A METHODOLOGY FOR BUILDING A DATA-ENCLOSING TUNNEL FOR AUTOMATED ONLINE-FEEDBACK IN SIMULATOR TRAINING

Laura Marcano[1,*], Davide Manca[2], Anis Yazidi[3], Tiina Komulainen[1]

[1]Department of Electronic Engineering
OsloMet – Oslo Metropolitan University
PO Box 4, 0130, Oslo

[2]*PSE-Lab*, Process Systems Engineering Laboratory
Dipartimento di Chimica, Materiali e Ingegneria Chimica "Giulio Natta"
Politecnico di Milano
Piazza Leonardo da Vinci 32, 20133 Milano, Italy

[3]Department of Computer Science
OsloMet – Oslo Metropolitan University

*Corresponding author: phone +4745553909
e-mail: laura.marcano@oslomet.no

Abstract

Extensive research confirms that feedback is the key to an effective training. However, in many domains, human trainers, who can provide feedback to trainees, are considered not only a costly but also a scarce resource. For trainees to be more independent and undergo self-training and unbiased support, effective automated feedback is highly recommended. We resort to elements from the theory of data mining to devise a data-driven automated feedback system. The data-enclosing tunnel is a novel concept that may be used to detect deviations from correct operation paths and be the base for automated feedback. Two case studies demonstrate the viability of this methodology and its usefulness in industrial simulation scenarios. Case study 1 focuses on the increase of oil production, whilst case study 2 decreases gas production. The data-enclosing tunnel is validated and compared with three other assessment methods. These methods are simpler versions of the data-enclosing tunnel method, as they are three variants of a baseline approach Data Enclosing Band (DEB), namely DEB1, DEB2, DEB3. The methods accuracy is determined by calculating how precisely they can classify new data. The data-enclosing tunnel yields the highest accuracy, 94.3 %, compared to 81.4 %, 62.9 %, and 70 % for DEB1, DEB2, DEB3 respectively.

Keywords: data analysis, data mining, automated feedback, industrial training, simulator training.

1 Introduction

Feedback is a crucial factor in effective simulator training. It is possible for trainees to learn from their errors if they receive clear and timely diagnostic feedback about their performance (Kluge et al., 2009, Salas et al., 2012). Typically, a trainer is responsible for guiding the trainees through the simulation task and providing relevant feedback when necessary. Nevertheless, the availability of expert instructors is decreasing, mainly due to retirement (Komulainen and Sannerud, 2018, Nazir and Manca, 2015). Therefore, industries are facing the challenge of fulfilling the increased training demand with a limited number of instructors. This situation could be overcome with the implementation of simulator training practices that allow the trainees to be more independent so that the need for expert instructors can be alleviated (Marcano et al., 2019).

One way of helping trainees to be more independent during simulator training consists of offering real-time automated feedback (Bell et al., 2008, Malakis and Kontogiannis, 2012, Manca et al., 2014). If trainees receive automated feedback, they will not have to rely exclusively on the instructor. Further, with automated feedback, trainees can receive comments and guidance faster, since they will not have to wait until the instructor is available. Also, automated feedback allows remote training, which can represent a cost reduction for industries. If remote training for technical skills is promoted before on-site training, the time needed in the training center could be compressed. Thus, costs related to the operators' mobility could be saved. Automated online feedback can also motivate operators to train more often by themselves since they will count on having relevant and prompt guidance, and they will be able to train at their own pace. On the other hand, automated feedback could also be used as a support tool for novice instructors. It could guide inexperienced instructors on what kind of feedback to give to the trainees.

Automated feedback for simulator training is not a new concept. Automated feedback has been already an active topic for research especially in health-care education (Rhienmora et al., 2011, Sewell et al., 2008). There is also extensive research on intelligent tutoring systems (ITSs) as an educational tool to help trainees outside the classroom (Mohamed and Lamia, 2018), to learn a new language (Mahmoud and Abo El-Hamayed, 2016), and even in serious games (Goldberg and Cannon-Bowers, 2015), which are games designed for a training purpose other than pure entertainment. Gonzalez-Sanchez et al. (2014) indicate that ITSs could become a steady and economic alternative to human instructors. However, little research can be found specifically on automated feedback for industrial simulator training (Manca et al., 2014, Speshilov and Khabarov, 2017). Research has been done on how to improve operators performance based on the analysis of operational records (Sebzalli et al., 2000, Lee et al., 2000, Yamashita, 2000). However, these studies did not aim to develop automated feedback. The operator training simulator market is expected annually to exceed USD 20 billion by 2025 (Market Study Report, 2019). This gives an overview of the great importance and extension of this field. Therefore, it is essential to intensify research efforts in the same area.

This paper presents and discusses a novel data-mining approach to provide automatic feedback to trainees. Our approach resorts to a novel concept called data-enclosing tunnel, which can be seen as a data envelope describing the expected evolution of the simulation process. We show that by using the data-enclosing tunnel we can automatically detect deviations from correct executions paths and issue an automated corrective feedback to the user. As an industrial large-scale simulation use case, we consider the dynamic process simulator K-Spice (Kongsberg, 2009), from Kongsberg Digital.

2 Methodology

Figure 1 shows the different steps of the data-enclosing tunnel methodology. Such a methodology is based on a data mining approach. Data mining is the process of examining large amounts of data to discover novel and useful information (Baker, 2010). The steps in the methodology are described in detail in Sections 2.1-2.8. Every step is primarily based on the researchers' practical experience gained during the development of this work.

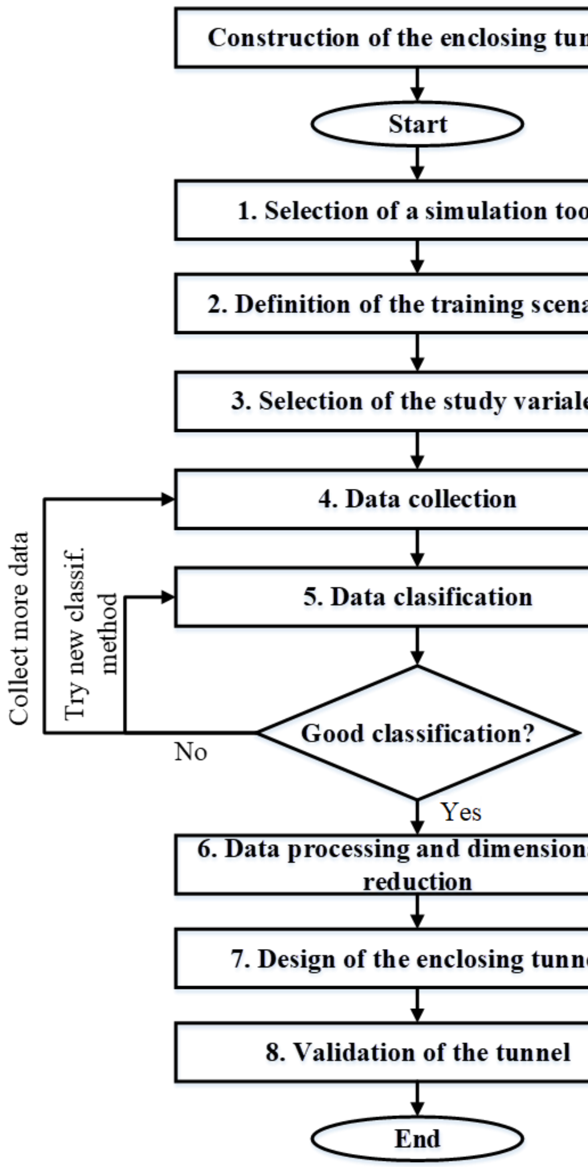


Figure 1. Overview diagram of the methodology for constructing a data-enclosing tunnel.

2.1 Simulation tool

The simulation tool is a dynamic model of the process. It should have functionalities for saving and exporting historical data. It should feature training scenarios, and offer the possibility to create/configure them. If feedback messages cannot be issued within the simulation tool, the tool should be able to connect with a server and send the data to an external program where the feedback message can be displayed.

2.2 Defining a training scenario

The training scenario for automated feedback can be selected from the available ones (in the simulator) or created from scratch. The training scenario should suggest clear operational goals and well-defined learning objectives. The performance of the trainee can be tracked based on whether they are reaching the established goal or not.

2.3 Selection of study variables

The selection of variables to be recorded and monitored depends on the case study. It is advisable to choose those variables that are related to the operational goals and learning objectives. In addition, the complexity of the process also plays an important role when it comes to selecting such variables. Complex processes may require monitoring a large number of variables (Ghosh et al., 2014). In these cases, Key Performance Indicators (KPIs) and Operator Performance Indicators (OPIs) are valuable tools. Performance indicators can be useful metrics based on the combination of several variables. KPIs refer to the production efficiency of the industrial process, while OPIs refer to human performance (Manca et al., 2012, Marcano and Komulainen, 2018). The use of performance indicators can be useful to simplify the number of variables to study.

2.4 Data collection

Several literature manuscripts highlight the great value that can be found in the analysis of process operational data (Sebzalli et al., 2000, Yamashita, 2000, Shu et al., 2016). It is worth gathering data that describe different ways in which a training scenario can be carried out. This data should be rich enough to document different routes that allow either solving or failing a task so that a useful feedback tool can be developed based on the analysis of these records.

Since the feedback tool is developed to support trainees, the data collected should record the performance of actual users when they solve the proposed training scenario. However, sometimes data from actual users is not available, either because the performance of previous users has not been recorded, or because the tool is developed for a new scenario that has not been tested yet. In those cases, the reference data can be generated by implementing an algorithm that makes different combinations of plausible actions. A repository of several probable actions, good and bad, should be produced based on the knowledge gathered from observing actual trainees (and possibly expert users and trainers) using the simulator. The algorithm should randomly choose among several alternatives from the repository and create different combinations of them to solve the scenario, thus ensuring human unpredictability.

2.5 Data classification

The data gathered from one user corresponds to one sample of the overall data. Each sample consists of multivariate time series. It is necessary to classify the samples that correspond to good execution paths and the ones that correspond to bad execution paths, to create balanced groups to do the training and the validation, i.e. each group should have the same amount of good and bad paths. The simplest way to do this is to label the data records of the actual user right after they solved the training scenario. Likewise, in the case of generated data, the data should be classified as soon as it is created.

Nevertheless, if the data is not labeled as soon it is created, there are different methods to cluster it based on its characteristics. In order to cluster data, it is necessary to use a notion of similarity. This can be done by calculating the distance between every possible combination of pairs of execution paths. Marcano et al. (2018) present a detailed explanation of three different methods that can be used to calculate the distances between the execution paths, and how the data can be classified and labeled as good or bad based on these distances.

2.6 Data processing and dimensionality reduction

If the training data is multi-dimensional, it is preferable to reduce the data dimension (Ghosh et al., 2014). In the following, we describe the approach applied in our case studies that use the principal component analysis, PCA. The PCA analysis must be executed for different time slots that include all the training data. This allows ensuring that all the samples of the training data are compared with each other.

Each time slot is defined using the sliding window algorithm (Fumarola et al., 2009). The number of elements must be chosen according to the window size. If the number chosen is w, this means that the first time slot covers the range from the first to the w^{th} element of each sample, i.e. the range [1-(1+w-1)], as shown in Figure 2. The second time slot covers the range from the second to the w^{th} plus one element of each sample, i.e. the range [2-(2+w-1)], and so forth until the entire time-range for each sample is covered, [(L-w+1)-L] being the last time slot range, where L is the length of each sample. The average value of the elements within each range is taken for each sample. Each average corresponds to a row in a matrix featuring as many rows as the samples in the training data (see also Figure 2). The first PCA is calculated for the matrix of the first time slot. The second PCA will be calculated for the matrix of second time slot, and so forth. The analysis of the data is made with the PC1s and the PC2s obtained with the PCA calculated for each time slot matrix. The window size depends on the PCA projection. The number should be adjusted to reduce the noise in the graph of the scores of PC1 vs the scores of PC2 vs time. Empirical practice shows that the larger the window size, the smoother the graph will be since in this case more data samples are averaged-out. However, it is not desirable to use a too large window size, because there is a risk of losing valuable trending information of the data. Nonetheless, the definition of the optimal window size is a tedious task. In this study, we opted for an empirical choice of the window size by experimentally trying different sizes and retaining the value that gave us a smooth curve.

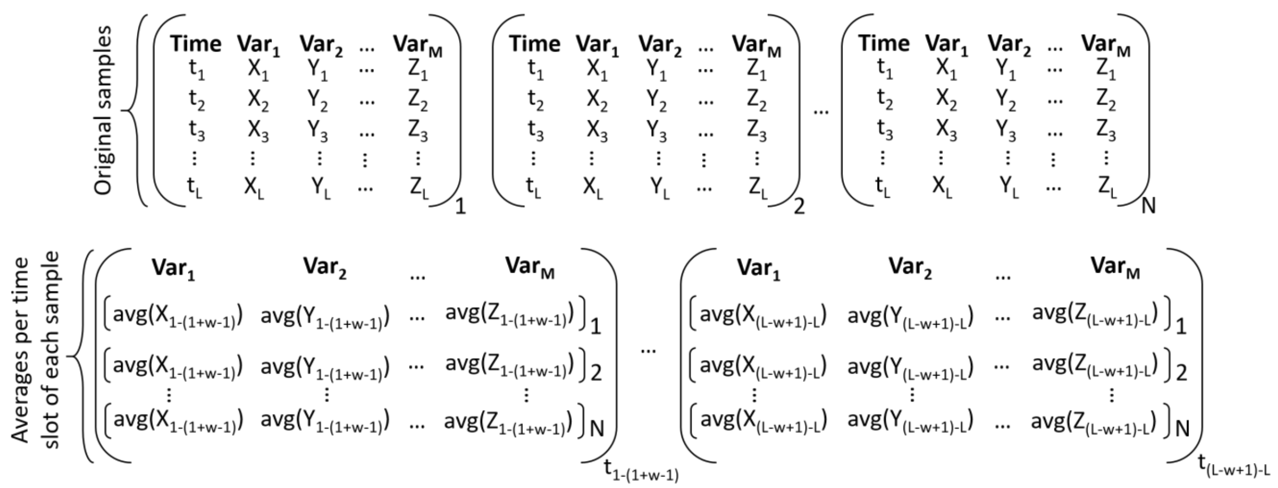


Figure 2. Data handling overview.

2.7 Enclosing tunnel

To construct the enclosing tunnel (Marcano et al., 2018), first, the data projected on the PCA plane (Figure 3a), corresponding only to the good execution paths, must be observed to identify the points in time where the data makes drastic changes, this is shown in Figure 3b with the red horizontal lines. Then, the data must be studied right on each of these points; to do this it should be observed from the 2D plane formed by the PC1 scores vs the PC2 scores. Next, the points projected on this plane are framed using the minimal enclosing circle problem (Weisstein, 2018). Figure 3c shows an example of the first circle built projecting all the data before the first change, that is approximately from 0 to 50 s. Eventually, there will be as many circles as the points where data changes, as shown in Figure 3d. The enclosing tunnel is constructed by drawing a surface around all those circles. Each colored line in these figures represents a different execution path, i.e. a different way according to which a training scenario is carried-out.

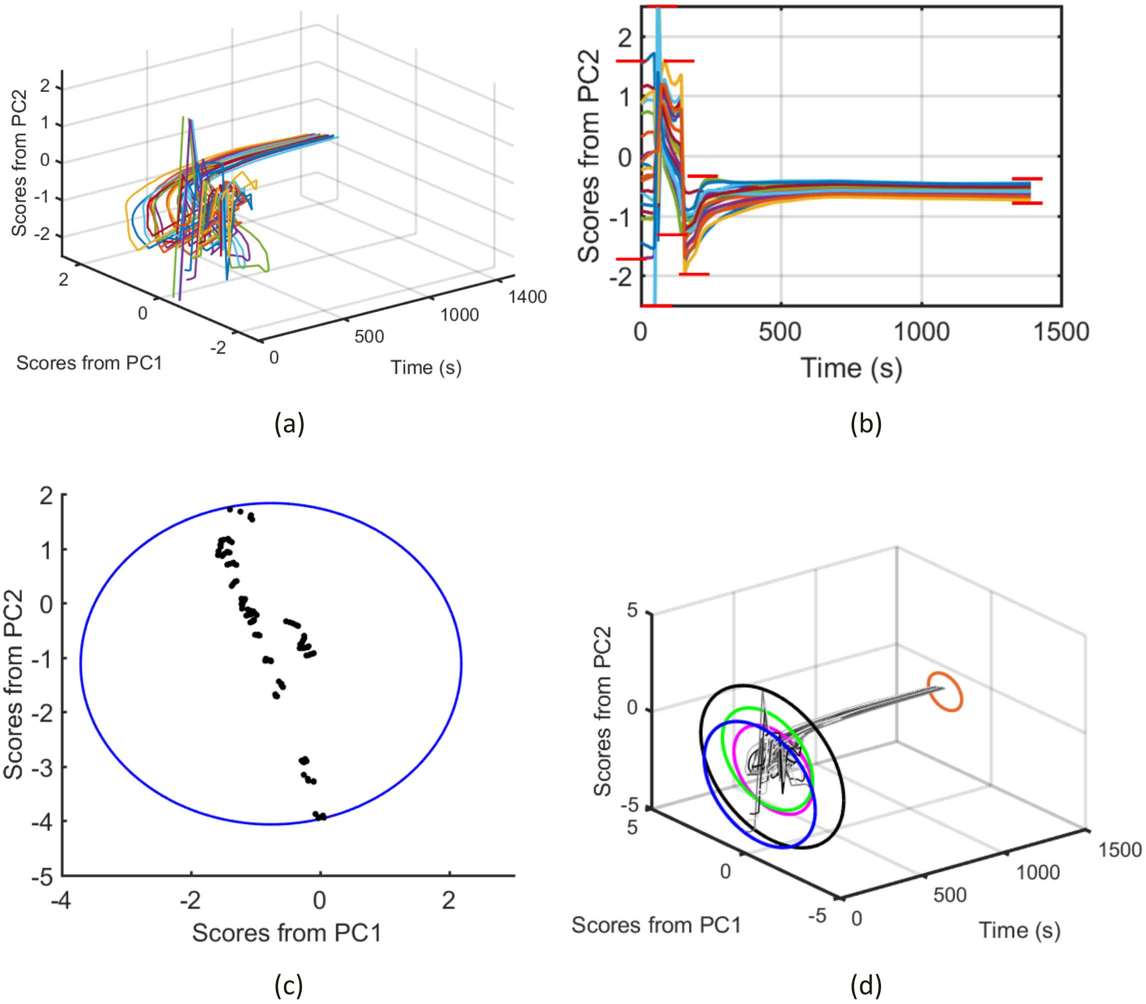


Figure 3. Overview of the setup of the data-enclosing tunnel. (a) Projection of the data on the PCA plane. (b) Lateral view of the data projected on the PCA plane. (c) Front view of the data projected on the PCA plane, first minimum enclosing circle. (d) 3D view of the data projected on the PCA plane and all the minimum enclosing circles built.

2.8 Validation of the tunnel

The dimensionality of the validation data must be reduced using the PCA models obtained with the training data. Then, the projected validation data should be plotted together with the enclosing tunnel. Later, it must be determined which execution paths fall inside the tunnel, which ones outside and for how long. Eventually, together with the data labels, those determine during the classification step (Section 2.5), the accuracy of the tunnel can be calculated. Two metrics were established to determine the accuracy of the enclosing tunnel:

1. Execution paths that fall outside the enclosing tunnel for more than 35 % of the total scenario time are considered as "bad".
2. Execution paths that fall outside the enclosing tunnel for more than 50 % of the total scenario time are considered as "very bad".

It is worth observing that the 35 % and 50 % values, which denote a bad and very bad execution path, are empirically chosen. These metrics were tested to evaluate the method performance from two different perspectives, given that, in some cases, a more flexible metric might be still acceptable. A more flexible metric means that the trainee can take more time to figure out how to correct a mistake when they went wrong. Further, depending on the metric used, the difficulty of the simulation exercise can be controlled. For more experienced trainees, the threshold can be lowered down to tolerate only small deviations from the optimal path. Finally, the validation results of the tunnel must also be compared with a state of the art trajectory, which could be used as a baseline.

3 Case studies

The following paragraphs present how we developed, implemented, and tested the enclosing tunnel methodology, which is applied to two training scenarios.

3.1 Simulation tool for the case studies

The process simulations used to train the trainees were carried out on K-Spice (Kongsberg, 2009), a dynamic simulator from Kongsberg Digital. K-Spice enables detailed dynamic simulation of oil and gas processes and control systems. It is a Windows-based tool designed for different engineering applications, including operators' training (Kongsberg, 2009). The training scenarios were simulated with K-Spice oil and gas production model. The model consists of a three-stage, three-phase separation train, the utility systems, and emulated control and safety systems (Komulainen and Løvmo, 2014).

3.2 Training scenarios

We developed two simple scenarios to strengthen the overall understanding of an oil and gas production process. The first scenario calls the trainee for increasing the oil production, which is one of the main goals of an oil production facility. The second scenario calls the trainee for decreasing the gas production; this situation occurs when it is necessary to control the gas pressure in the system or the quality of the exported gas. The two training scenarios were defined as follows:

Scenario 1 (SC1): the target is to increase, in 30 min, at least +10 % the oil production flow compared to the initial conditions of the simulation.

Scenario 2 (SC2): the target is to decrease, in 30 min, 10 % of the gas production compared to the initial conditions.

The trainee must fulfil the goals without compromising the correct operation of the process; this means that the changes made by the trainee must not create process upsets such as over-pressuring the system, overflows, leakages, process shutdown and the like. The trainee should be able to execute actions that lead to smooth transitions in the system.

3.3 Monitored variables

In the generic oil and gas production model, the sections with the most relevant process information for the two training scenarios are the wells, the high-pressure separator (HP-separator), the export pump and the gas export compressor, the oil and gas export sections, and the high-pressure flare (HP-flare). The monitored variables of these sections are: 1) total sum of outlet flow rates from the wells; 2) inlet flow rate of the HP-separator; 3) pump power consumption; 4) compressor power consumption; 5) oil export flow rate; 6) gas export flow rate; and 7) HP-flare flow rate, as shown in Figure 4.

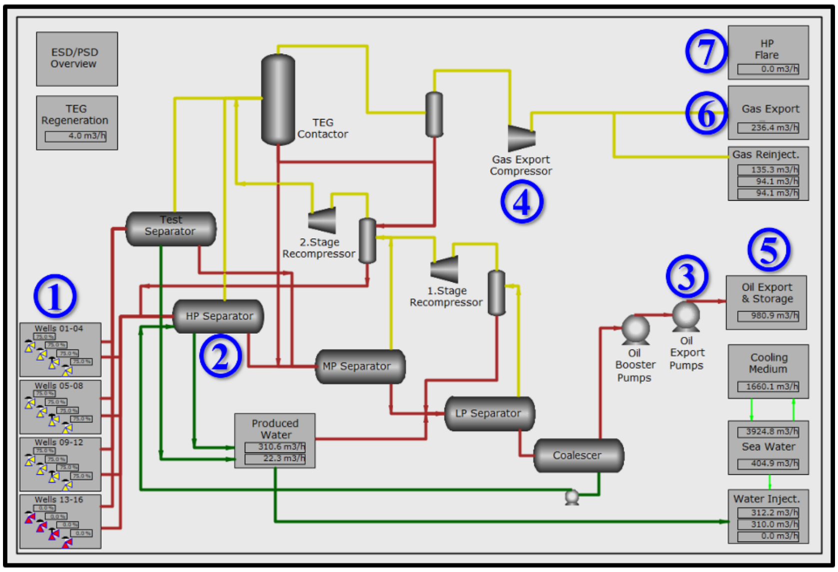


Figure 4. Overview of the generic oil and gas production process and monitored variables.

3.4 Data collection for the case studies

The monitored and recorded data was generated with an algorithm that ensures the creation of different execution paths for the training scenarios. The algorithm chooses an execution path based on random selections from several possible actions. The actions are defined based on the observations and results gathered from the simulator training sessions mentioned in (Marcano et al., 2017). Only one main action, with a maximum of two subsequent actions, can take place per execution path. For SC1, the delay options between the subsequent actions are set to 15, 45, 60, 120, and 180 s. Indeed, during the simulator training sessions, we noticed that the participants did not wait more than 3 min (i.e., 180 s) to make changes in the simulation.

For SC2, the delays for the first action were the same as for SC1. Conversely, the delays between the first action and the subsequent actions were set to 180, 240, and 300 s to provide enough time for the trainees to evaluate the percentage change of the gas flow rate.

The algorithm chooses an execution path as follows:

1. The first action is taken randomly among the main options.
2. Whether the first action is followed by one, two or no more actions is also decided randomly.
3. Depending of the total amount of selected actions, a delay value is also randomly chosen for each action.
4. If the final combination of actions and delays is different from previous configurations, the execution path is saved. Otherwise, go back to step one.

The main actions of each scenario studied are explained below.

3.4.1 Scenario 1: Increase oil production

Below five possible actions that the trainees may execute attempting to solve the scenario are explained. There exist other possibilities, but the ones chosen are those that were observed more often during the simulator training sessions mentioned in (Marcano et al., 2017).

1. *Increasing the flow from a well*

   Increasing the flow from a well is the right decision when trying to increase oil production; this can be done by opening a choke valve. We assumed that if the first decision of the trainee is to open a choke valve, then, the following actions, if any, should be to open more choke valves. Once the first choke valve is opened, the algorithm decides randomly whether one, two, or no more valves will be further opened. The opening range of the choke valves is also a random decision between two options: 85 % and 100 %. In the simulation, all those choke valves that are open, are set at 75 % opening.

2. *Decreasing the flow from a well*

   As a rule, decreasing the flow from a well is an incorrect approach as the oil production is expected to increase. To decrease the flow from a well, the trainee has to close a choke valve. If the trainee is confused and closes a choke valve by mistake, the next actions might be closing even more valves. However, the trainee may notice the error and try to fix it by reopening the closed valve and opening an extra one. The algorithm decides randomly whether the action of closing a choke valve is followed by one, two, or no more actions. In case of one more action, this could be either closing another valve or reopening the one that was closed. In case of two subsequent actions, these would be to reopen the closed valve and open an extra one. How much a choke valve is closed is also a random decision between two options: 0 % and 65 %. As far as the opening range is concerned, the same above conditions apply.

3. *Opening an Emergency Shutdown (ESD) valve*

   The simulator training sessions discussed in Marcano et al. (2017) allowed noticing that some participants opened an ESD valve mistaking for a choke valve. Given that opening an ESD valve is a rare mistake, we did not define subsequent actions for it.

4. *Increasing the pressure set point of the HP-separator*

   Opening the HP-separator outlet valve may occur due to a misconception. Indeed, some trainees think that by increasing the outlet flow from the HP-separator, the oil production would increase as well. The next step is to choose whether to proceed with one, two, or no more actions. If two actions are chosen, these are set to be the opening of two choke valves. If only one more action is selected, this can be opening either a choke valve or an ESD valve.

5. *Opening the outlet control valve of the HP-separator*

   Increasing the pressure of the HP-separator leads the system to switch on the high-pressure flare. This action allows accounting for execution paths with a negative environmental impact. In case of only one following action, this can be opening either a choke valve or an ESD valve. In case of two following actions, both of them will be opening a choke valve.

3.4.2 Scenario 2: Decrease gas production

In the following, we describe four possible actions that the trainees may execute attempting to solve the scenario. There are further possibilities, but the ones chosen are those that were observed to be more intuitive for the trainees as commented in (Marcano et al., 2017).

1. *Decreasing the flow from a well*

   Decreasing the opening of a choke valve from 75 % to 60 % is the right decision when trying to decrease 10 % of the initial gas production. If this happens, it will be enough to reach the goal, so no more actions will follow. However, a trainee might consider fully closing a choke valve or moving to values that might not be suitable for reaching the scenario's goal. Therefore, they will have to reopen the choke valve. Then, if the trainee opens the valve too much, they might have to close it again. Several options are defined to cover most of the aforementioned alternatives; these are presented in Table 1.

Table 1. Defined options when the first action is closing a choke valve.

Sequence condition	Close choke valve down to (%)	Reopen choke valve up to (%)	Reclose choke valve down to (%)
If only first action	60	-	-
If first action followed by one action	0	40	-
		60	-
	40	50	-
		60	-
	70	60	-
If first action followed by two actions	0	40	60
	40	50	
	70	65	

(-) Not applicable

2. Increasing the flow from a well

Increasing the flow from a well is an incorrect approach when the gas production needs to be decreased. If the trainee is not sure of what they are doing, they might make this mistake. On the other hand, the trainee could notice the error and try to fix it by closing the opened valve. One, two, or no more actions may follow the opening of a choke valve. Table 2 shows the available options for this case.

Table 2. Defined options when the first action is opening a choke valve

Sequence condition	Open choke valve up to (%)	Close choke valve down to (%)	Reopen choke valve up to (%)
If only main action	85	-	-
	100	-	-
If main action followed by one action	85	0	-
		50	-
	100	0	-
		50	-
If main action followed by two actions	85	0	20
			60
		50	60
	100	0	20
			60
		50	60

(-) Not applicable

3. *Closing the Emergency Shutdown (ESD) valve of a well*

   The trainee can choose to close the ESD valve of a well. This action would decrease the gas production significantly more than 10 %. Consequently, this is an incorrect procedure. Closing the ESD valve of a well drastically affects the gas flow. Therefore, only one subsequent action may follow this one, and this is reopen the ESD valve.

4. *Closing the Emergency Shutdown (ESD) valve from the HP-separator to the Contactor*

   The trainee may be mistaken and think that if the gas flow from the HP-separator decreases then the gas production drops too. Therefore, they might reduce the opening of the ESD valve of the HP-separator that regulates the flow to the Contactor. Then, when noticing that this decision barely affects the gas production flow, they might continue closing the valve until the gas accumulates in the system, the pressure increases, and finally the high-pressure flare is operated.

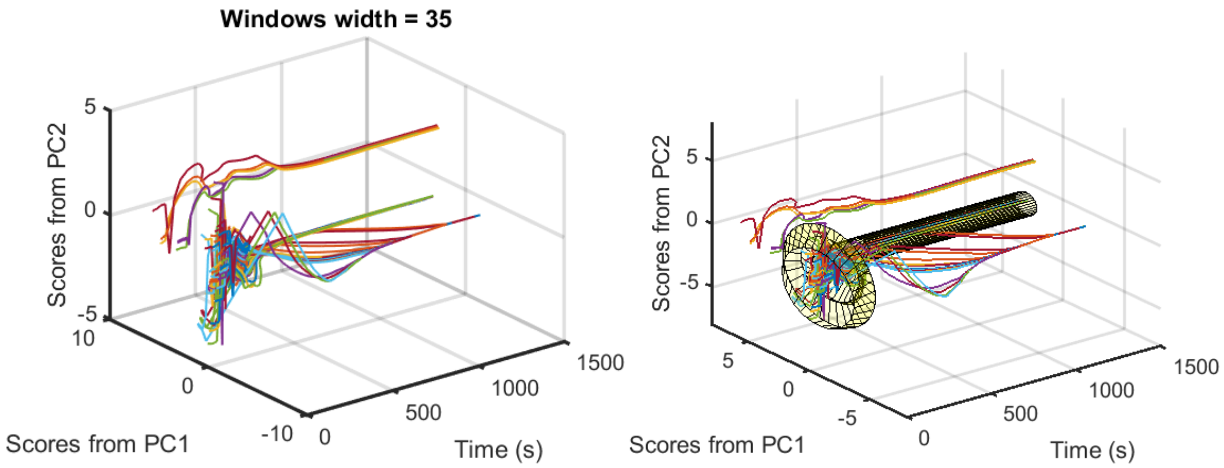
3.5 Classification of the case studies data

For SC1, 75 different samples were generated, of which two-thirds were used for training and one-third for validation, i.e. 50 samples for training and 25 for validation. The training and validation sets had a balanced number of good and bad execution paths. The data used for the first scenario was not labelled as soon as it was generated, so it was classified using hierarchical clustering, and later labelled as good or bad. A detailed explanation of how the data was classified can be found in (Marcano et al., 2018).

For SC2, 200 different samples were generated, of which 65 % were used for training and 35 % for validation, i.e. 130 samples for training and 70 for validation. Again, we ensured that each group had a balanced number of good and bad execution paths. The data used for the second scenario was labelled as soon it was generated.

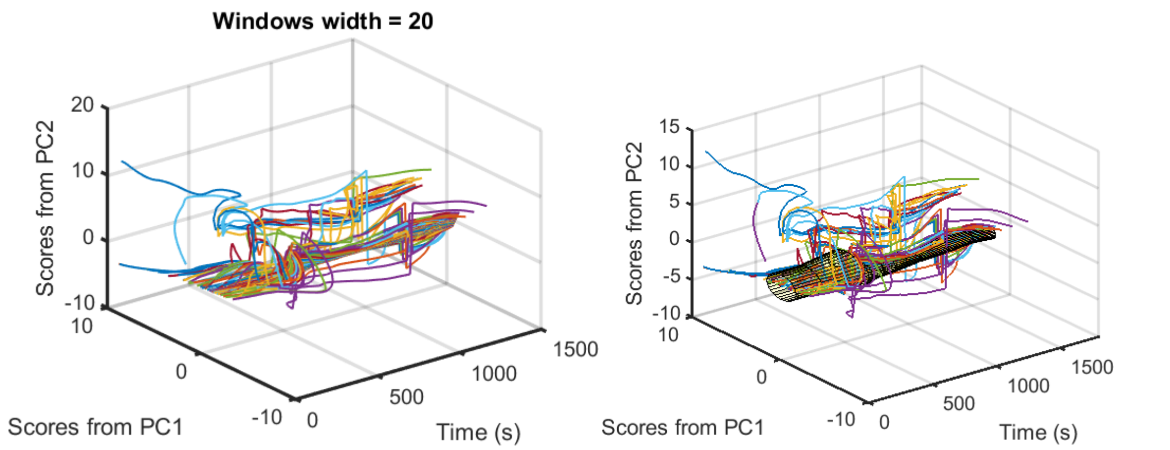
3.6 Data processing and dimensionality reduction of the case studies

The time moving average in SC1 was calculated using a window size of 35 elements. Conversely, the most suitable window size for SC2 was of 20 elements. As mentioned in the methodology (see Section 2.6) the size of the moving average is adjusted until the graph of the scores of PC1 vs the scores of PC2 vs time is smooth, based only on empirical observation of the graph. This is done to decrease the noise in the curves, distinguish each path clearly, and later build the data-enclosing tunnel. Figure 5a and Figure 6a show the curves of the training data scores of PC1 vs the scores of PC2 vs time, for SC1 and SC2 respectively. The figures show the distribution of the data. It can be seen that the curves form clusters in some areas of the graph. Some of these clusters correspond to good execution paths and some to bad execution paths, although it is easier to appreciate the groups in Figure 5a since fewer data are used for SC1. Each colored line in the figures corresponds to a different execution path generated as explained in Section 3.4.



(a) (b)

Figure 5. (a) SC1 – Scores from PC2 vs Scores from PC1 vs Time / (b) SC1 – Data-enclosing tunnel and training data.



(a) (b)

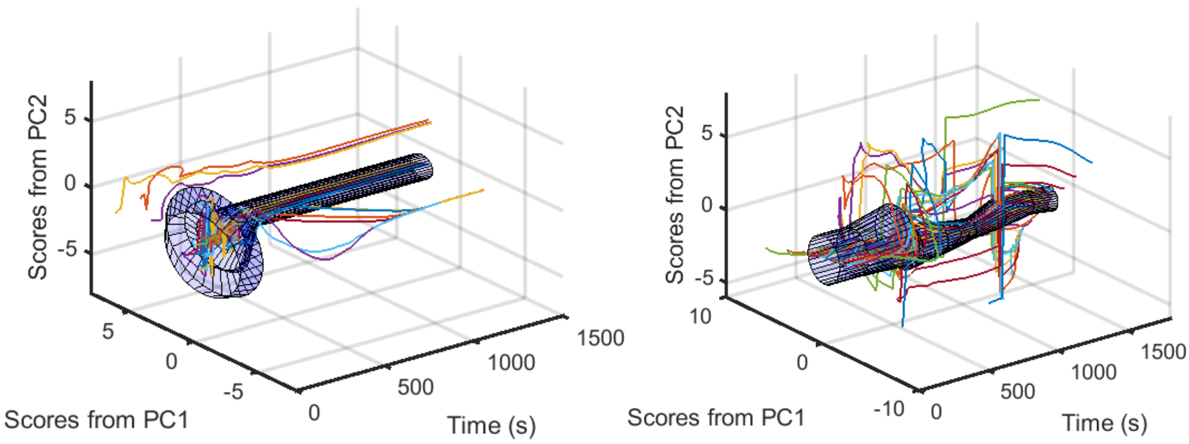
Figure 6. (a) SC2 – Scores from PC2 vs Scores from PC1 vs Time / (b) SC2 – Data-enclosing tunnel and training data.

3.7 Enclosing tunnels of the case studies

The enclosing tunnel designed for SC1 has five different radiuses, Figure 5b shows the SC1 training data plotted together with its corresponding data-enclosing tunnel. The tunnel designed for SC2 has eleven different radiuses; Figure 6b presents the SC2 training data plotted together with its related data-enclosing tunnel. The data enclosing tunnels were created only using the good execution paths, as explained in Section 2.7. Figures 5b and 6b present the tunnels plotted with all the training data to show that the curves inside the tunnel are the good execution paths and the curves outside the tunnels are the bad execution paths.

3.8 Validation of the tunnels of the case studies

As indicated in the methodology, the validation of the tunnel was made by calculating how many of the execution paths in the validation data ended correctly inside or outside the enclosing tunnel. Figure 7a and Figure 7b show the tunnels from each scenario plotted together with the validation data.



(a) (b)

Figure 7. Data-enclosing tunnel and validation data: (a) SC1 (b) SC2.

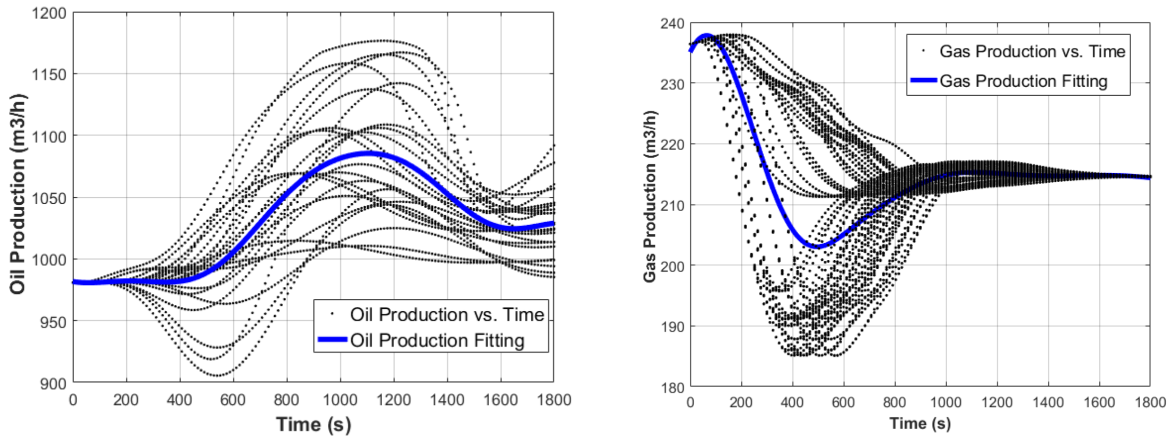
Nevertheless, with the aim of having different benchmarking points, we developed a method more straightforward than the enclosing tunnel. We created a data enclosing band, which evaluates separately each studied variable, without dimensionality reduction. The main idea was to develop a simpler method that could be executed faster and with lower efforts, so that the performance of the data-enclosing tunnel, which is a more complex method, could be compared to a simpler one.

The idea of an enclosing band is also known as confidence band. Two different implementations can be found in Skelton and Willms (2014) and Lee and Hyun (2011).

The construction of the band consists in choosing or defining a reference path from the good execution paths. Once a reference path is established, the data-enclosing band is created by setting a limit above and below the reference path. The enclosing band was generated three times, each one with a different and simpler approach than the previous one. All of them were compared with the tunnel. Each of the three approaches for developing the enclosing band is explained in the following.

3.8.1 Data Enclosing Band: Approach 1 (DEB1)

1. Reference path: it was defined by running a curve fitting procedure for each of the studied variables. The curve fittings were run using only the good execution paths of the training data, as only the good execution paths were used to build the enclosing tunnel. Figure 8a and Figure 8b show the curve fitting for the variables oil production and gas production of SC1 and SC2, respectively.



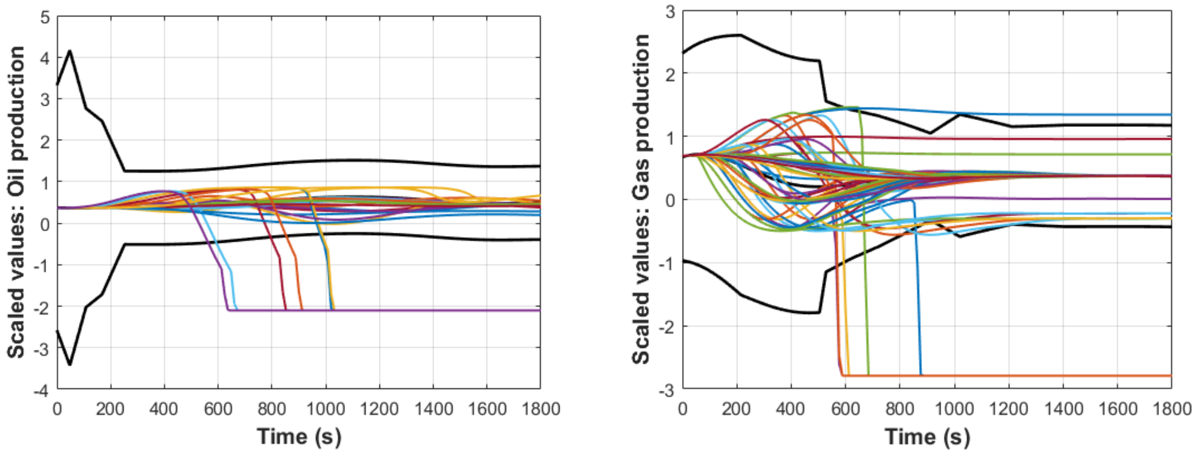
(a) (b)

Figure 8. (a) SC1 – DEB1 – Curve fitting of the variable oil production / (b) SC2 – DEB1 – Curve fitting of the variable gas production.

2. Data scaling: the training data was grouped per variables, one matrix for each variable. Given that in both of our case studies seven variables were monitored, there were seven matrices with as many columns as samples in the training data of each case study. The mean values

and the standard deviations of each of the matrices were calculated. These parameters were used later to scale the reference path and the validation data.

3. Enclosing band: after establishing the reference path and scaling, the following step was to design the enclosing band. In this approach, the band was created by summing up and subtracting from the scaled reference path the radiuses of the tunnel. Figure 9a and Figure 9b show the enclosing band together with the scaled validation data of SC1 and SC2, respectively. Figure 9a corresponds to the scaled variable, oil production, of SC1. Figure 9b corresponds to the scaled variable, gas production, of SC2.

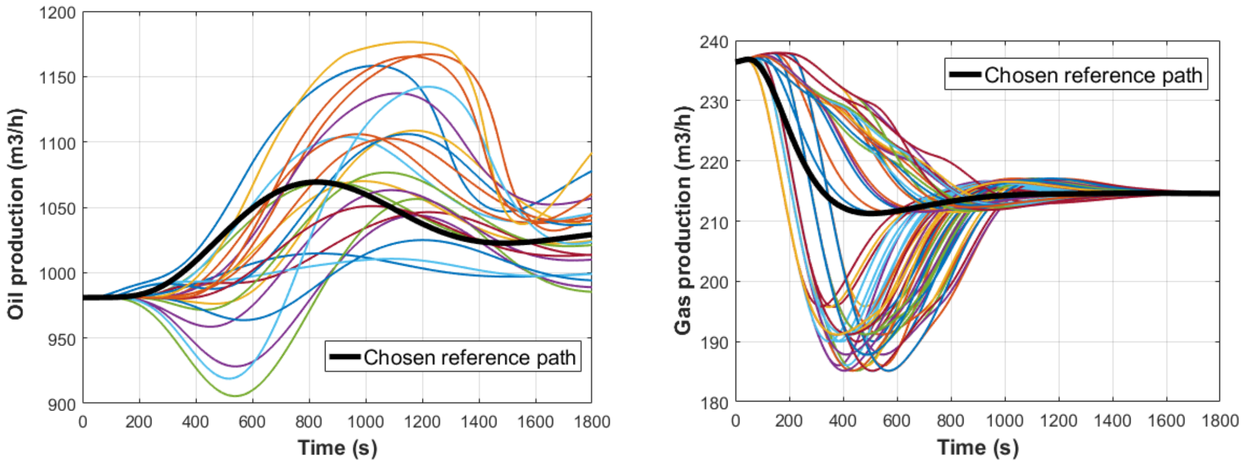


(a) (b)

Figure 9. Data enclosing band and validation data (a) SC1 – DEB1 – Variable: Scaled oil production / (b) SC2 – DEB1 –Variable: Scaled gas production.

3.8.2 Data Enclosing Band: Approach 2 (DEB2)

1. Reference path: it was chosen from the good execution paths of the training data. The reference was selected by observing the execution paths of one variable only. The observed variable was the one that represents better the achievement of the scenario objective. The variable observed in SC1 was the oil production, while for SC2 it was the gas production. Figure 10a and Figure 10b show the reference paths for SC1 and SC2, respectively.

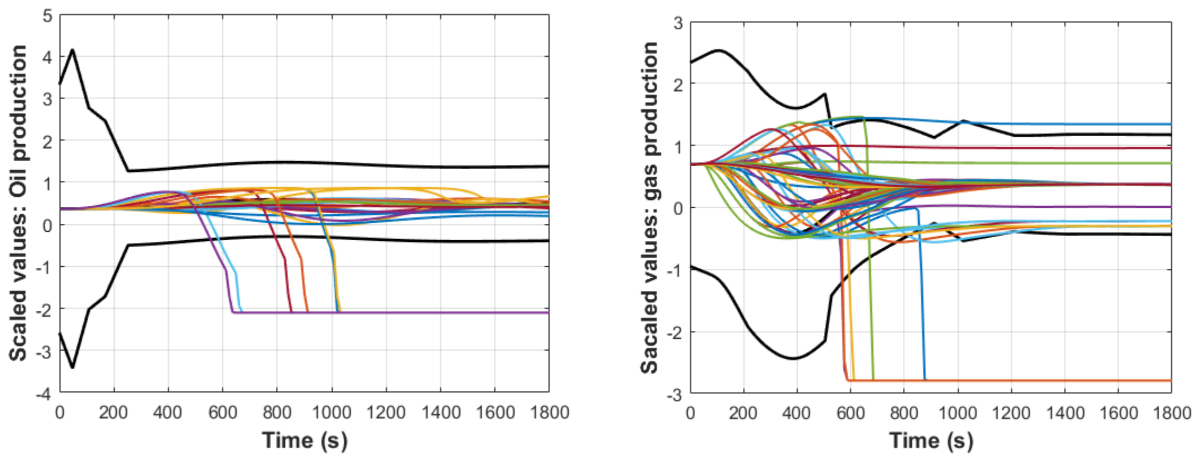


(a) (b)

Figure 10. (a) SC1 – DEB2 – Chosen reference path among the oil production paths / (b) SC2 – DEB2 – Chosen reference path among the gas production paths.

2. Data scaling: it was done as in DEB1.

3. Enclosing band: after choosing the reference path and scaling, the enclosing band was created by summing up and subtracting from the scaled reference path the radiuses of the tunnel. Figure 11a and Figure 11b show the enclosing band together with the scaled validation data of SC1 and SC2, respectively. It can be noticed that the results with DEB1 and DEB2 seem to be very similar, even though the reference paths were established differently.



(a) (b)

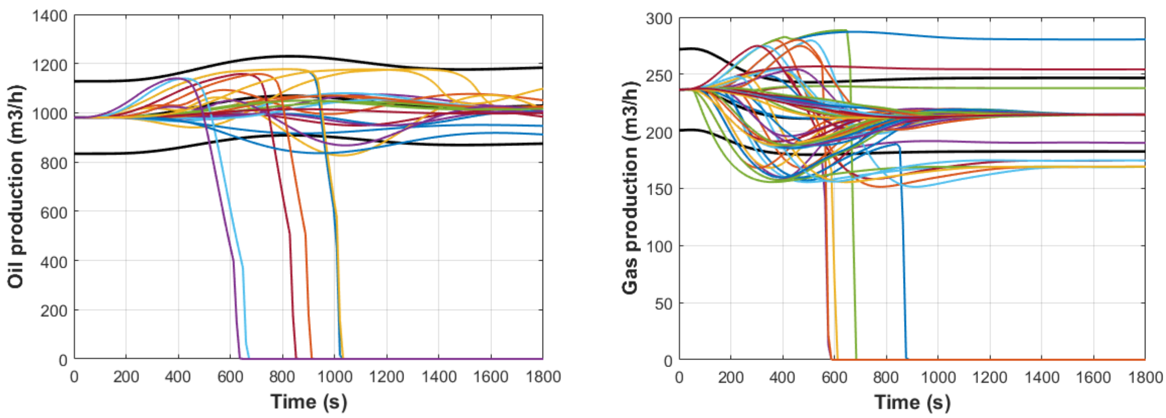
Figure 11. Enclosing band and validation data (a) SC1 – DEB2 – Variable: Scaled oil production / (b) SC2 – DEB2 –Variable: Scaled gas production.

3.8.3 Data Enclosing Band: Approach 3 (DEB3)

1. Reference path: it is the same as in DEB2 (see Figure 10a and Figure 10b) for the chosen reference path.

2. Data scaling: the data was not scaled.

3. Enclosing band: it consists in creating the enclosing band using a generic factor. The factor was calculated by assuming that the tunnel radiuses were scaled data. The radiuses were transformed into "actual variables" using the scaling parameters determined in the previous two approaches. Once the radiuses were converted into their version of each of the seven variables, the resulting matrix was compared with the chosen reference path to determine the relationship between them. By doing so, a factor was calculated for each of the two training scenarios. The average between the two factors was taken to get a final generic value, which was 15 %. The enclosing band was created by summing up and subtracting 15 % from the reference path. Figure 12a and Figure 12b show the enclosing band together with the validation data of SC1 and SC2, respectively.



(a) (b)

Figure 12. Enclosing band and validation data (a) SC1 – DEB3 – Variable: Oil production / (b) SC2 – DEB3 – Variable: Gas production.

3.8.4 Validation of the enclosing bands

For each of the case studies, there were seven bands, one for each of the monitored variables. Therefore, to validate the enclosing band, first, the percentage of residence of each variable path within its corresponding band was calculated. The validation of the bands is also based on the metrics established for the enclosing tunnel (see Section 2.8). If any of the variables falls outside of its associated band more than 35 or 50 % of the total time, the execution path related to such variable is classified as bad. Next, the validation of the enclosing band follows the same way as the tunnel one. Based on the known labels of the validation data, i.e. knowing which of the paths are good and which ones bad, the enclosing band is validated by calculating how many of the execution paths in the validation set ended correctly inside or outside the band.

3.8.5 Comparison of the methods

Table 3 and Table 4 present the different accuracies obtained for each of the methods studied. We consider four subgroups of classification: 1) True Positives (TPs), which denote the good execution paths that fall inside the tunnel/band; 2) True Negatives (TNs), which denote bad execution paths that fall outside the tunnel/band; 3) False Positives (FPs), which refer to bad execution paths that fall inside the tunnel/band; and 4) False Negatives (FNs), which refer to good execution path that fall outside the tunnel/band. The accuracy is defined as follows:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Table 3 reports the accuracy of SC1 that is the same for all the methods when Metric 1 is used (a path is considered bad if it falls outside the tunnel/band more than 35 % of the total time). In case of Metric 2 (a path is considered bad if it falls outside the tunnel/band more than 50 % of the total time), DEB1 and DEB2 have a lower accuracy while the accuracy of the enclosing tunnel and DEB3 remain the same.

The results of SC2 are notoriously different from those of SC1. When it comes to SC2, Table 4 shows that the tunnel method is the most accurate regardless of the implemented metric.

Table 3. Comparison of the accuracy of the methods for SC1.

Method	Metric 1: 35 % outside is "bad"					Metric 2: 50 % outside is "very bad"				
	FP	FN	TP	TN	Acc. (%)	FP	FN	TP	TN	Acc. (%)
SC1 Tunnel	3	0	12	10	88	3	0	12	10	88
SC1 DEB1	3	0	12	10	88	5	0	12	8	80
SC1 DEB2	3	0	12	10	88	6	0	12	7	76
SC1 DEB3	0	3	9	13	88	0	3	9	13	88

Table 4. Comparison of the accuracy of the methods for SC2.

Method	Metric 1: 35 % outside is "bad"					Metric 2: 50 % outside is "very bad"				
	FP	FN	TP	TN	Acc. (%)	FP	FN	TP	TN	Acc. (%)
SC2Tunnel	4	0	35	31	94.3	10	0	35	25	85.7
SC2 DEB1	13	0	35	22	81.4	18	0	35	17	74.3
SC2 DEB2	9	17	18	26	62.9	15	0	35	20	78.6
SC2 DEB3	21	0	35	14	70.0	21	0	35	14	70.0

The subgroups of classification can also be analyzed with a confusion matrix. A confusion matrix is a table that describes the performance of a classification method on a set of test data for which the true values are known (Data School, 2014). Figure 13 shows how to read a confusion matrix. The values in the diagonal (green boxes) are the correct classifications, i.e. the true positives and the true negatives. The final values in the diagonal (yellow box), correspond to the overall correct classifications, i.e. the accuracy, and the overall incorrect classifications. The values outside the diagonal (red boxes) correspond to misclassifications, i.e. false positives and false negatives. Reading the confusion matrix vertically, the results presented in the last row of the first column refer only to the actual number of good execution paths. One can observe both the percentage of good execution paths that were classified correctly and the percentage of good execution paths that were misclassified. The same is true for the last row of the second column, but this case refers only to the actual number of bad execution paths. By reading the confusion matrix horizontally, the values shown in the last column of the first row refer to the total amount of predicted positives. One can observe the percentage of correct and incorrect positives. With reference to the last column of the second row, these values refer to the total amount of predicted negatives. One can observe the percentage of correct and incorrect negatives. Figures 14, 15, 16, and 17 respectively show the confusion matrix of each of the methods using Metric 1, for SC2.

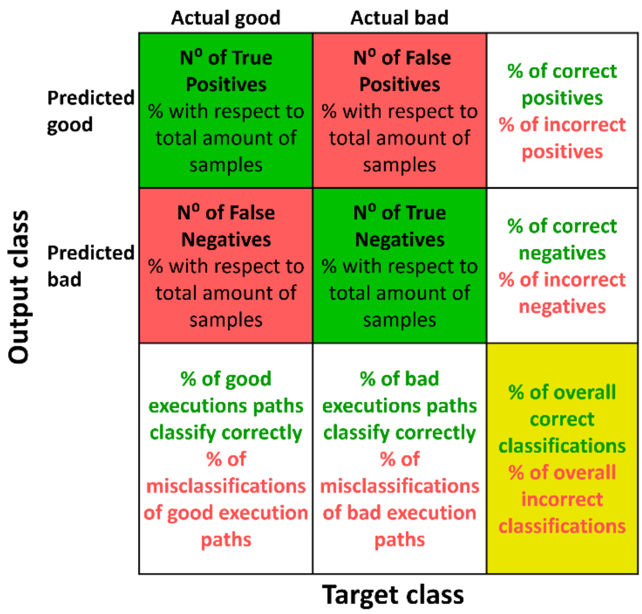


Figure 13. Explanation of the confusion matrix.

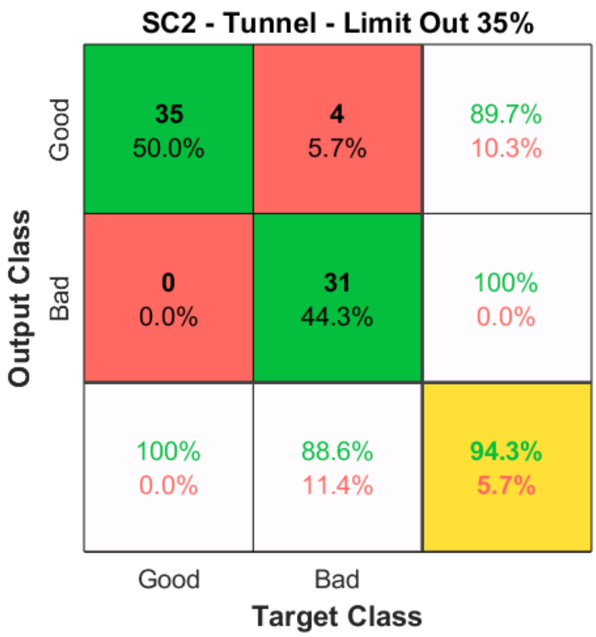


Figure 14. SC2 – Confusion matrix of the data-enclosing tunnel using Metric 1.

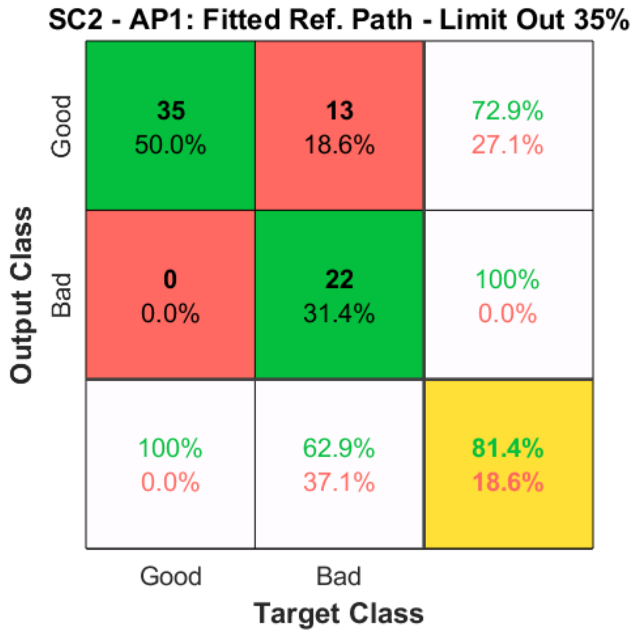


Figure 15. SC2 – Confusion matrix DEB1 using Metric 1.

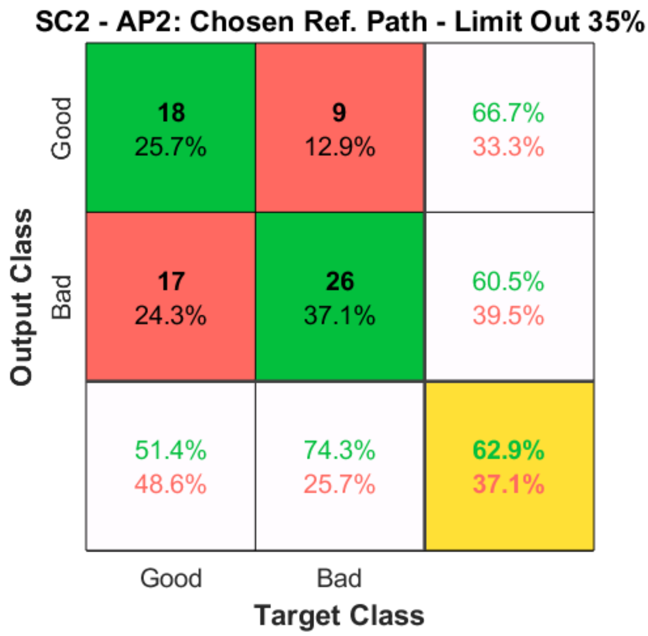


Figure 16. SC2 – Confusion matrix DEB2 using Metric 1.

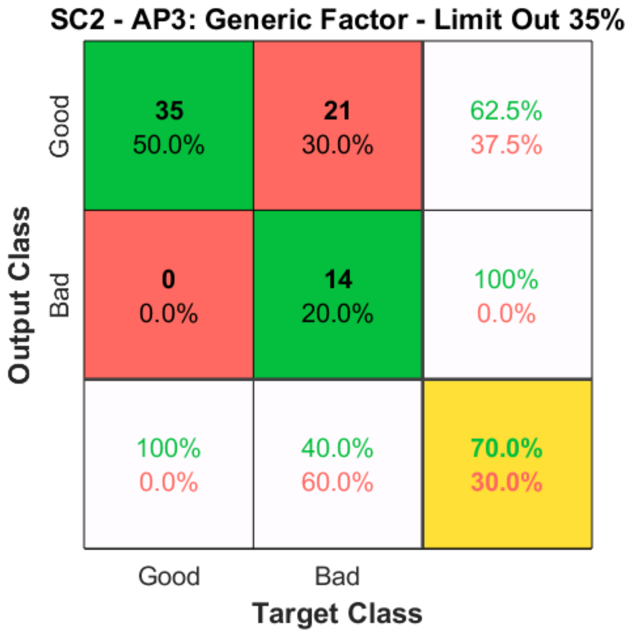


Figure 17. SC2 – Confusion matrix DEB3 using Metric 1.

4 Discussion

This work presented a methodology for constructing a data-enclosing tunnel to be used as an online feedback tool for simulator-training scenarios. Also, two case studies of the proposed methodology were developed. A data-enclosing tunnel was built for two different training scenarios to demonstrate the usefulness and viability of the methodology presented and to validate it.

Given that we did not have available actual user data, the data was generated with an algorithm. The SC1 data was not labeled as soon the data was generated. Therefore, a clustering method was used. On the other hand, the data used for the second scenario was labeled as soon as it was generated. Moreover, the data generated for the second scenario was larger than the first one.

To validate the tunnels built for each of the training scenarios, we determined how many of the execution paths in the validation data ended correctly either inside or outside the tunnel based on the data labels. Besides, we developed a simpler method, an enclosing band. The enclosing band aims to compare our data-enclosing tunnel method with another that represents the state of the art. There is not much research related to online feedback for simulator training based on the evaluation of good execution paths, something similar to it can be found in (Alamehtä, 2018). We developed a simpler method that works in a 2D plane without dimensionality reduction, which means that all the variables are studied individually.

Table 3 shows the accuracy results obtained for SC1. In case of Metric 1, the same accuracy, 88 %, was obtained with all the methods studied. With Metric 2, the accuracy of DEB1 and DEB2 was lower. This can be explained by observing the number of false positives, which increases in both cases with Metric 2. Given that Metric 2 has a higher tolerance towards the time an execution path can fall outside the tunnel/band without being considered bad, some execution paths get misclassified. The tunnel and DEB3 have the same accuracy using both metrics. The lack of variation in the accuracy of the methods for SC1 can be due to the size of the data, which could be considered small, given that it had only 50 samples for training and 25 for validation. Consequently, SC1 is a rather simple problem, and the information that is possible to get from the data is obtained with all the tested methods.

On the other hand, the accuracy results achieved for SC2 are more versatile (see Table 4). The results obtained with Metric 1 show that the data-enclosing tunnel is the most accurate among the four methods. In addition, DEB1 is more accurate than DEB2. We can argue that the accuracy of DEB1 is higher since the reference path used to create the enclosing band was determined more meticulously than for DEB2. A curve fitting represents better the general behavior of many curves (DEB1) than only one curve chosen randomly from the lot (DEB2). DEB2 and DEB3 are the less complex of the four methods. Indeed, the accuracies obtained with these methods are the lowest.

With reference to the results obtained with Metric 2 (Table 4), once again the data-enclosing tunnel is the most accurate of the four methods. However, the accuracy of the tunnel with Metric 2 decreases. With reference to the number of false positives, it is possible to observe that a more flexible metric for SC2 leads to a larger number of misclassifications of the bad paths. The same happens with DEB1 (Table 4). On the contrary, the accuracy of DEB2 increases when Metric 2 is used, which indicates that having a more flexible metric for this case helps classifying correctly those execution paths that with Metric 1 did not fall within the right category, i.e. the number of true positives of DEB2 increases when using Metric 2. In case of DEB3, the accuracy remains the same with any metrics, which was also the case for SC1. This can be ascribed to the simplicity of DEB3 that does not allow achieving differences in the accuracy of the method when varying the metric.

The variety of the results obtained for SC2 may also be due to the size of the data, which in this case is larger than the one for SC1, having 130 samples for training and 70 for validation. In general, based on the results with SC2 which have a notorious variability, it is worth observing that the tunnel is the most accurate of all the investigated methods based on any of the two metrics, with DEB3 being the less accurate.

It was noticed that the number of execution paths to study has an important impact on the results. The higher the number of samples, the more variations can be observed in the performance of the different tested methods. By increasing the number of samples, it is possible observing that the data-enclosing tunnel is the most robust of all the methods studied. DEB1, which is based on curve fitting, is the second more consistent method. Hence, one can argue that the more elaborate the technique, the better the accuracy results.

The monitored variables play an essential role in the implementation of each of the proposed methods. It is crucial to select the most relevant variables that can build a clear view of the process status and of the scenario objectives, to be able to construct a data-enclosing tunnel/band that will make an accurate evaluation of the data and consequently, effective feedback can be delivered to the trainees.

Finally, labelling the data as soon it is generated has a significant influence on the amount of work needed to implement the proposed methods. If the data is tagged right away, this facilitates the workflow for the data classification. It is highly recommended for those working with simulator training, to save the trainees' records and add a description of their performance so that in the future it will be easier to handle that data.

5 Conclusions

The methodology presented in this work was effectively implemented in two case studies. We demonstrated how to use the methodology and how to follow each of the related steps with an application to two industrial cases, which were developed with the dynamic process simulator K-Spice, from Kongsberg Digital. We presented the data mining results from each of the scenarios: classification, processing, and dimensionality reduction of the data. Further, different situations that the user might encounter when using the methodology were illustrated, as well as how to deal with such conditions as non-labeled data from the beginning or not available data from actual users of the simulator.

The two data-enclosing tunnels developed for each of the case studies were validated and compared with three other simpler methods. It was noticed that the size of the data had a significant influence on the accuracy of the methods.

When executing the data mining process, the larger the data the larger amount of information that can be extracted from it and more variability can be observed among the results. The complexity of the methods also has a significant influence on their accuracy. The most elaborate and complex methods had more substantial accuracy than the simplest ones. This means that the data-enclosing tunnel is the most accurate of all the methods evaluated, which indicates that the tunnel is the method that could detect more efficiently if a trainee deviates from the good execution paths.

On the other hand, even though less accurate, the simplest approach also has some advantages. As long as there are not so many variables to be evaluated individually (in our case studies we had seven) when it refers to reaction time, the simplest method (DEB3) would be the fastest in detecting when the trainee is deviating from the good execution path, given that the data neither needs to be reduced nor scaled. However, since the simplest method is less accurate, using it encompasses the risk of not giving any feedback to the trainee when they are taking wrong actions. Further, as mentioned above, it is also advisable to consider the number of monitored variables. Complex processes require a large number of variables to be monitored, the larger the number of variables, the longer the time that will be needed to determine if they do not fall inside the established limits of the enclosing band. This would not be the case of the data-enclosing tunnel, given that it has the advantage of dimensionality reduction. Nevertheless, further work needs to be done to evaluate and corroborate these hypotheses.

Moreover, future work also includes the development and testing of a user interface for the deployment of an automated feedback tool. The interface should show guiding messages using natural language so that the trainee does not have to read a number of values on the screen. The testing of the tool should be carried out with actual trainees that could provide their opinion on their user experience.

Acknowledgments

The authors (LM, AY, TK) would like to thank the research funder, Oslo Metropolitan University, Faculty of Technology, Art and Design.

References

ALAMEHTÄ, A. 2018. *Automatic plant operator performance evaluation in a large scale operator training simulator.* Master's, University of Oulu.

BAKER, R. S. J. D. 2010. Data Mining. *In:* PETERSON, P., BAKER, E. & MCGAW, B. (eds.) *International Encyclopedia of Education.* 3rd ed. Oxford: Elsevier.

BELL, B. S., KANAR, A. M. & KOZLOWSKI, S. W. J. 2008. Current issues and future directions in simulation-based training in North America. *The International Journal of Human Resource Management,* 19**,** 1416-1434.

DATA SCHOOL. 2014. *Simple guide to confusion matrix terminology* [Online]. Available: https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/ [Accessed March 19 2019].

FUMAROLA, F., CIAMPI, A., APPICE, A. & MALERBA, D. A Sliding Window Algorithm for Relational Frequent Patterns Mining from Data Streams. *In:* GAMA, J., COSTA, V. S., JORGE, A. M. & BRAZDIL, P. B., eds. Discovery Science, 2009// 2009 Berlin, Heidelberg. Springer Berlin Heidelberg, 385-392.

GHOSH, K., RAMTEKE, M. & SRINIVASAN, R. 2014. Optimal variable selection for effective statistical process monitoring. *Computers & Chemical Engineering,* 60**,** 260-276.

GOLDBERG, B. & CANNON-BOWERS, J. 2015. Feedback source modality effects on training outcomes in a serious game: Pedagogical agents make a difference. *Computers in Human Behavior,* 52**,** 1-11.

GONZALEZ-SANCHEZ, J., CHAVEZ-ECHEAGARAY, M. E., VANLEHN, K., BURLESON, W., GIRARD, S., HIDALGO-PONTET, Y. & ZHANG, L. A system architecture for affective meta intelligent tutoring systems. International Conference on Intelligent Tutoring Systems, 2014. Springer, 529-534.

KLUGE, A., SAUER, J., SCHÜLER, K. & BURKOLTER, D. 2009. Designing training for process control simulators: a review of empirical findings and current practices. *Theoretical Issues in Ergonomics Science,* 10**,** 489-509.

KOMULAINEN, T. M. & LØVMO, T. 2014. Large-Scale Training Simulators for Industry and Academia. *Proceedings of the 55th Conference on Simulation and Modelling (SIMS 55), Modelling, Simulation and Optimization, 21-22 October 2014, Aalborg, Denmark.* Aalborg, Denmark: Linköping University Electronic Press, Linköpings universitet.

KOMULAINEN, T. M. & SANNERUD, A. R. 2018. Learning transfer through industrial simulator training: Petroleum industry case. *Cogent Education,* 5**,** 1-19.

KONGSBERG. 2009. *K-Spice: A new and powerful dynamic process simulation tool* [Online]. Available: https://www.kongsberg.com/en/kongsberg-digital/news/2009/june/0625_kpice/ [Accessed March 06 2019].

LEE, J. & HYUN, S. 2011. Confidence bands for the difference of two survival functions under the additive risk model. *Journal of Applied Statistics,* 38**,** 785-797.

LEE, S., JEONG, I. & IL, M. 2000. Development of evaluation algorithms for operator training system. *Computers & Chemical Engineering,* 24**,** 1517-1522.

MAHMOUD, M. H. & ABO EL-HAMAYED, S. H. 2016. An intelligent tutoring system for teaching the grammar of the Arabic language. *Journal of Electrical Systems and Information Technology,* 3**,** 282-294.

MALAKIS, S. & KONTOGIANNIS, T. 2012. Refresher Training for Air Traffic Controllers: Is It Adequate to Meet the Challenges of Emergencies and Abnormal Situations? *The International Journal of Aviation Psychology,* 22**,** 59-77.

MANCA, D., NAZIR, S., COLOMBOA, S. & KLUGEB, A. 2014. Procedure for Automated Assessment of Industrial Operators. *CHEMICAL ENGINEERING,* 36**,** 391-396.

MANCA, D., NAZIR, S., LUCERNONI, F. & COLOMBO, S. 2012. Performance Indicators for the Assessment of Industrial Operators. *In:* IAN DAVID LOCKHART, B. & MICHAEL, F. (eds.) *Computer Aided Chemical Engineering.* Elsevier.

MARCANO, L., HAUGEN, F. A., SANNERUD, R. & KOMULAINEN, T. 2019. Review of simulator training practices for industrial operators: How can individual simulator training be enabled? *Safety Science,* 115**,** 414-424.

MARCANO, L. & KOMULAINEN, T. Constructive Assessment Method for Simulator Training. Proceedings of The 9th EUROSIM Congress on Modelling and Simulation, EUROSIM 2016, The 57th SIMS Conference on Simulation and Modelling SIMS 2016, 2018. Linköping University Electronic Press, 395-402.

MARCANO, L., KOMULAINEN, T. M. & HAUGEN, F. A. 2017. Implementation of performance indicators for automatic assessment. *In:* ANTONIO ESPUÑA, M. G. & LUIS, P. (eds.) *Computer Aided Chemical Engineering.* Elsevier.

MARCANO, L., YAZIDI, A., MANCA, D. & KOMULAINEN, T. M. Using the concept of data enclosing tunnel as an online feedback tool for simulator training. The 59th Conference on Simulation and Modelling (SIMS 59), 2018 Oslo Metropolitan University, Norway. Linköping University Electronic Press, 132-139.

MARKET STUDY REPORT, L. 2019. *Operator Training Simulator Market By Component, By Simulation Environment, Forecast, 2019 - 2025,* [Online]. Available: https://www.marketstudyreport.com/reports/operator-training-simulator-market [Accessed].

MOHAMED, H. & LAMIA, M. 2018. Implementing flipped classroom that used an intelligent tutoring system into learning process. *Computers & Education,* 124**,** 62-76.

NAZIR, S. & MANCA, D. 2015. How a plant simulator can improve industrial safety. *Process Safety Progress,* 34**,** 237-243.

RHIENMORA, P., HADDAWY, P., SUEBNUKARN, S. & DAILEY, M. N. 2011. Intelligent dental training simulator with objective skill assessment and feedback. *Artificial Intelligence in Medicine,* 52**,** 115-121.

SALAS, E., TANNENBAUM, S. I., KRAIGER, K. & SMITH-JENTSCH, K. A. 2012. The Science of Training and Development in Organizations: What Matters in Practice. *Psychological Science in the Public Interest,* 13**,** 74-101.

SEBZALLI, Y. M., LI, R. F., CHEN, F. Z. & WANG, X. Z. 2000. Knowledge discovery from process operational data for assessment and monitoring of operator's performance. *Computers & Chemical Engineering,* 24**,** 409-414.

SEWELL, C., MORRIS, D., BLEVINS, N. H., DUTTA, S., AGRAWAL, S., BARBAGLI, F. & SALISBURY, K. 2008. Providing metrics and performance feedback in a surgical simulator. *Computer Aided Surgery,* 13**,** 63-81.

SHU, Y., MING, L., CHENG, F., ZHANG, Z. & ZHAO, J. 2016. Abnormal situation management: Challenges and opportunities in the big data era. *Computers & Chemical Engineering,* 91**,** 104-113.

SKELTON, A. & WILLMS, A. R. 2014. An algorithm for continuous piecewise linear bounding of discrete time series data. *BIT Numerical Mathematics,* 54**,** 1155-1169.

SPESHILOV, K. & KHABAROV, V. 2017. Creating a Model of an Operator of a Simulator Complex using Commonsense Reasoning. *Procedia Engineering,* 178**,** 394-400.

WEISSTEIN, E. W. 2018. *Minimal Enclosing Circle* [Online]. Available: http://mathworld.wolfram.com/MinimalEnclosingCircle.html [Accessed 24 April 2018].

YAMASHITA, Y. 2000. Supervised learning for the analysis of process operational data. *Computers & Chemical Engineering,* 24**,** 471-474.