Institution of MECHANICAL ENGINEERS
Journal of RISK AND RELIABILITY

Proc IMechE Part O:
J Risk and Reliability
2018, Vol. 232(3) 320–334

DOI: 10.1177/1748006X17694494
journals.sagepub.com/home/pio
SAGE

# Formal reliability analysis of oil and gas pipelines

**Waqar Ahmad[1], Osman Hasan[1], Sofiène Tahar[2] and Mohamed Salah Hamdi[3]**

**Abstract**

Depending on the operational environment, installation location, and aging of oil and gas pipelines, they are subject to various degradation mechanisms, such as cracking, corrosion, leaking, and thinning of the pipeline walls. Failure of oil and gas pipelines due to these degradation mechanisms can lead to catastrophic events, which, in the worst case, may result in the loss of human lives and huge financial losses. Traditionally, paper-and-pencil proof methods and Monte Carlo based computer simulations are used in the reliability analysis of oil and gas pipelines to identify potential threats and thus avoid unwanted failures. However, paper-and-pencil proof methods are prone to human error, especially when dealing with large systems, while simulation techniques primarily involve sampling-based methods, i.e., not all possible scenarios of the given systems are tested, which compromises the accuracy of the results. As an accurate alternative, we propose to use a higher-order-logic theorem proving for the reliability analysis of oil and gas pipelines. In particular, this paper presents the higher-order-logic formalization of commonly used reliability block diagrams (RBDs), such as series, parallel, series–parallel, and *k*-out-of-*n*, and provides an approach to utilize these formalized RBDs to assess the reliability of oil and gas pipelines. For illustration, we present a formal reliability analysis of a pipeline transportation subsystem used between the oil terminals at the Port of Gdynia, Poland, and Dębogórze.

**Keywords**

Reliability block diagrams (RBDs), higher-order-logic theorem proving, probability theory, oil and gas pipelines

Date received: 10 February 2016; accepted: 20 January 2017

## Introduction

Oil and gas pipelines are indeed the most pivotal part of the present-age energy-delivery system[1] and thus one of the foremost requirement of oil and gas industries and their supply chain is to ensure that the pipelines continue to function free of risk. Oil and gas pipelines are known for their susceptibility to leaks and catching fire, which may lead to an explosion and thus may be responsible for a catastrophic event. For instance, a big explosion was caused by the methane gas leakage,[2] on 20 April 2010, at the Deepwater Horizon oil rig operated by Transocean, which is a subcontractor of British Petroleum. Owing to this incident, which was caused by the loss of the platform's well control system, 11 workers died instantly and the rig also sank and was completely destroyed, causing millions of gallons of oil to spill out into the Gulf of Mexico. This is considered one of the largest accidental marine oil spills in the history of the petroleum industry and, even now, it continues to damage the marine and wildlife habitats, as well as the Gulf's fishing and tourism industries.[2] There are tens of thousands of miles in length of oil and gas pipelines around the world; these are becoming increasingly susceptible to failures owing to aging. Hence, rigorous reliability and failure analysis of oil and gas pipelines is very important to minimize the chances of disasters, such as the British Petroleum disaster.

Pipeline integrity management[3] is a process to evaluate the risks posed to pipelines, with the objective of improving their reliability. According to the US Pipeline Safety Improvement Act of 2002,[4] natural gas

[1]School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Pakistan
[2]Department of Electrical and Computer Engineering, Concordia University, Canada
[3]Information Systems Department, Ahmed Bin Mohammed Military College, Qatar

**Corresponding author:**
Waqar Ahmad, School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Sector H-12, Islamabad 44000, Pakistan.
Email: waqar.ahmad@seecs.nust.edu.pk

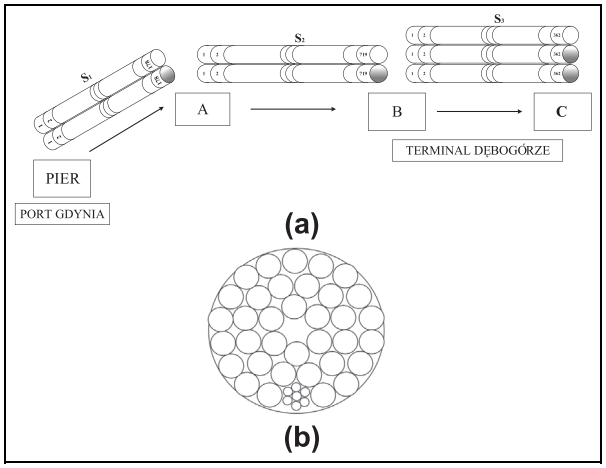


**Figure 1.** Pipeline systems: (a) RBD of pipelines at the oil terminal Dębogórze;[9] (b) cross-section of M-80-200-10 steel rope.[6]

transmission companies must conduct risk analysis of pipeline segments at high consequence areas, i.e., areas where a failure in the gas pipeline would have a significant impact on public safety or the environment.

Reliability block diagram (RBD),[5] which is a graphical technique to analyze the impact of the individual reliabilities of system components on the overall reliability of the system, provides an efficient tool for conducting pipeline integrity management. This pipeline reliability analysis involves three major steps:[6]

(a) partitioning the given pipeline into segments and constructing its equivalent RBD;[6]
(b) assessing the reliability of the individual segments;
(c) evaluating the reliability of the entire pipeline based on the RBD and the reliability of its individual segments.

The reliability of the individual pipeline segments is usually expressed in terms of their failure rates $\lambda$ and a time-to-failure random variable, such as an exponential[7] or Weibull distribution.[6] A single oil or gas pipeline can be modeled using series RBD configurations[8] but complex pipeline networks can be modeled using a combination of series and parallel RBD configurations.[9] For example, Figure 1 depicts some portions of a real-world pipeline system. Figure 1(a) presents the RBD of the oil terminal pipeline network at Dębogórze.[9] Based on the redundancy structure and operational state of this pipeline, its reliability can be evaluated using series–parallel and $k$-out-of-$n$ RBDs.[9] The three-stranded steel rope of M-80-200-10 illustrated in Figure 1(b), is composed of 36 strands: 18 outer, 12 inner, and 6 more inner strands. The reliability of this steel rope can be evaluated by considering it as a parallel system and utilizing a parallel RBD configuration.[6]

The aforementioned three-step process commences by gathering data from in-line inspection tools to detect cracks, corrosion, or damage[10,11] and selecting suitable failure models for the individual segments of the pipeline. The gathered data, and the failure model are then analyzed, using either paper-and-pencil methods or computer simulations to assess the failure probability or reliability of the complete oil and gas pipelines.[7,9] However, the paper-and-pencil based analytical methods are prone to human error when it comes to the analysis of large systems, such as the transmission network of oil and gas pipelines. It commonly occurs that many key assumptions, which are in the minds of mathematicians, are not properly documented; hence, not all reliability constraints can be entirely passed to the reliability assurance engineers. These undocumented assumptions or constraints might become the primary cause of catastrophic events. Conversely, numerous commercially available software tools, such as DNV-GL[12] and ReliaSoft,[13] are available for the reliability assessment of oil and gas pipelines. These tools are mainly based on Monte Carlo simulation, which is a sampling-based method, since exhaustive testing for all possible scenarios is not computationally feasible, given the large size of the system models and the involvement of so many variables of continuous nature in the reliability analysis of oil and gas pipelines. Thus, just like paper-and-pencil proof methods, the analysis results of computer simulations cannot be completely trusted, since there is always some risk of missing a corner case from the test vectors used for the simulation. These analysis inaccuracies are a severe limitation in the case of oil and gas pipelines, as an uncaught system bug might endanger human and animal lives or lead to a significant financial loss.

Formal methods,[14] which are computer-based mathematical reasoning techniques, have been used to overcome the inaccuracy limitations of paper-and-pencil proof methods and simulation and can thus be used to play a vital role in the development of dependable systems.[15] The main process behind the formal analysis of a system is first to construct a mathematical model of the given system using a state machine or an appropriate logic and then to use logical reasoning and deduction methods to verify formally that the system exhibits the desired characteristics, which are also specified mathematically using an appropriate logic.

Formal methods are mainly categorized into two mainstream techniques: model checking[16] and theorem proving.[17] Model checking is a state-based technique, in which system behavior, specified as a state machine, is analyzed by verifying the temporal properties exhaustively over the entire state-space of the formal model of the given system using a computer, while theorem proving enables the use of logical reasoning to verify relationships between a system and its properties as theorems, specified in an appropriate logic, using a computer. Both model checking and theorem proving have been used for the probabilistic analysis of systems,[18–20] which is the foremost requirement for conducting reliability analysis. However, owing to the

state-based nature of model checking, this method suits Markov-chain-based reliability analysis quite well, whereas, in the context of RBD-based reliability analysis, model checking can be used to analyze the properties of dynamic RBDs only.[21] Conversely, theorem proving based on expressive higher-order-logic (HOL),[22] which is a system of deduction with a precise semantics and can be used for the formal modeling of any system that can be described mathematically, enables a variety of data types, such as lists and real numbers, to be utilized and can be used to verify generic mathematical expressions. Thus, leveraging on the probability theory formalization in HOL,[23] the theorem proving technique has the potential to provide an accurate and rigorous alternative for the reliability analysis of oil and gas pipelines.

Theorem proving has been recently used for the formalization of different types of RBD, such as series[8], parallel[24], parallel–series[24] and series–parallel,[25] to conduct formal reliability analysis of many applications, including simple oil and gas pipelines with serial components,[8] wireless sensor network protocols,[24] and logistic supply chains.[25] However, many real-world pipelines[9] have some operational states that require only some of their redundant subsystems to be operational. These kinds of pipeline operational behavior have been modeled using a $k$-out-of-$n$ RBD configuration. Thus, we need to formalize the $k$-out-of-$n$ RBD configurations along with other RBD configurations to facilitate the accurate analysis of a wide range of oil and gas pipelines.

The main novel contributions of the paper are:

- an improved formalization approach for commonly used RBDs, in particular series–parallel RBD, which is much more compositional in nature than existing formalizations of RBDs[24,25] and can be easily extended to model any kind of complex RBDs within the HOL theorem prover;
- formalization of a $k$-out-of-$n$ RBD configuration and its variants, which is essentially required along with other RBD configurations to facilitate the accurate reliability analysis of a wide range of real-world systems, including oil and gas pipeline networks;[9]
- formal reliability analysis of a complex pipeline network used in the oil terminal in Dębogórze,[9] which is designated for the reception of oil products from ships and the storage and transportation of oil products by carriages or cars.

The rest of the paper is organized as follows. The next section presents a review of the related work. Then follows an overview of the proposed methodology, which has been used to conduct formal reliability analysis of oil and gas pipeline system. To facilitate the understanding of the paper for non-experts in theorem proving, we then present a brief introduction of theorem proving, the HOL theorem prover, and the HOL probability theory formalization. This is followed by the description of our formalization of the RBD configurations. The RBD-based formal reliability analysis of oil and gas pipelines is presented before conclusions are drawn and discussions on future work made.

## Related work

Many computer software tools, such as DNV-GL,[12] ReliaSoft,[13] and the ASENT Reliability analysis tool,[26] support RBD-based reliability analysis and provide powerful graphical editors, which can be used to construct RBD models of oil and gas pipelines. These tools generate samples from exponential or Weibull random variables to model the reliabilities of the system components. These samples are then processed using computer arithmetic and numerical techniques to compute the reliability of the complete system. Although these software tools provide more scalable and faster analysis than paper-and-pencil based analytical methods, they cannot ascertain the absolute correctness of the system because of their inherent sampling-based nature, the involvement of pseudorandom numbers, and numerical methods.

Formal methods, such as Petri nets, have also been used to model RBDs,[27] as well as dynamic RBDs,[21] which are used to describe the reliability behavior of systems. Petri net verification tools, based on model checking principles, are then used to verify behavioral properties of the RBD models to identify design flaws.[21,27] Similarly, the probabilistic model checker, PRISM,[28] has been used for the quantitative verification of various safety and mission-critical systems, such as failure analysis for an industrial product development workflow,[29] an airbag system,[30] and the reliability analysis of a global navigation satellite system that enables an aircraft to determine its position (latitude, longitude, and altitude).[31] However, owing to the state-based models, only state-related property verification, such as deadlock checks, reachability, or safety property verification, is supported by these approaches; i.e., we cannot verify generic reliability relationships for the given system using the approaches presented by Robidoux et al.[21] or Norman and Parker.[30]

A number of HOL formalizations of probability theory are available.[23,32,33] Hurd's formalization of probability theory[32] has been utilized to verify sampling algorithms of a number of commonly used discrete[32] and continuous[34] random variables based on their probabilistic and statistical properties. Moreover, this formalization has been used to conduct the reliability analysis of a number of applications, such as memory arrays[34] and electronic components.[35] However, Hurd's formalization of probability theory only supports use of the whole universe as the probability space. This feature limits its scope; thus, this probability theory cannot be used to formalize more than a single continuous random variable, whereas, in the case of reliability analysis

of pipelines, a number of continuous random variables are required. Recent formalizations of probability theory by Mhamdi[23] and Hölzl [33] are based on extended real numbers (including $\pm\infty$) and provide formalization of the Lebesgue integral to reason about advanced statistical properties. These theories also allow the use of any arbitrary probability space, a subset of the universe, and thus are more flexible than Hurd's formalization.

Leveraging on the high expressiveness of HOL and the inherent soundness of theorem proving, Mhamdi's formalized probability theory[23] has recently been used for the formalization of RBDs, including series,[8] parallel,[24] parallel–series,[24] and series–parallel.[25] These formalizations have been used for the reliability analysis of many applications including simple oil and gas pipelines with serial components,[8] wireless sensor network protocols,[24] and logistic supply chains.[24] Moreover, the probability theory in the HOL theorem prover[23] has also been used to conduct fault-tree-based formal failure analysis of a satellite's solar array[36] and availability analysis.[37] However, to the best of our knowledge, no HOL formalization of the *k*-out-of-*n* RBD configuration, which is frequently used to capture the behavior of complex oil and gas pipelines, has been reported in the literature so far. In this paper, we overcome this limitation by presenting a HOL formalization of the *k*-out-of-*n* RBD configuration, which, in turn, is used, along with the series–parallel RBD configuration, to formally verify generic reliability expressions for an oil and gas pipeline network used in the oil terminal of Dębogórze.[9]

## Proposed methodology

The proposed methodology for the formal reliability analysis of oil and gas pipeline systems, depicted in Figure 2, allows us to formally verify the reliability expressions corresponding to the given pipeline system description and thus formally check that the given pipeline system satisfies its reliability requirements. The core component of this methodology is the HOL formalization of the notions of probability, reliability, and RBDs.

The given oil and gas pipeline system is first partitioned into segments and the corresponding RBD model is constructed. This model can then be formalized in HOL using the aforementioned core formalizations, particularly the formalization of commonly used RBD configurations. The next step is to assign failure distributions, e.g. exponential and Weibull, to individual components of the given pipeline system. These distributions are also formalized by building on formalized probability theory and are used, along with the formal RBD model, to formalize the given reliability requirements as a proof goal in HOL. The user must reason about the correctness of this proof goal using a theorem prover by building on the core formalizations of probability and reliability theories. If all the subgoals are discharged, we obtain formally verified reliability expressions, which correspond to the reliability requirements of the given pipeline system. Otherwise, we can use the failing subgoals to debug our formalizations of the model (formal RBD model) and requirements (proof goal) or the originally specified model and requirements, as depicted by the red line in Figure 2.

## Preliminaries

In this section, we give a brief introduction to theorem proving and the HOL theorem prover in particular. This will be followed by an overview of the formalization of probability theory[23] and the notion of reliability in HOL that we build on to formalize RBD

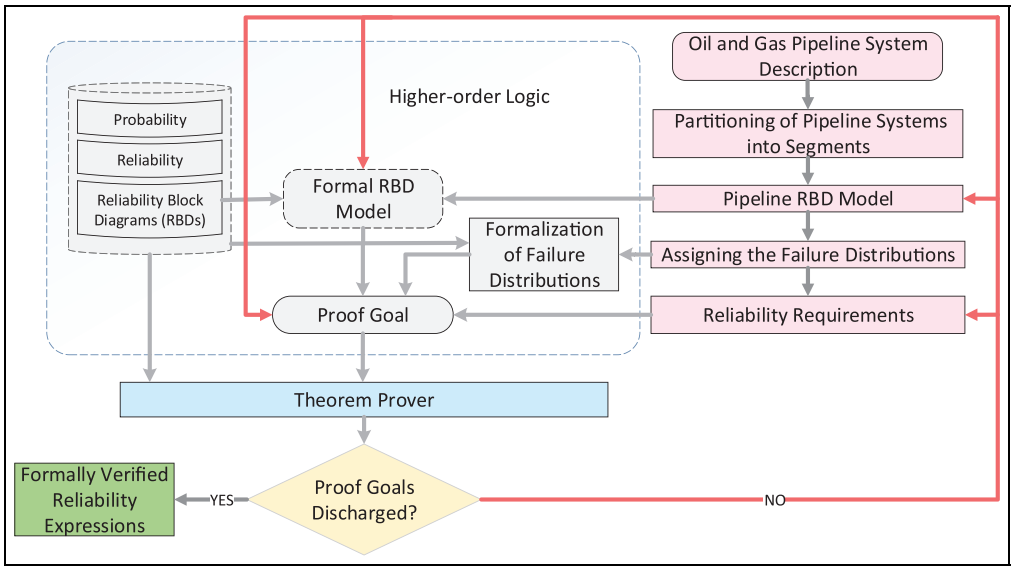


**Figure 2.** Methodology for formal pipeline reliability analysis.

configurations. The intent is to introduce the main ideas behind these foundations to facilitate understanding of the paper.

## Theorem proving

Theorem proving[22] is a widely used formal verification technique. The system that needs to be analyzed is mathematically modeled in an appropriate logic and the properties of interest are verified using computer-based formal tools. The use of formal logic as a modeling medium makes theorem proving a very flexible verification technique as it is possible to formally verify any system that can be described mathematically. The core of theorem provers usually consists of some well-known axioms and primitive inference rules. Soundness is assured as every new theorem must be created from these basic or already proved axioms and primitive inference rules.

The verification effort of a theorem in a theorem prover varies from trivial to complex, depending on the underlying logic. For instance, first-order logic[38] utilizes propositional calculus and terms (constants, function names, and free variables) and is semi-decidable. A number of sound and complete first-order logic automated reasoners are available that enable completely automated proofs. More expressive logics, such as HOL,[39] can be used to model a wider range of problems than first-order logic, but theorem proving for these logics cannot be fully automated and thus involves user interaction to guide the proof tools. For reliability analysis of pipelines, we need to formalize (mathematically model) random variables as functions; their distribution properties are verified by quantifying over random variable functions. Henceforth, first-order logic does not support such formalization and we need to use HOL to formalize the foundations of reliability analysis of pipelines. Consequently, the proofs of the properties of these definitions require human guidance, which can be quite time consuming and requires a deep understanding of the mathematical reasoning behind the proof.

## Higher-order-logic theorem prover

The HOL theorem prover is an interactive theorem prover developed at the University of Cambridge, UK, for conducting proofs in HOL. It utilizes the simple type theory of Church[40] along with Hindley–Milner polymorphism[41] to implement HOL, which has been successfully used as a verification framework for both software and hardware, as well as a platform for the formalization of pure mathematics.

The HOL core consists of only five basic axioms and eight primitive inference rules, which are implemented as ML functions. Soundness is assured, as every new theorem must be verified by applying these basic axioms and primitive inference rules or any other previously verified theorems or inference rules.

We utilized the HOL theories of Booleans, lists, sets, positive integers, real numbers, measure, and probability in our work. In fact, a primary motivation for selecting the HOL theorem prover for our work was to benefit from these built-in mathematical theories. Table 1 provides the mathematical interpretations of some frequently used HOL symbols and functions, which are inherited from existing HOL theories, in this paper.

## Formalization of probability and reliability in higher-order logic

Mathematically, a measure space is defined as a triple $(\Omega, \Sigma, \mu)$, where $\Omega$ is a set, called the sample space, $\Sigma$ represents a $\sigma$-algebra of subsets of $\Omega$, where the subsets are usually referred to as measurable sets, and $\mu$ is a measure with domain $\Sigma$. A probability space is a measure space $(\Omega, \Sigma, Pr)$, such that the measure, referred to as the probability and denoted $Pr$, of the sample space is 1. In the HOL formalization of probability theory,[23] given a probability space $p$, the functions `space`, `subsets`, and `prob` return the corresponding $\Omega$, $\Sigma$, and $Pr$, respectively. This formalization also includes the formal verification of some of the most widely used

**Table 1.** HOL symbols and functions.

| HOL symbol | Standard symbol | Meaning |
|---|---|---|
| $\wedge$ | and | Logical and |
| $\vee$ | or | Logical or |
| $\sim$ | not | Logical negation |
| :: | cons | Adds a new element to a list |
| + + | append | Joins two lists together |
| HD L | head | Head element of list *L* |
| TL L | tail | Tail of list *L* |
| EL n L | element | *n*th element of list L |
| MEM a L | member | True if *a* is a member of list *L* |
| LENGTH L | length | Length of list *L* |
| $\lambda\ x.t$ | $\lambda x.t$ | Lambda abstraction function that maps $x$ to $t(x)$ |
| SUC n | $n + 1$ | Successor of *n* |
| lim(.λn.f(n)) | $\lim_{n\to\infty} f(n)$ | Limit of a real sequence *f* |
| $\{x\|P(x)\}$ | $\{\lambda x.P(x)\}$ | Set of all $x$ that satisfy the condition $P$ |

probability axioms, which play a pivotal role in formal reasoning about reliability properties.

A random variable is a measurable function between a probability space and a measurable space. The measurable functions belong to a special class of functions, which preserves the property that the inverse image of each measurable set is also measurable. A measurable space refers to a pair $(S, \mathcal{A})$, where $S$ denotes a set and $\mathcal{A}$ represents a nonempty collection of subsets of $S$. Now, if $S$ is a set with finite elements, then the corresponding random variable is termed a discrete random variable; otherwise it is called a continuous one.

The probability that a random variable $X$ is less than or equal to some value $t$, $Pr(X \leqslant t)$, is called the cumulative distribution function. It characterizes the distribution of both discrete and continuous random variables. The cumulative distribution function has been formalized in HOL as[8]

```
⊢ ∀ p X t. CDF p X t =
      distribution p X {y|y ⩽ Normal t}
```

where the variables $p$, $X$, and $t$ represent a probability space, a random variable, and a *real* number, respectively. The function `Normal` takes a *real* number as its inputs and converts it to its corresponding value in the *extended real* data-type, i.e, it is the *real* data-type with the inclusion of positive and negative infinity. The function `distribution` takes three parameters: a probability space $p$, a random variable $X$, and a set of extended real numbers, and outputs the probability of a random variable $X$ that acquires all the values of the given set in probability space $p$.

Now, reliability $R(t)$ is stated as the probability of a system or component performing its desired task over a certain interval of time $t$

$$R(t) = Pr(X > t) = 1 - Pr(X \leqslant t) = 1 - F_X(t) \qquad (1)$$

where $F_X(t)$ is the cumulative distribution function. The random variable $X$, in this definition, models the time to failure of the system and is usually modeled by the exponential random variable with parameter $\lambda$, which corresponds to the failure rate of the system. Based on the HOL formalization of probability theory,[23] equation (1) has been formalized as[8]

```
⊢ ∀ p X t. Reliability p X t = 1 - CDF p X t
```

The series RBD, presented by Ahmed et al.,[8] is based on the notion of mutual independence of random variables, which is one of the most essential prerequisites for reasoning about the mathematical expressions for all RBDs. If $N$ reliability events are mutually independent then

$$Pr\left(\bigcap_{i=1}^{N} A_i\right) = \prod_{i=1}^{N} \Pr(A_i) \qquad (2)$$

This concept has been formalized as[8]

```
⊢ ∀ p L. mutual_indep p L =
∀ L1 n. PERM L L1 ∧
1 ⩽ n ∧ n ⩽ LENGTH L ⇒
  prob p (inter_list p (TAKE n L1)) =
  list_prod (list_prob p (TAKE n L1))
```

The function `mutual_indep` accepts a list of events $L$ and probability space $p$ and returns *True* if the events in the given list are mutually independent in the probability space $p$. The predicate `PERM` ensures that its two lists as its arguments form a permutation of one another. The function `LENGTH` returns the length of the given list. The function `TAKE` returns the first $n$ elements of its argument list as a list. The function `inter_list` performs the intersection of all the sets in its argument list of sets and returns the probability space if the given list of sets is empty. The function `list_prob` takes a list of events and returns a list of probabilities associated with the events in the given list of events in the given probability space. Finally, the function `list_prod` recursively multiplies all the elements in the given list of real numbers. Using these functions, the function `mutual_indep` models the mutual independence condition such that for any one or more events $n$ taken from any permutation of the given list $L$, the property $\Pr(\cap_{i=1}^{N} A_i) = \prod_{i=1}^{N} \Pr(A_i)$ holds.

## Formalization of the reliability block diagrams

Reliability block diagrams[42] are graphical structures consisting of blocks and connector lines. The blocks usually represent the system components and the connection of these components is described by the connector lines. The system is functional if at least one path of properly functional components from input to output exists; otherwise it fails.

An RBD configuration can follow any of these three basic patterns of component connections: (i) series; (ii) active redundancy; or (iii) standby redundancy. In the series connection, shown in Figure 3(a), all the components should be functional for the system to remain functional. Whereas, in active redundancy, all the components in at least one of the redundant stages must be functioning in the fully operational mode. The components in active redundancy, in Figure 3, might be connected in a parallel structure (Figure 3(b)) or a combination of series and parallel structures, as shown in Figure 3(c). In standby redundancy, not all components are required to be active, as shown in Figure 3(d). This type of RBD is known as $k$-out-of-$n$ RBD, where at least $k$ components out of $n$ system components must be in an active state. Three types of information are necessary to build the RBD of a given system: (i) functional interaction of the system components, (ii) reliability of each component, and (iii) mission times at which the reliability is desired. This information is then utilized by design engineers to identify the appropriate RBD configuration (series, parallel

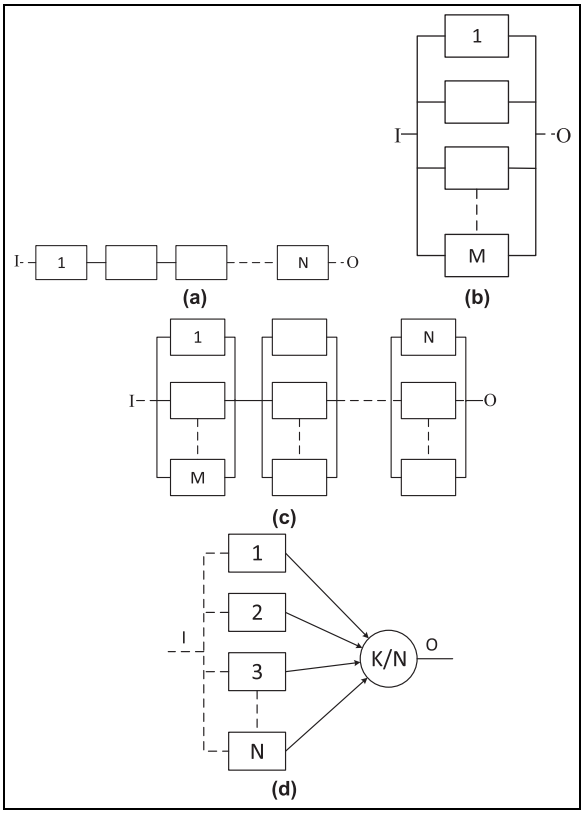


**Figure 3.** Reliability block diagrams: (a) series; (b) parallel; (c) series–parallel; (d) *k*-out-of-*n*.

or series–parallel) to determine the overall reliability of the given system.

The most commonly used RBD configurations used for the reliability analysis of oil and gas pipelines include series, parallel, and a combination of both, and are depicted in Figure 3. In this paper, we present their formalization, which can be used in turn to formally model the structures of oil and gas pipelines in HOL and reason about their reliability, availability, and maintainability characteristics.

## Higher-order-logic formalization of reliability event

In this paper, we have verified the reliability expressions for the commonly used RBD configurations by using reliability event lists, where a single event represents the scenario when the given system or component does not fail before a certain time

**Definition 1.** ⊢ ∀ p X t.
```
rel_event p X t =
PREIMAGE X {y | Normal t < y} ∩ p_space p
```

The function `PREIMAGE` takes two arguments, a function *f* and a set *s*, and returns a set, which is the domain of the function *f* operating on a given range set *s*. The function `rel_event` accepts a probability space *p*, a random variable *X*, representing the failure time of a system or a component, and a real number *t*, which represents the time index at which the reliability is desired. It returns an event representing the reliable functioning of the system or component at time *t*.

Similarly, a list of reliability events is derived by mapping the function `rel_event` on each element of the given random variable list in HOL as

**Definition 2.** ⊢ ∀ p L t.
```
rel_event_list p L t =
MAP (λa. rel_event p a t) L
```

where the HOL function `MAP` takes a function *f* and a list and returns a list by applying the function *f* on each element of the given list.

In the following sections, we present the HOL formalization of RBDs on any reliability event list of arbitrary length.[24,25] The notion of reliability event is then incorporated in the formalization while carrying out the reliability analysis of oil and gas pipelines, as described later in this paper.

## Formalization of series reliability block diagram

The reliability of a system with components connected in series is considered to be reliable at time *t* only if all of its components are functioning reliably at time *t*, as depicted in Figure 3(a). If $A_i(t)$ is a mutually independent event that represents the reliable functioning of the *i*th component of a serially connected system with *N* components at time *t*, the overall reliability of the complete system can be expressed as[5]

$$R_{\text{series}}(t) = Pr\left(\bigcap_{i=1}^{N} A_i(t)\right) = \prod_{i=1}^{N} R_i(t) \tag{3}$$

We formalized the serial RBD configuration as[8]

**Definition 3.** ⊢ (∀ p.
```
series_struct p [] = p_space p) ∧
(∀ p h t. series_struct p (h::t) =
        h ∩ series_struct p t)
```

This function takes a list of events *L* corresponding to the failure of individual components of the given system and the probability space *p* and returns the intersection of all of the elements in a given list *L* and the whole probability space, if the given list is empty. Based on this function definition, the result of equation (3) is formally verified as

**Theorem 1.** ⊢ ∀ p L. `prob_space p ∧ ¬ NULL L ∧`
```
mutual_indep p L ∧ in_events p L ⇒
 (prob p (series_struct p L) =
  list_prod (list_prob p L))
```

The first assumption ensures that *p* is a valid probability space based on the probability theory in HOL.[23] The next two assumptions guarantee that the list of

events, representing the reliability of individual components, must have at least one event and that the reliability events are mutually independent. The predicate `in_events` ensures that each member of the given event list $L$ must in be in event space $p$. The conclusion of the theorem represents equation (3). It is important to note that our `series_struct` definition accepts a list of reliability events and is thus different from the corresponding formalization, presented by Ahmed et al.,[8] which accepts a list of random variables and is not general enough to cater for nested RBDs.

### Formalization of parallel reliability block diagram

The reliability of a system with parallel connected submodules, depicted in Figure 3(b), mainly depends on the component with the maximum reliability. In other words, the system will continue functioning as long as at least one of its components remains functional. If the event $A_i(t)$ represents the reliable functioning of the $i$th component of a system with $N$ parallel components at time $t$, the overall reliability of the system can be mathematically expressed as[5]

$$R_{\text{parallel}}(t) = Pr\left(\bigcup_{i=1}^{N} A_i(t)\right) = 1 - \prod_{i=1}^{N}(1 - R_i(t)) \qquad (4)$$

To formally verify equation (4), we first define the parallel RBD configuration in HOL as

**Definition 4.**
```
⊢ (parallel_struct [] = {}) ∧
(∀ h t. parallel_struct (h::t) =
        h ∪ parallel_struct t)
```

The function `parallel_struct` accepts a list of reliability events and returns the parallel structure reliability event by recursively performing the union operation on the given list of reliability events or an empty set if the given list is empty.

Now, using Definition 4, we can formally verify equation (4) as

**Theorem 2.**
```
⊢ ∀ p L. prob_space p ∧ ¬ NULL L ∧
 mutual_indep p L ∧ in_events p L ⇒
  (prob p (parallel_struct L) =
   1 - list_prod
     (one_minus_list (list_prob p L)))
```

This theorem is verified under the same assumptions as Theorem 1. The conclusion of the theorem represents equation (4), where, the function `one_minus_list`, which accepts a list of *real* numbers $[x_1, x_2, x_3, \ldots, x_n]$ and returns the list of *real* numbers, such that each element of this list is 1 minus the corresponding element of the given list, i.e., $[1 - x_1, 1 - x_2, 1 - x_3, \ldots, 1 - x_n]$. The proof of Theorem 2 is primarily based on Theorem 1, along with the fact that, given the list of $n$ mutually independent events, the complements of these $n$ events are also mutually independent.

### Formalization of series–parallel reliability block diagram

If in each serial stage the components are connected in parallel, as shown in Figure 3(c), then the configuration is termed as a series–parallel structure. If $A_{ij}(t)$ is the event corresponding to the proper functioning of the $j$th component connected in an $i$th subsystem at time $t$, then the reliability of the complete system can be expressed mathematically as[5]

$$R_{\text{series}-\text{parallel}}(t) = Pr\left(\bigcap_{i=1}^{N}\bigcap_{j=1}^{M} A_{ij}(t)\right) = \prod_{i=1}^{N}\left(1 - \prod_{j=1}^{M}(1 - R_{ij}(t))\right) \qquad (5)$$

By extending the RBD formalization approach, presented in Theorems 1 and 2, we formally verify the generic reliability expression for the series–parallel RBD configuration, given in equation (5), in HOL as

**Theorem 3.**
```
⊢ ∀ p L. prob_space p ∧
(∀z. MEM z L ⇒ ¬ NULL z) ∧
in_events p (FLAT L) ∧
mutual_indep p (FLAT L) ⇒
 (prob p
   (series_struct p of parallel_struct) L =
 (list_prod of
     (λa. 1 - list_prod (one_minus_list
       (list_prob p a)))) L)
```

The first assumption in Theorem 3 is similar to the one used in Theorem 2. The next three assumptions ensure that the sublists corresponding to the serial substages are not empty and the reliability events corresponding to the subcomponents of the parallel–series configuration are valid events of the given probability space $p$ and are also mutually independent. The HOL function `FLAT` is used to flatten the two-dimensional list, i.e., to transform a list of lists into a single list. The conclusion models the right-hand side of equation (5). The infixr function, `of`, connects series and parallel RBD configurations using the HOL function `MAP` and thus facilitates the natural readability of complex RBD configurations. It is formalized in HOL as

```
⊢ ∀ g f. f of g = (f o (λa. MAP g a))
```

The proof of Theorem 3 uses the results of Theorems 1 and 2 and also requires a lemma that, given the list of mutually independent reliability events, an event corresponding to the series or parallel RBD structure is independent, in probability, with the corresponding event associated with the series–parallel RBD configurations.

### k-out-of-n reliability block diagram

An $n$-component system is said to be in the $k$-out-of-$n$ configuration if we need at least $k$ components out of

the total $n$ components to be functional for the overall functionality of the system.[5] The RBD for a $k$-out-of-$n$ configuration is depicted in Figure 3(d). This behavior can be modeled by utilizing the concept of binomial trials, which are used to find the chances of at least $k$ successes in $n$ trials. Now, if $R$ is the reliability of the $k$ components that are functioning correctly among $n$ components, the reliability of the overall system can be expressed mathematically as[5]

$$\begin{aligned} & R_{k|n}(t) \\ & = \Pr\left(\bigcup_{i=k}^{n}\{\textit{exactly i components functioning properly}\}\right) \\ & = \sum_{\mathrm{i=k}}^{\mathrm{n}}\left(\binom{n}{k}R^{i}(1-R)^{n-1}\right) \end{aligned} \tag{6}$$

The HOL formalization of $k$-out-of-$n$ RBD is

**Definition 5.** ⊢ ∀ p X k n.
k_out_n_struct p X k n =
BIGUNION (IMAGE
(λ x. PREIMAGE X {Normal (&x)} ∩ p_space p)
{x | k ≤ x ∧ x < SUC n})

The function k_out_n_struct accepts a probability space $p$, a binomial random variable $X$, and two variables, $k$ and $n$, which represent the number of successes and total number of trials, respectively. It then returns the union of the corresponding events that are associated with the binomial random variable $X$, which takes values from the set {x | k ⩽ x ∧ x < SUC n}. The function IMAGE takes a function $f$ and an arbitrary domain set and returns a range set by applying the function $f$ to all the elements of the given domain set. The function BIGUNION returns the union of all the element of given set of sets.

To verify equation (6), we first define a function bino_dist_rand in HOL, which ensures that the random variable $X$ is exhibiting the binomial distribution, as

**Definition 6.** ⊢ ∀ p X R n.
bino_dist_rand p X R n =
(∀x. distribution p X Normal (&x) =
(& binomial n x)*(R pow x)*
(1 - R) pow (n-x))

Similarly, we define a function in_events_k|n to ensure that all the corresponding events that are associated with the binomial random variable $X$ are drawn from the events space $p$

**Definition 7.** ⊢ ∀ p X R n.
in_events_k|n p X n=
(λx. PREIMAGE X Normal(&x) ∩ p_space p) ∈
((count (SUC n)) → events p)

Now, we verify equation (6) in HOL as

**Theorem 4.** ⊢ ∀ p n k X R. prob_space p ∧
k ≤ n ∧ in_events_k|n p X n ∧
bino_dist_rand p X R n ⇒
(prob p (k_out_n_struct p X k n) =
sum (k, SUC n - k)
(λx. (& binomial n x)* (R pow x)*
(1- R) pow (n-x)))

The first and second assumptions ensure that $p$ is a valid probability space and the number of successes of trials $k$ must be less than or equal to the total number of trials $n$. In the third assumption, the function in_events_k|n takes a probability space $p$, a time-to-failure random variable $X$, and a natural number $n$ and ensures that all the $n$ corresponding events that are associated with the random variable $X$ are drawn from the events space $p$. The function bino_dist_rand, in the last assumption, takes the probability space $p$, the time-to-failure random variable $X$, the success probability $R$, and the natural number $n$ and ensures that the random variable $X$ is exhibiting a binomial distribution with success probability $R$, which, in our case, is the reliability of each of the $n$-identical components connected in a $k$-out-of-$n$ structure. The conclusion of the theorem represents equation (6).

An interesting property of equation (6) is that if we put $k = 1$, the structure reduces to a simple parallel structure with components having identical reliabilities. This can be expressed mathematically as

$$R_{1|n}(t) = 1 - (1-R)^{n} \tag{7}$$

This property can be formally verified in HOL as

**Theorem 5.** ⊢ ∀ p n X R. prob_space p ∧
(1 ≤ n) ∧ in_events_k|n p X n ∧
bino_dist_rand p X R n ⇒
(prob p (k_out_n_struct p X 1 n) =
1 - (1 - R) pow n)

Similarly, if the number of successfully functioning components is equal to the total number of components in the $k$-out-of-$n$ configuration, i.e., $k = n$, then the structure reduces to the series configuration of the system with components having identical reliabilities $R$. The HOL formalization of this property is

**Theorem 6.** ⊢ ∀ p n X R. prob_space p ∧
(1⩽n) ∧ in_events_k|n p X n ∧
bino_dist_rand p X R n ⇒
(prob p (k_out_n_struct p X n n) =R pow n)

This formalization of the RBD configurations provides the basis for conducting the RBD-based formal reliability analysis of real-world oil and gas pipeline networks. The distinguishing feature of this formalization is that the variables are quantified for all values; moreover, the theorems are verified for $n$-component RBD configurations. This feature enables us to provide the reliability analysis for large oil and gas pipelines by catering any arbitrary number of pipeline segments,

which is a feature that cannot be provided by model checking and simulation tools.

## Formal reliability analysis of an oil pipeline network

In this section, we illustrate the practical effectiveness of the formalization, presented in the previous section, in analyzing real-world oil and gas pipeline systems. For this purpose, consider the pipeline system depicted in Figure 1(a). It has three pipeline subsystems S1, S2, and S3, which connect the oil terminals A, B, and C in serial order, starting from the oil terminal at the Port of Gdynia.[9] The oil trucks are unloaded at the Port of Gdynia, which is connected by a pipeline subsystem S1 to oil terminal A. The pipeline subsystem S2 provides a path of oil transportation between oil terminals A and B. Similarly, the pipeline subsystem S3 connects oil terminals B and C. At oil terminal C, the wagons transport the oil to the Port of Gdynia railway station and then to the interior regions of the country. There are two identical pipelines in subsystem S1; both of these are partitioned into 178 pipe segments of length 12 m. The identical pipelines in subsystem S2 are partitioned into 717 pipe segments of length 12 m. Similarly, the subsystem S3 is composed of three identical pipelines, which are partitioned into 360 pipe segments of either 10 m or 7.5 m in length.[9]

To conduct the reliability analysis of these oil pipeline subsystems, an effective method is to consider the operational state of these pipeline subsystems while transporting the oil from one oil terminal to the other. This method enables us to select an appropriate RBD structure and thus leads to trustworthy reliability analysis results. There are four main operational states of these pipeline subsystems. [9]

- The operation state z1 is used to transport oil from oil terminal B to oil terminal C using 2-out-of-3 pipelines in subsystem S3.
- The operation state z2 is used to transport oil from the terminal part C to part B using 1-out-of-3 pipelines in subsystem S3.
- The operation state z3 is used to transport oil from the terminal part B through part A to part at the Port of Gdynia using 1-out-of-2 pipelines in subsystem S2 and 1-out-of-2 pipelines in subsystem S1.
- The operation state z4 represents the state when the system is idle, i.e., no oil is transported. At this state, the system can be modeled as three series–parallel RBD structures.

### Formalization of exponential failure distribution

We consider that each pipeline segment is exhibiting the exponential failure distribution, which can be formalized in HOL as

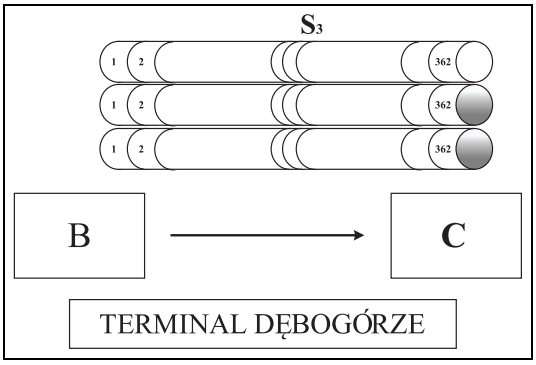


**Figure 4.** Port oil transportation system at operation state z1.

**Definition 8.** ⊢ ∀ p X l. exp_dist p X l = ∀ t. (CDF p X t = if 0 ⩽ t then 1 − exp (− l * t) else 0)

The function exp_dist guarantees that the cumulative distribution function of the random variable $X$ is that of an exponential random variable with a failure rate $l$ in a probability space $p$. We classify a list of exponentially distributed random variables based on this definition as

**Definition 9.** ⊢ ∀ p L. list_exp p [] L = T ∧ ∀ p h t L. list_exp p (h::t) L = exp_dist p (HD L) h ∧ list_exp p t (TL L)

The function list_exp accepts a list of failure rates, a list of random variables $L$, and a probability space $p$. It guarantees that all elements of the list $L$ are exponentially distributed with the corresponding failure rates, given in the other list, within the probability space $p$. For this purpose, it utilizes the list functions HD and TL, which return the *head* and *tail* of a list, respectively. Next we model a two-dimensional list of exponential distribution functions to model failure characteristic of pipeline segments in the operational states z2, z3, and z3 in HOL as

**Definition 10.** ⊢ (∀ p L. list_list_exp p [] L = T) ∧ ∀ h t p L. list_list_exp p (h::t) L = list_exp p h (HD L) ∧ list_list_exp p t (TL L)

The list_list_exp function accepts two lists, i.e., a two-dimensional list of failure rates and random variables $L$. It calls the function list_exp recursively to ensure that all elements of the list $L$ are exponentially distributed with the corresponding failure rates, given in the other list, within the probability space $p$.

### Formal reliability assessment of pipeline subsystems at various operational states

At the system operational state z1, the system is composed of the subsystem S3, which is a series 2-out-of system containing three series partitioned pipelines, as

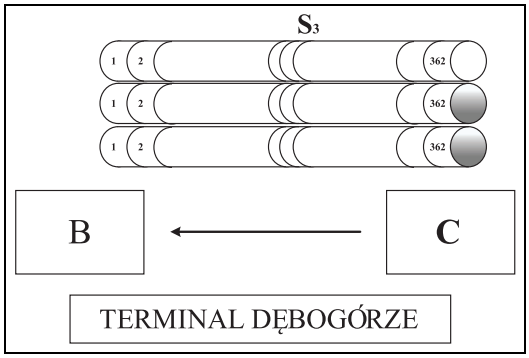


**Figure 5.** Port oil transportation system at operation state z2.

shown in Figure 4. The reliability of the pipeline system operating in z1 can be expressed mathematically as

$$R_{\text{pipeline\_z1}} = 3\exp^{-2\Sigma_{i=1}^{N}\lambda_i t} * (1 - \exp^{-\Sigma_{i=1}^{N}\lambda_i t}) + 3\exp^{-3(\Sigma_{i=1}^{N}\lambda_i)t} \quad (8)$$

We model the RBD configuration, as shown in Figure 4, in HOL as

**Definition 11.** ⊢ ∀ p X R n.
```
rel_pipeline_z1 p X 2 3 =
prob p (k_out_n_struct p X 2 3)
```

Based on this definition, we have formally verified equation (8) in HOL as

**Theorem 7.** ⊢ ∀ p p′ X C L t.
```
(A1): prob_space p ∧ prob_space p′
(A2): in_events_k|n p X 3 ∧
(A3): bino_dist_rand p X
    (pipeline p′(rel_event_list p′ Lt)) 3 ∧
(A4): 0 ⩽ t ∧
(A5): ¬ NULL (rel_event_list p′ L t) ∧
(A6): mutual_indep p′
        (rel_event_list p′ L t) ∧
(A7): list_exp p′ C L ∧
(A8): (LENGTH C = LENGTH L) ⇒
  (rel_pipeline_z1 p X 2 3 =
  3*exp (2 * -list_sum C * t) *
    (1 - exp (-list_sum C * t)) +
      3*exp (-3*list_sum C * t))
```

The assumptions A1, A2, and A3 are similar to those used in Theorem 4, except that the variable $n$ is specified with the natural number 3 and the reliability $R$, in the assumption A3, is replaced by the reliability of the series partitioned identical pipelines, which was described by Ahmed et al.[8] The function `rel_event_list` accepts a probability space $p'$, a list of random variables $L$, representing the failure time of individual components, and a real number $t$, which represents the time index at which the reliability is desired. It returns a list of events, representing the proper functioning of all the individual components at time $t$. It is to be noted that the probability space $p$ for binomial random variable $X$ is different than the probability space $p'$ for time-to-failure random variables, which are assigned to each pipeline segment. The next two assumptions (A4 and A5) ensure that the time index must be positive and that the length of the corresponding events constituted by the random variables in the list $L$ should not be empty, respectively. This is followed by the assumption (A6) that all events are mutually independent; the last two assumptions (A7 and A8) assign failure rates to the exponentially distributed random variables, which are associated with the pipeline segments, and also ensure that the lists of failure rates and random variables have the same length. The `list_exp` function accepts a list of failure rates $C$, a list of random variables $L$, and a probability space $p$. It guarantees that all elements of the list $L$ are exponentially distributed with corresponding failure rates given in the list $C$ within the probability space $p$. The conclusion of the theorem evaluates the reliability of this configuration by utilizing Theorem 4.

At the operational state z2, the system is composed of a series–parallel subsystem S3, which contains three pipelines with the structure, as shown in Figure 5. The reliability of the pipeline system operating in state z2 can be expressed mathematically as

$$R_{\text{pipeline\_z2}} = \prod_{i=1}^{N}\left(1 - \prod_{j=1}^{3}\left(1 - \exp^{\lambda_{ij}t'}\right)\right) \quad (9)$$

We model the reliability of the RBD configuration representing the pipeline system operating at state z2, as shown in Figure 5, in HOL as

**Definition 12.** ⊢ ∀ p L t.
```
rel_pipeline_z2 p L t =
prob p((series_struct p of parallel_struct)
    (List_rel_event_list p L t))
```

where $L$ is a two-dimensional list, which contains the list of random variables associated with the three pipelines. The function `List_rel_event_list` accepts a probability space $p$, a list of random variables, representing the failure time of individual components, and a real number $t$, which represents the time index at which the reliability is desired. It returns a two-dimensional list of events by mapping the function `rel_event_list` on every element of the given two-dimensional list of random variables, which, in turn, models the proper functioning of all individual components at time $t$. To model the three pipeline system operating in state z2 exactly using the series–parallel RBD configuration, it is necessary that each member list of this two-dimensional list $L$ must have length $\leqslant 3$. For this purpose, we have formally defined a function `len_mem_list_le`, which takes a natural number $n$ and a two-dimensional list $L$

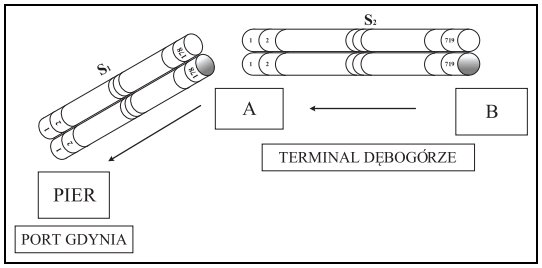


**Figure 6.** Port oil transportation system at operation state z3.

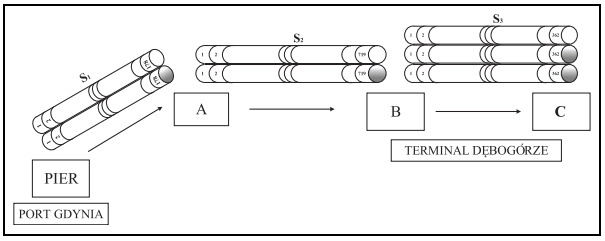


**Figure 7.** Port oil transportation system at operation state z4.

and ensures that the length of each member of given list $L$ must not be greater than $n$, in HOL as

**Definition 13.** ⊢ ∀ n L.
```
len_mem_list_le n L =
(∀x. MEM x L ⇒ (LENGTH x ⩽ n)
```

**Theorem 8.** ⊢ ∀ L C p t.
```
(A1): (0 ⩽ t) ∧(A2): (prob_space p) ∧
(A3): in_events p
        (FLAT (List_rel_event_list p L t)) ∧
(A4): (mutual_indep p
        (FLAT(List_rel_event_list p L t)) ∧
(A5): (∀z. MEM z
    (List_rel_event_list p L t)⇒ ¬NULL z) ∧
(A6): (∀n. n < LENGTH L ⇒
      (LENGTH (EL n L) = LENGTH (EL n C)) ∧
(A7): list_list_exp p C L ∧
(A8): len_mem_list_le 3 L ⇒
  (rel_pipeline_z2 p L t =
   list_prod (one_minus_list
      (list_exp_func_list C t)))
```

where the two-dimensional list $C$ represents the corresponding failure rates of exponentially distributed random variables in the list $L$. The first assumption (A1) of this theorem ensures that the time index is always positive. The next three assumptions (A2 to A4) are similar to those used in Theorem 3. Assumption A5 ensures that the list of random variables associated with the reliabilities of pipeline segments is not empty. Assumptions A6 and A7 guarantee that the length of the list of random variables and the corresponding list of failure rates for pipeline segments is the same and the exponential distributions of the pipeline segments, connected in the series–parallel structure, are associated with their corresponding failure rates, respectively. The last assumption (A8) ensures that the length of the member list of two-dimensional exponentially distributed random variables list $L$ must not be greater than three, thus allowing us to model the behavior, which is discussed in the description of Definition 13. The conclusion of Theorem 8 models the reliability of the series–parallel pipeline system in the operational state z2. The function `list_exp_func_list` accepts a two-dimensional list of failure rates and returns a list with products of one minus the exponentials of every sublist. For example `list_exp_func_list`

$$[[c1; c2; c3]; [c4; c5]; [c6; c7; c8]]x =$$
$$[(1 - e^{-(c1)x}) * (1 - e^{-(c2)x}) *$$
$$(1 - e^{-(c3)x}); (1 - e^{-(c4)x}) * (1 - e^{-(c5)x}); (1 - e^{-(c6)x}) *$$
$$(1 - e^{-(c7)x}) * (1 - e^{-(c8)x})]$$

At the system operational state z3, the series configuration is composed of two series–parallel subsystems S1 and S2, each containing two pipelines with the structure shown in Figure 6. The reliability of the pipeline system at operating state z3 can be expressed mathematically as

$$R_{\text{pipeline\_z3}} = \left(\prod_{i=1}^{N} 1 - \prod_{j=1}^{2} \left(1 - \exp^{\lambda_{ij}t}\right)\right) * \left(\prod_{k=1}^{M} 1 - \prod_{l=1}^{2} \left(1 - \exp^{\lambda_{kl}t}\right)\right) \tag{10}$$

where the arbitrary variables $N$ and $M$ represent the number of segments in the pipelines S1 and S2, respectively. The first part in the right-hand side of this equation corresponds to the reliability of the pipeline system S1 and the second part corresponds to the pipeline system S2, shown in Figure 6. The HOL formalization of reliability of pipeline system at operation state z3 is as

**Definition 14.** ⊢ ∀ p L1 L2 t.
```
rel_pipeline_z3 p L1 L2 t =
prob p ((series_struct p of parallel_struct)
     (List_rel_event_list p L1 t) ∩
 (series_struct p of parallel_struct)
     (List_rel_event_list p L2 t))
```

where the two-dimensional lists $L1$ and $L2$ contain the time-to-failure random variables associated with the pipelines S1 and S2, modeled by series–parallel RBD configuration at operation state z3, respectively. Now, based on Definition 14, we formally verified the reliability expression, given in equation (10), in HOL as

**Theorem 9.** ⊢ ∀ L1 L2 C1 C2 p t.
(A1): 0 ⩽ t ∧ (A2): prob_space p ∧
(A3): in_events p
(FLAT(List_rel_event_listp(L1 ++ L2) t)) ∧
(A4): (mutual_indep p
(FLAT(List_rel_event_list p (L1 ++ L2)t)) ∧
(A5): (∀z. MEM z
(List_rel_event_list p (L1 ++ L2) t) ⇒
¬NULL z) ∧
(A6): (∀n. n < LENGTH (L1 ++ L2) ⇒
(LENGTH (EL n (L1 ++ L2) =
LENGTH (EL n (C1 ++ C2))) ∧
(A7): list_list_exp p (C1 ++ C2) (L1 ++ L2) ∧
(A8): len_mem_list_le 2 L1 ∧
len_mem_list_le 2 L2 ⇒
(rel_pipeline_z3 p L1 L2 t =
list_prod (one_minus_list
(list_exp_func_list C1 t)) *
list_prod (one_minus_list
(list_exp_func_list C2 t)))

The assumptions are similar to those used in Theorem 8 and the conclusion models the reliability of the system, as given in equation (10).

At the system operational state z4, the system is formed by a series RBD and composed of three pipeline subsystems S1, S2, S3, and thus covers the complete pipeline system as shown in Figure 7. The reliability of the pipeline system at operation state z4, as shown in Figure 7, can be expressed mathematically as

$$R_{\text{pipeline\_z4}} = \prod_{i=1}^{N}\left(1-\prod_{j=1}^{2}\left(1-\exp^{\lambda_{ij}t}\right)\right)*\prod_{k=1}^{M}\left(1-\prod_{l=1}^{2}\left(1-\exp^{\lambda_{kl}t}\right)\right)*\prod_{p=1}^{R}\left(1-\prod_{q=1}^{3}\left(1-\exp^{\lambda_{pq}t}\right)\right) \quad (11)$$

where the first part of the right-hand side of this equation corresponds to the reliability of the pipeline system S1 and the second part to the pipeline system S2, shown in Figure 7, respectively.

The HOL formalization of the reliability of pipeline system at operation state z3 is as

**Definition 15.** ⊢ ∀ p L1 L2 L3 t.
rel_pipeline_z4 p L1 L2 L3 t =
prob p ((series_struct p of parallel_struct)
(List_rel_event_list p L1 t) ∩
(series_struct p of parallel_struct)
(List_rel_event_list p L2 t) ∩
(series_struct p of parallel_struct)
(List_rel_event_list p L3 t))

The reliability expression, given in equation (11), can be formally verified in HOL as

**Theorem 10.** ⊢ ∀ L1 L2 L3 C1 C2 C3 p t.
(A1): 0 ⩽ t ∧ (A2): prob_space p ∧
(A3): in_events p (FLAT
(List_rel_event_listp(L1 ++ L2 ++ L3)t)) ∧
(A4): (mutual_indep p (FLAT
(List_rel_event_list p(L1 ++ L2 ++ L3) t)) ∧
(A5): (∀z. MEM z
(List_rel_event_list p (L1 ++ L2 ++ L3) t)
⇒¬ NULL z) ∧
(A6): (∀n. n < LENGTH (L1 ++ L2 ++ L3) ⇒
(LENGTH (EL n (L1 ++ L2 ++ L3) =
LENGTH (EL n (C1 ++ C2 ++ C3))) ∧
(A7): list_list_exp p (C1 ++ C2 ++ C3)
(L1 ++ L2 ++ L3) ∧
(A8): len_mem_list_le 2 L1 ∧
len_mem_list_le 2 L2 ∧
len_mem_list_le 3 L3 ⇒
(rel_pipeline_z4 p L1 L2 L3 t =
list_prod (one_minus_list
(list_exp_func_list C1 t)) *
list_prod (one_minus_list
(list_exp_func_list C2 t)) *
list_prod one_minus_list
(list_exp_func_list C3 t)))

The assumptions of this theorem are similar to those used in Theorem 9; the conclusion of the theorem evaluates the reliability of the pipeline system shown in Figure 7. The proofs of Theorems 7 to 10 are primarily based on induction and verified by utilizing RBD configuration theorems, presented earlier in this paper, along with some fundamental axioms of probability theory.

These theorems provide a comprehensive RBD-based formal reliability analysis by considering different operational states of the given pipeline system. The distinguishing features of the formally verified results, presented in this section, include their generic nature, i.e., all the variables are universally quantified and thus can be specialized to obtain the reliability of the given pipeline network for any given parameters. The correctness of the results is guaranteed, owing to the involvement of a sound theorem prover in their verification. This fact ensures that all the required assumptions for the validity of the result accompany the theorems, which is not the case with the corresponding paper-and-pencil based proofs for the same relations for the give pipeline network.[9]

The formally verified reliability expressions, which are presented in Theorems 7 to 10, provide useful insights for system design engineers and can be used to certify reliability results obtained using traditional techniques, such as paper-and-pencil methods or computer simulations. For instance, it is very useful to know the pipeline segment with the least reliability and how it can effect the reliability of overall pipeline system. So, by keeping this in mind, our formalization enables reliability design engineers to analyze the effect of pipeline segments with low reliability on the overall pipeline

system accurately, owing to the involvement of a mechanized reasoning process within the sound core of the HOL theorem prover. Moreover, the individual failure rates of the pipeline segments can be easily provided to these theorems in the form of a list, i.e., $C$. Another novelty worth mentioning is that the function `len_mem_list_le` can be utilized to model any number of parallel pipeline systems, for instance, in pipeline systems S1 and S2, the function takes value 2 to model two parallel pipelines; in pipeline system S3, it takes the natural number 3 to model three pipelines.

These benefits are not shared by any other computer-based reliability analysis approach for oil and gas pipelines; this clearly indicates the usefulness of the proposed approach. These added benefits are attained at the cost of the explicit guidance required to formalize the results, presented in this and the previous section. Our proof script for these formally verified results is composed of more than 7000 lines of code and took about 250 man-hours of effort.[43] Most of the effort was made on the formalization of RBD configurations and the verification of their corresponding generic reliability expressions. This formalization considerably facilitated the formalization of the oil and gas pipeline system, as the analysis only took about 2500 lines of HOL code and very little manual interaction, compared with the theorems, presented in this section.

## Conclusions and future work

Many probabilistic reliability assessment techniques have been developed during the last two decades to assess the reliability of oil and gas pipelines. However, the analysis based on these probability theoretic approaches is carried out using informal system analysis methods, like simulation or paper-and-pencil, and thus does not ensure accurate results. The accuracy of the pipeline reliability assessment results is critical for oil and gas pipelines since even minor flaws in the analysis could trigger the loss of many human lives or cause heavy damages to the environment. To achieve this goal and overcome the inaccuracy limitation of the traditional probabilistic analysis techniques, we propose to build on our proposed formalization of RBDs to formally reason about the reliability of oil and gas pipelines using HOL theorem proving. For illustration purposes, we formally verified the reliability expressions of the oil pipeline system between the oil terminals at the Port of Gdynia and Dębogórze.

To facilitate the utilization of our proposed approach, we plan to build a graphical user interface (GUI) that can be used to capture any RBD model, such as a oil and gas pipeline system RBD, from the user and return the formally verified reliability expression, using an HOL theorem prover that is running seamlessly underlying this GUI, of the given system. This would bring great benefits to non-HOL users, like industrial reliability engineers, in many respects. For instance, it can be used to certify the results estimated by the design engineer and provide an opportunity at the design stage to correct this estimated result, if it is not validated by the HOL theorem prover. We also plan to formalize multistate reliability theory,[44] which is based on a semi-Markov process, and can be used to reason about the impact of change in time on reliability of the system. This formalization would require the formalization of a semi-Markov chain and its associated concepts.

### Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

### Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Qatar National Research Fund (a member of the Qatar Foundation) (NPRP grant number [5 - 813 - 1 134]). The statements made herein are solely the responsibility of the authors.

### References

1. Veresen. Alliance Pipeline, http://www.vereseninc.com/our-business/pipelines/alliance-pipeline/ (accessed 07 February 2017).
2. National Commission on the BP Deepwater Horizon Oil Spill Offshore Drilling. Deep Water: The Gulf oil disaster and the future of offshore drilling. Report to the President, USA, January 2011.
3. Wenman T and Dim JC. Pipeline integrity management. In: *Abu Dhabi international petroleum conference and exhibition*, Abu Dhabi, United Arab Emirates, 11–14 November 2012. Richardson, TX: Society of Petroleum Engineers.
4. Parker CM. Pipeline industry meets grief unimaginable: congress reacts with the Pipeline Safety Improvement Act of 2002. *Nat Resources J* 2004; 44: 243.
5. Narasimhan K. Reliability engineering: theory and practice. *TQM Magazine* 2005; 17(2): 209–210.
6. Kołowrocki K. *Reliability of large systems*. Amsterdam: Elsevier, 2004.
7. Zhang Z and Shao B. Reliability evaluation of different pipe section in different period. In: *IEEE international conference on service operations and logistics, and informatics*, Beijing, China, 12–15 October 2008, pp.1779–1782. Piscataway, NJ: IEEE.
8. Ahmed W, Hasan O, Tahar S, et al. Towards the formal reliability analysis of oil gas pipelines. In: Watt SM, Davenport JH, Sexton AP, et al. (eds) *Intelligent computer mathematics*. Cham: Springer, 2014, vol. 8543, pp.30–44. LNCS.
9. Soszynska J. Reliability and risk evaluation of a port oil pipeline transportation system in variable operation conditions. *Int Press Vessels Pip* 2010; 87(2–3): 81–87.

10. GE Oil and Gas. Pipeline integrity management, http://site.ge-energy.com/prod_serv/serv/pipeline/en/index.htm (accessed 16 February 2017)
11. Creaform. Pipecheck—Pipeline integrity assessment software, http://www.creaform3d.com/en/ndt-solutions/pipecheck-damage-assessment-software (accessed 16 February 2017).
12. DNV-GL. http://www.dnvgl.com/oilgas/ (accessed 16 February 2017).
13. ReliaSoft. http://www.reliasoft.com/ (accessed 16 February 2017).
14. Hasan O and Tahar S. Formal verification methods. In: Khosrow-Pour M (ed.) *Encyclopedia of information science and technology*. Hershey, PA: IGI Global, 2014, pp.7162–7170.
15. Thomas M. The role of formal methods in achieving dependable software. *Reliab Eng Syst Saf* 1994; 43(2): 129–134.
16. Clarke E, Grumberg O and Peled D. *Model checking*. Cambridge, MA: MIT Press, 2000.
17. Harrison J. *Handbook of practical logic and automated reasoning*. Cambridge, UK: Cambridge University Press, 2009.
18. Hasan O and Tahar S. Performance analysis of ARQ protocols using a theorem prover. In: *International symposium on performance analysis of systems and software*, Austin, TX, 20–22 April 2008, pp.85–94. Piscataway, NJ: IEEE.
19. Kwiatkowska M, Norman G and Parker D. Probabilistic model checking for systems biology. In: Iyengar MS (ed.) *Symbolic systems biology*. Burlington, MA: Jones and Bartlett, 2010, pp.31–59.
20. Elleuch M, Hasan O, Tahar S, et al. Formal analysis of a scheduling algorithm for wireless sensor networks. In: *Formal methods and software engineering*. Qin S and Qiu Z (eds). Berlin: Springer, 2011, vol. 6991, pp.388–403. LNCS.
21. Robidoux R, Xu H, Xing L, et al. Automated modeling of dynamic reliability block diagrams using colored Petri nets. *IEEE Trans Syst Man Cybern Part A Syst Humans* 2010; 40(2): 337–351.
22. Gordon M. Mechanizing programming logics in higher-order logic. In: Birtwistle G and Subrahmanyam PA (eds) *Current trends in hardware verification and automated theorem proving*. New York: Springer, 1989, pp.387–439.
23. Mhamdi T, Hasan O and Tahar S. On the formalization of the Lebesgue integration theory in HOL. In: Kaufmann M and Paulson LC (eds) *Interactive theorem proving*. Berlin: Springer, 2011, vol. 6172, pp.387–402. LNCS.
24. Ahmed W, Hasan O and Tahar S. Formal reliability analysis of wireless sensor network data transport protocols using HOL. In: *Wireless and mobile computing, networking and communications*, Abu-Dhabi, United Arab Emirates, 19–21 October 2015, pp.217–224. Piscataway, NJ: IEEE.
25. Ahmad W, Hasan O, Tahar S, et al. Towards formal reliability analysis of logistics service supply chains using theorem proving. In: *International workshop on the implementation of logics, EPiC Series in Computing*, Suva, Fiji, 23 November 2015, volume 40, pp.1–14.
26. ASENT. RBD analysis. https://www.raytheoneagle.com/asent/rbd.htm (accessed 16 February 2017).
27. Signoret JP, Dutuit Y, Cacheux PJ, et al. Make your Petri nets understandable: reliability block diagrams driven Petri nets. *Reliab Eng Syst Saf* 2013; 113: 61–75.
28. PRISM. http://www.prismmodelchecker.org/ (accessed 16 February 2017)
29. Herbert L and Hansen Z. Restructuring of workflows to minimise errors via stochastic model checking: an automated evolutionary approach. *Reliab Eng Syst Saf* 2016; 145: 351–365.
30. Norman G and Parker D. Quantitative verification: formal guarantees for timeliness, reliability and performance. Technical report, The London Mathematical Society and the Smith Institute, 2014, http://www.prismmodelchecker.org/papers/lms-qv.pdf. (accessed 16 February 2017)
31. Lu Y, Peng Z, Miller AA, et al. How reliable is satellite navigation for aviation? Checking availability properties with probabilistic verification. *Reliab Eng Syst Saf* 2015; 144: 95–116.
32. Hurd J. *Formal verification of probabilistic algorithms*. PhD Thesis, University of Cambridge, UK, 2002.
33. Hölzl J and Heller A. Three chapters of measure theory in Isabelle/HOL. In: van Eekelen M, Geuvers H, Schmalz J, et al. (eds) *Interactive theorem proving*. Berlin: Springer, 2011, vol. 6172, pp.135–151. LNCS.
34. Hasan O, Tahar S and Abbasi N. Formal reliability analysis using theorem proving. *IEEE Trans Comput* 2010; 59(5): 579–592.
35. Abbasi N, Hasan O and Tahar S. An approach for lifetime reliability analysis using theorem proving. *J Comput Syst Sci* 2014; 80(2): 323–345.
36. Ahmed W and Hasan O. Towards formal fault tree analysis using theorem proving. In: Kerber M, Carette J, Kaliszyk C, et al. (eds) *Intelligent computer mathematics*. Cham: Springer, 2015, vol. 9150, pp.39–54. LNCS.
37. Ahmed W and Hasan O. Formal availability analysis using theorem proving. In: Ogata K, Lawford M and Liu S (eds) *International conference on formal engineering methods*. Cham: Springer, 2016, vol. 9150, pp.226–242. LNCS.
38. Fitting M. *First-order logic and automated theorem proving*. New York: Springer, 1996.
39. Brown C. *Automated reasoning in higher-order logic*. London: College Publications, 2007.
40. Church A. A formulation of the simple theory of types. *J Symb Logic* 1940; 5: 56–68.
41. Milner R. A theory of type polymorphism in programming. *J Comput Syst Sci* 1977; 17: 348–375.
42. Bilintion R and Allan R. *Reliability evaluation of engineering systems: concepts and techniques*. New York: Springer, 1992.
43. Ahmed W. Formal risk analysis of oil and gas pipelines, http://save.seecs.nust.edu.pk/projects/frogp (accessed 16 February 2017).
44. Natvig B. Multistate reliability theory. In: Ruggeri F, Kenett RS and Faltin FW (eds) *Encyclopedia of statistics in quality and reliability*. Hoboken, NJ: Wiley-Blackwell, 2007.