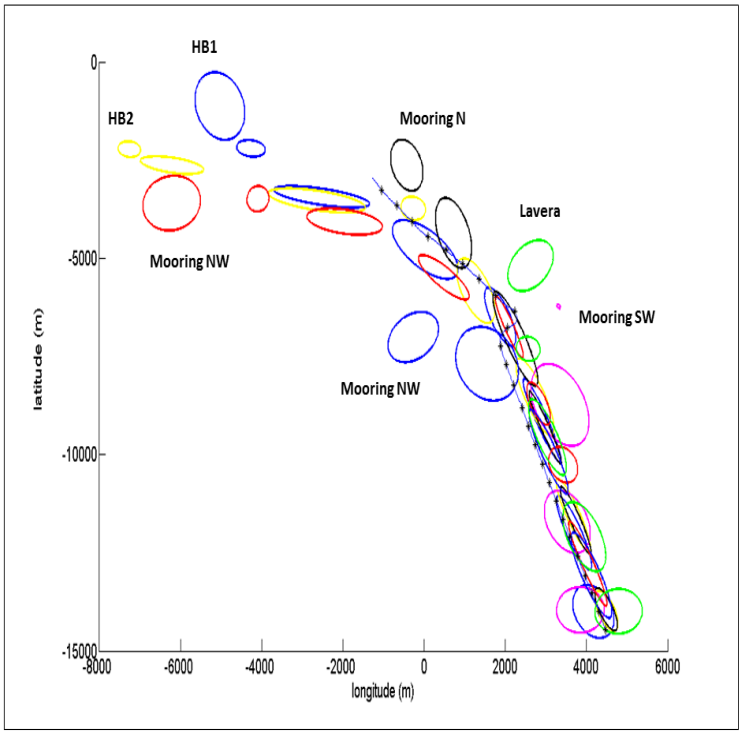


Figure 9: Example of ship's trajectory heading towards Mooring N

When the ship is entering the harbor and is moving through common areas, the probabilities' values $\left\{P\left(O_1...O_t/\lambda_i\right)_{1\leq i\leq 7}\right\}$ are very close to each other (see Table 1). Thus, it is not possible to know exactly which kind of behavior was performed.

The corresponding behavior is well defined when the ship starts reaching its corresponding area (Mooring North). This result is given by $P(O_1...O_t/\lambda_4) \geq \{P(O_1...O_t/\lambda_i)\}_{1\leq i\leq 7,\, i\neq 4}$ where λ_4 is the model of ship's behavior that is heading towards Mooring North (see Table 2).

We tried to recognize a set of 30 new ships entering the harbor. Among this set, only 5 trajectories were not properly classified. In this case, the recognition rate is 83%.

After defining and recognizing usual ship behavior, we were interested on unusual ship behavior. We can assume unusual ship behavior to be an unexpected behavior during a ship's movement. For example, a ship may suddenly change its direction, as shown in (Figure 10 - (a)), so that it heads toward Lavera after heading toward the Mooring area. Another example is when a ship is moving unawares outside its fairway in (Figure 10 - (b)). In such situations, a particular attention should be given by the guard.

Table 1: Probabilities values computed for ship entering the harbor (see Figure 9)

t	1	2	3
$P(O/\lambda_1)$	-15,4	-31,13	47,07
$P(O/\lambda_2)$	-14,84	-29,64	-44,1
$P(O/\lambda_3)$	-15,02	-31,04	-47,03
$P(O/\lambda_4)$	-14,31	-29,40	-45,07
$P(O/\lambda_5)$	-14,99	-30,02	-44,76
$P(O/\lambda_6)$	-14,75	-30,44	-46,07
$P(O/\lambda_7)$	-14,24	-29,19	-44,91

Table 2: Probabilities values computed for ship moving close to Mooring North's area (see Figure 9)

t	26	27	28
$P(O/\lambda_1)$	-974,59	-1147,46	-1345,37
$P(O/\lambda_2)$	-526,25	-798,56	-770,23
$P(O/\lambda_3)$	-1134,12	-1441,55	-1812,01
$P(O/\lambda_4)$	-406,88	-399,8	-424,1
$P(O/\lambda_5)$	-461,49	-533,01	-623,3
$P(O/\lambda_6)$	-378,15	-443,89	-489,5
$P(O/\lambda_7)$	-331,52	-500,77	-643,29

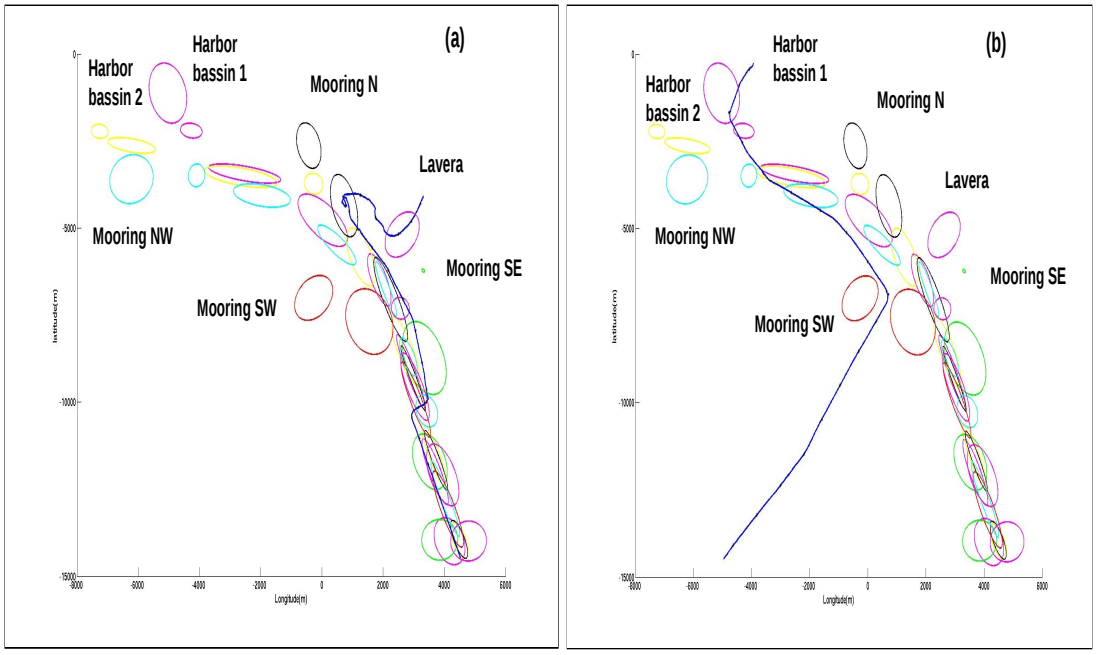


Figure 10: Examples of abnormal ship behavior

Nevertheless, a trajectory change detected by our probabilistic system is not enough to consider the ship's behavior as dangerous. However, if we know that boat change unawares its heading or its speeds (inappropriate acceleration), the ship's behavior can be considered as abnormal. In addition and as we can see from Figure 10, abnormal ships' behavior have non-recurrent, non-regular and random trajectories so that associated data (track systems) cannot be used to learn the behavior models of those ships. To cover the problems that we met with our probabilistic system in modeling abnormal ships behavior, we chose a complementary approach: the synchronous reactive language. The idea was to extract events (logical data) from the track systems and use them to define behavior models as synchronous reactive systems. We will introduce the reactive modulus in the next section.

5.2. Reactive modulus

5.2.1. Model of behavior

In this system, a ship behavior is modeled by a reactive system implemented in Esterel language that we called scenario. As we have already stated, a ship behavior can be usual or uncommon. Among the uncommon behavior, we have observed two types of scenarios: a non-threatening scenario of ship behavior that does not respect the maritime rules and a dangerous scenario of a ship showing alarming behavior. Thus, we have chosen to implement three types of scenario:

- *Usual scenarios*: the set of scenarios of all commercial boats' behavior that are mandatory to exactly follow the fairways in Figure 5 – (a);
- *Aggressive scenarios*: the set of scenarios that detects abnormal and threatening ship's behavior;
- *Transgressive scenarios*: the set of scenarios that does not respect the maritime rules.

Note that the reactive modulus was basically conceived to deal with abnormal behavior of commercial and small ships that can move freely in the harbor without any restriction. However, in the same manner as probabilistic modulus, we also implemented usual scenarios in this modulus that basically deal with commercial boats' behavior. The resulting performances of both systems will be discussed in the Section 5.3.

We listed in Table 3 some examples of the 40 scenarios that we imagined and implemented.

Usual scenarios	Aggressive scenarios	Transgressive scenarios
towards HB1 towards HB2 towards Mooring N towards Mooring NW towards Mooring SE towards Mooring SW towards Lavera	Change heading in restricted area While Change heading violently in forbidden area Change speed violently in restricted area Change speed violently in forbidden area	Goes too fast in forbidden area Goes too fast in restricted Area Goes too fast in HB1 Goes too fast in HB2 Small Boat in HB1 Small Boat in HB2 Very Fast Boat

Table 3: Exemples of constructed scenarios

5.2.2. Modeling process

Seeing that we have considered a ship behavior as a reactive system, we descided to convert the track systems (positions, speed, time) to logical events that we called facts (Figure 11). They concern the presence or absence of the boat on an area (*is_in_area_A1,...*, *is_in_area_B*), laws on those areas (*is_in_restricted_area,...*, *is_in_forbidden_area*), kinematic computations (change_violently_heading,...,is_too_fast) and ship's type (*is_commercial_boat,...,is_small_boat*).

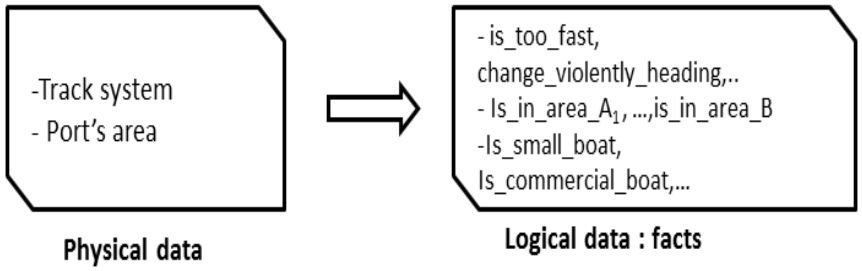


Figure 11: from physical to logical data

The reactive system that models ship's behavior was implemented in Esterel language whose operators manipulate the constructed facts as well as the logical events describing recognition levels of the ship behavior. In fact, each Esterel program takes the facts as input and reacts instantaneously by sending a recognition level signal (Figure 12).

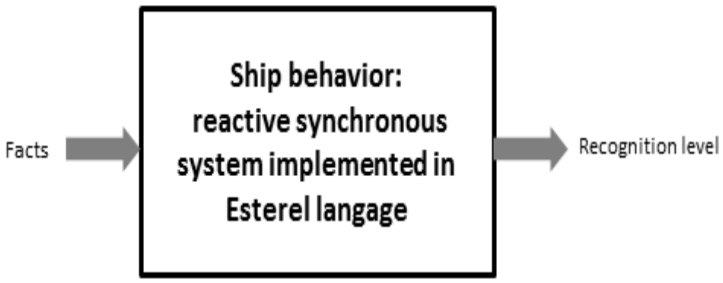


Figure 12: Ship behavior modeled by a reactive system

We have considered 5 signals of recognition levels:

- *standby*: the scenario has been initialized but, so far and regarding the input events suite the scenario has reacted to, it is still not recognizing the boat's behavior;
- *running*: because of particular input events the scenario has reacted to, it has begun its recognition process;
- *pre_matching*: because of particular input events the scenario has reacted on, it has started to match the behavior;
- *non_matching*: the scenario's recognition is no longer matching;
- *matching*: the scenario is recognized.

Note that: *standby*, *running* and *pre_matching* are used as outputs of aggressive and transgressive scenarios; *standby*, *matching* and *non_matching* are used as outputs of usual scenarios. The structure of aggressive and transgressive scenarios is very similar. For this example, we will only describe the example of an aggressive scenario.

```
module ChangeHeadingViolentlyInForbiddenArea:
input is_in_forbidden_area,
change_violently_heading, timeOut;
output standby, running, matching;
loop
   emit standby;
   await is_in_forbidden_area;
   emit running;
   abort
      await change_violently_heading;
      emit matching;
      await timeOut
   when [not is_in_forbidden_area]
end loop
end module
```

Figure 13: Exemple of agressive scenario

An exemple of agressive scenario. Let us take the example of the aggressive scenario of Figure 13 where the behavior to recognize is a sudden change in the boat heading while it moves in a forbidden area:

- The instant the module is run, whatever the values of the facts, the *standby* signal is sent: the scenario is simply initialized;
- At the following instant, the boat moves in a forbidden area (i.e. if the signal *is_ in_forbidden_ area* is present) then the *running* signal is sent: The scenario begins to be recognized;
- Thereafter, while the boat remains in a forbidden area, if it suddenly changes its headings (i.e. the signal *is_ in_forbidden_ area* continues to be present and if the signal *change_ violently_ heading* begins to be present) then the module sends out the *matching* signal: The scenario is recognized and the module begins to wait for a *timOut* temporization;
- The instant the ship exits the forbidden area (i.e. the instant the signal *is_ in_forbidden_ area* becomes absent) then the module sends the *standby* signal whatever the status of the signal *change_ violently_ heading* is (preemption by using the operator "weak abort").

The ship's behavior is no longer considered aggressive if it leaves the forbidden area so we do not need to control its heading. Consequently, the scenario is no longer recognized, sends back the *standby* signal and begins to wait for the boat to return to a forbidden area. At the end of the temporization *timeOut*, our specification has been chosen to reinitialize the scenario. Indeed, we consider the scenario has sufficiently warned the security guard about the dangerous level of this boat. The module returns to its initialization state and the *standby* signal is sent.

A nominal behavior. Commercial boats have very precise, regular and pre-defined behavior, and the corresponding Esterel programs always present the same structure:

- waiting for the boat to be recognized as a commercial one;

- waiting for the boat to pass through a sequence of precisely defined areas describing this commercial boat's mandatory channel;

- accepting the boat track to be lost during one or more areas and reappear again afterwards;

- raising a red level alarm (for a non-matching signal) only when the boat enters areas from which it is strictly forbidden.

Consequently, we have designed a specific computer language that we called SCEnario (SCE) to deal with commercial boats' nominal behavior and their possible deviances. This language is based on four kinds of information:

- a possible empty list of entry guards representing the set of constraints to be fulfilled before the recognition phase of the scenario can be considered as started;

- the ordred sequence of the boat's authorized areas;

- a possibly empty set of forbidden areas. When this set is empty, we consider all areas as forbidden except those in the sequence list;

- a possibly empty list of outgoing guards representing the set of constraints to be fulfilled before the recognition phase of the scenario can be considered as ended.

An SCE program can include input and output signal lists and gives out the same kind of recognition level as the other types of scenario:

- *standby* means the scenario has just been initialized;

- *running* means the scenario has fulfilled its guardes;

- *matching* means the scenario has proceeded correctly;

- *missing_zone* means the boat is still in a permitted area but, as there was a lack of sensor information, the boat is considered as having missed one or more areas during its movements;

- *non_matching* means the boat has entered a forbidden area.

We have implemented in the C++ language, a processor that translates a scenario expressed in the SCE language in an equivalent Esterel program. In the SECMAR project, all types of nominal ship behavior have been implemented for commercial boats using the SCE language.

```
scenario towardsHB1
                                        (a)
include_files
input_signals.strll
output_signals.strll

begin_gardes_list is_a_commercial_boat

nominal_sequence_list

Is_in_area_A1
Is_in_area_A2
Is_in_area_A3
Is_in_area_B
Is_in_area_C
Is_in_area_D
Is_in_area_E

end_gardes_list is_in_area_HB1.

end scenario
```

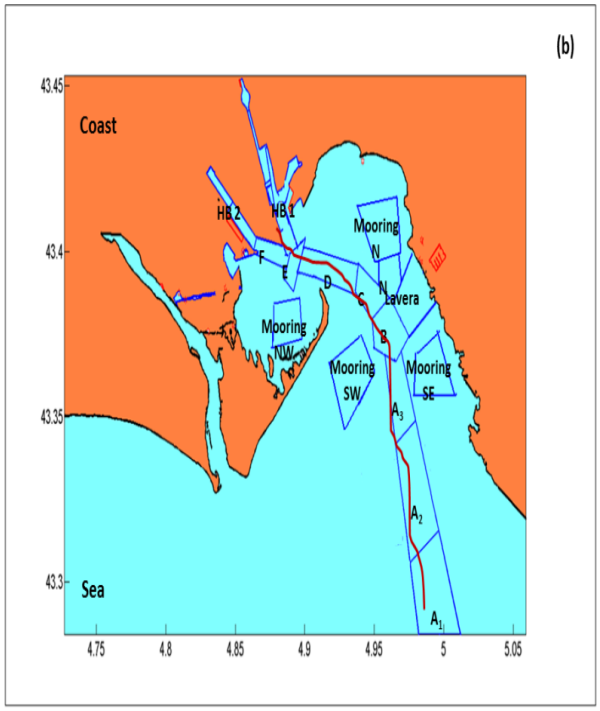


Figure 14: Example of a ship that heads towards HB2 (b) and its corresponding implenetation from Esterel (a)

An example of nominal behavior heading towards HB2 is presented in Figure 14:

1. The corresponding program (Figure 14) first includes input and output files (*inputs_signals_.strll* and *outputs_signals_.strll*).
2. The guard, i.e. the condition to start the scenario's recognition, is that the boat must be a commercial one (i.e *is_a_commercial_boat*).
3. The nominal areas sequence list requires that the boat pass through areas A1 to A3 and then areas B, C, D and E before reaching its terminal area Harbor Bassin 1 (HB1).

This is a description of the Esterel program of this scenario:

- At the first tick, the scenario is initialized (i.e. the *standby* signal is send);
- Thereafter, when the input signal *is_a_commercial_boat* is received by the module, the *matching* signal is sent;
- While the ship goes sequentially through : area A1, area A2; area A3, area B, area C, area D, area E and area HB1, a *matching* signal is sent;
- If an incomplete information is received stating that the ship has missed one or more areas while staying in authorized areas, a *missing_zone* signal is sent. Note that, the SECMAR system designer has considered that this signal carries too poor little information and has asked it to be replaced by a *matching* signal, ignoring the missing areas;

- When a boat enters a forbidden area, a *non_matching* signal is raised. But, as soon as the boat reenters an authorized area, the *matching* signal is emitted again;

5.2.3. Recognition procedure

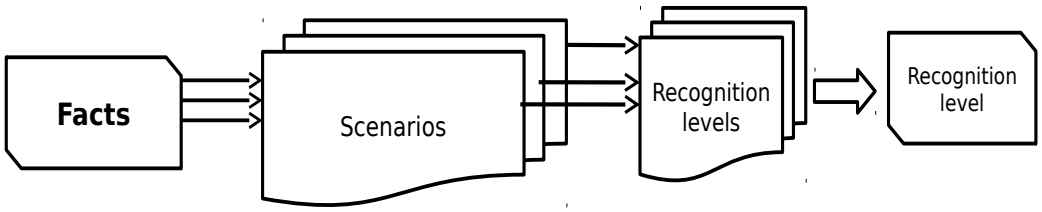


Figure 15: Recognition procedure of reactive modulus

The Figure 15 describes the overall recognition procedure. At each time-step, each scenario receives the set of the facts that are computed from the system tracks obtained after the sensor data merging procedure. Note that at each time-step, all Esterel programs running for the same boat receive the same set of ship information. At each time step, each scenario reacts instantaneously by sending a recognition signal. Consequently, a set of recognition signals is obtained. This set needs to be merged in order to obtain a single recognition level signal at each time step. At Marseilles' Harbor, the maximum number of boats is close to 500 during summer vacations due to a profusion of pleasure boats but less than 30 in other seasons of the year (except during a winter strike where we witnessed more than 70 oil-tankers at anchor). The maximum number of facts needed to describe a boat is less than 50. For the SECMAR project, the duration of the time-step has been estimated at around 3 seconds. This time is greater than the upper time boundary estimated for computing about 100 instantaneous facts per track system (i.e. per boat), to evaluate the 40 scenarios per boat, and merge the recognition level signals for each boat. So the synchronous hypothesis holds.

5.3. Complementarity of probabilistic and reactive modulus

After introducing separately the probabilistic and reactive modulus, we will describe in this section the overall behavior of the BAM. Seeing that both modulus work at the same time, we will explain how they react at each time-slice for a nominal ship behavior and an abnormal ship behavior (described case of Figure 10 –(a)).

We will consider, in this section, three types of colored alarms associated to three levels of recognition:

- Green alarm for a non-threatening behavior.
- Red alarm for threatening behavior.
- Grey alarm for unknown behavior.

5.3.1. Case of nominal behavior

In this section, we will introduce the example of nominal ship behavior that heads towards Lavera.

1. When the ship goes through common ship's behavior areas (i.e area A1, area A2, area A3);
   - The probabilistic modulus sends at each time-slice the most likely behavior by computing: $\arg\max_{\lambda_i} \{P(O_1 ... O_t/\lambda_i)\}_{1 \leq i \leq 7}$

- All nominal scenarios of the reactive modulus send green alarms. This means that observed ship is likely to perform any nominal behavior. Aggressive and transgressive scenarios send grey alarm.

2. When the ship overtakes the area B:
   - As described previously in 1, the probabilistic modulus sends the most likely behavior which is the behavior of ship heading towards Lavera.
   - For the reactive modulus, the following nominal scenarios: *towards Lavera*, *towards Mooring North, towards Harbor Bassin 1*, *towards Harbor Bassin 2* and t*owards Mooring North West* send green alarm, however all other scenarios send grey alarm.
3. When the ship is in area Lavera:
   - In the probabilistic modulus, $\log(P(O/\lambda_3)) \geq \log(P(O/\lambda_i))_{1\leq i\leq 7, i\neq 3}$ where λ_3 is associated to ship behavior that heads towards Lavera: The ship is heading towards Lavera.
   - In the reactive modulus, only the Esterel scenario *towards Lavera* will be sending a green alarm.

5.3.2. Case of abnormal behavior

In this section, we will describe the case of an abnormal ship behavior (Figure10 – (a)). The ship changes unawares its heading from Mooring North to Lavera.

1. When the vessel goes through area A1 until area N, the probabilistic and reactive modulus react exactly as described in the example 5.3.1.
2. When the ship changes suddenly its heading from Mooring North to Lavera.
   - By probabilistic computing, the probabilistic modulus detects a trajectory change.
   - In the reactive modulus, the aggressive scenario *Change Heading Violently* sends a red alarm. All the other scenarios send a grey alarm.

In this case of ship behavior, the probabilistic modulus detects a trajectory change which does not necessarily means that the ship is performing an abnormal behavior. This information is completed by an abrupt change in trajectory detected by the reactive modulus.

Thus, the ship's behavior is considered as abnormal.

6. Conclusion

Nowadays, the vessel detection system of most harbors record a large amount of data that are directly handled and analyzed by the security guards in order to recognize common and uncommon ships behavior. Since this task is difficult and fastidious for human, we have proposed in this paper, an automatic decision support system to help security guards to better monitor the vessels' behavior. In our knowledge, the proposed framework has never been proposed before. It is a combination of probabilistic and reactive approaches and constructed by two parallel sub-systems. In the probabilistic modulus, the ship behavior is modeled by the Hidden Markov Models approach. The model of behavior is initialized and then learned off-line. This system is suitable for commercial boats' nominal behavior and is able to predict and recognize on-line the ship behavior by probabilistic computing. The reactive modulus models a ship behavior by a reactive synchronous system programmed in Esterel language. As for the probablistic system, the models of ships behavior are also specified off-line. This system shows promising performance for handling the abnormal behavior of all ships moving in the harbor. The recognition is achieved by instatntaneous reaction and transmission of signals of recognition levels. Thus, probabilistic and event-based approaches proved to be complementary. Combining both approaches, we were able to construct a strong system

for surveying ship behavior in Marseilles' harbor. This system can be used suitably in all harbors on the condition that broad and accurate ship information sensor data is available, fairways of ship behavior are known in order to learn behavior models using a probabilistic approach and a large number of facts have been built up. Like airports, harbors can face threatening events such as terroristic attacks, making a decision support system particularly helpful.

References

[1] P. Bessiére, *Probability as an Alternative to Logic for Rational Sensory–Motor Reasoning and Decision*. Springer, 2008.

[2] J. Yang, Y. Xu, and C. Chen, "Hidden markov model approach to skill learning and its application to telerobotics," *Robotics and Automation, IEEE Transactions on*, vol. 10, no. 5, pp. 621–631, 1994.

[3] K. Han and M. Veloso, "Automated robot behavior recognition," vol. 9, pp. 249–256, 2000.

[4] N. Kuge, T. Yamamura, O. Shimoyama, and A. Liu, *A driver behavior recognition method based on a driver model framework*. SAE International, 2000.

[5] S. Sekizawa, S. Inagaki, T. Suzuki, S. Hayakawa, N. Tsuchida, T. Tsuda, and H. Fujinami, "Modeling and recognition of driving behavior based on stochastic switched ARX model," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 8, no. 4, pp. 593–606, 2007.

[6] X. Meng, K. Lee, and Y. Xu, "Human driving behavior recognition based on hidden markov models," in *Robotics and Biomimetics, 2006. ROBIO'06. IEEE International Conference on*, pp. 274–279, IEEE, 2006.

[7] C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins, and G. Pappas, "Symbolic planning and control of robot motion [grand challenges of robotics]," *Robotics & Automation Magazine, IEEE*, vol. 14, no. 1, pp. 61–70, 2007.

[8] A. Sowmya, D. Tsz-Wang So, and W. Hung Tang, "Design of a mobile robot controller using esterel tools," *Electronic Notes in Theoretical Computer Science*, vol. 65, no. 5, pp. 3–10, 2002.

[9] S. Levinson, L. Rabiner, and M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition," *The Bell System Technical Journal*, vol. 62, no. 4, pp. 1035–1074, 1983.

[10] L. R. Rabiner, "A tutorial on HMM and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257–286, 1989.

[11] A. Nefian and M. Hayes III, "Hidden Markov models for face recognition," in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, vol. 5, pp. 2721–2724, IEEE, 2002.

[12] J. Lee, J. Kim, J. Kim, *et al.*, "Data-driven design of hmm topology for on-line handwriting recognition," in *The 7th International Workshop on Frontiers in Handwriting Recognition*, Citeseer, 2000.

[13] N. Oliver and A. Pentland, "Graphical models for driver behavior recognition in a smartcar," pp. 7 –12, 2000.

[14] A. Pentland and A. Liu, "Modeling and prediction of human behavior," *Neural Computation*, vol. 11, no. 1, pp. 229–242, 1999.

[15] N. Halbwachs, *Synchronous programming of reactive systems*. 1998.

[16] E. Clarke, "Model checking," in *Foundations of Software Technology and Theoretical Computer Science*, pp. 54–56, Springer, 1997.

[17] A. Charles, "Representation and analysis of reactive behaviors: A synchronous approach," 1996.

[18] G. Berry, P. Couronne, and G. Gonthier, "Synchronous programming of reactive systems: an introduction to ESTEREL," 1987.

[19] V. Papailiopoulou, "Automatic test generation for lustre/scade programs," in *Proceedings of the 2008 23rd IEEE/ACM International Conference on Automated Software Engineering*, pp. 517–520, IEEE Computer Society, 2008.

[20] A. Benveniste, P. Le Guernic, and C. Jacquemot, "Synchronous programming with events and relations: the signal language and its semantics," vol. 16, pp. 103–149, Elsevier, 1991.

[21] G. Berry and G. Gonthier, "The esterel synchronous programming language; design, semantics, implementation," *Science of computer programming*, vol. 19, no. 2, pp. 87–152, 1992.

[22] S. Elloumi, J. Marmorat, V. Roy, and N. Maizi, "Modeling ship behavior based on hidden markov models," in *Signal Processing, Pattern Recognition, and Applications/722: Computer Graphics and Imaging*, ACTA Press, 2011.