International Journal of Automation and Computing X(X), X X, X-X
DOI: XXX

Application of a Reliability Model Generator to a Pressure Tank System

Kathryn Stockwell Sarah Dunnett

Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, Leicestershire LE11 3TU, UK

Abstract: A number of mathematical modelling techniques exist which are used to measure the performance of a given system, by assessing each individual component within the system. This can be used to determine the failure frequency or probability of failure of the system. Software is available to undertake the task of analysing these mathematical models after an individual or group of individuals manually create the models. The process of generating these models is time consuming and reduces the impact of the model on the system design. One way to improve this would be to automatically generate the model. In this work the procedure to automatically construct a model, based on Petri nets, for systems undergoing a phased-mission is applied to a pressure tank system, undertaking a four phase mission.

Keywords: Model generation, phased-mission, simulation, decision table, operational mode table.

1 Introduction

The design stage of any new system is a critical time to ensure that the system meets all required standards, particularly those where failures could result in fatalities. By modelling the reliability of the system, alternative design solutions could be investigated and the direction of the design could be influenced in order to meet these regulatory requirements. A number of mathematical modelling techniques exist to determine the reliability of a system, such as fault trees, event trees and Markov analysis. These models cannot usually be created by a member of the design team as they do not have the necessary expertise to carry out the process. Therefore a specialist group or team is often brought in to model the reliability of system designs. The process of creating these models is lengthy and can limit their usefulness, as during this time the design progresses. This causes a lag between the reliability predictions of the design and its development, which reduces the influence of reliability predictions.

The analysis of the models, once constructed, has been the main focus over the years and can now be completed effectively and efficiently. However, the construction of the models still requires significant time and effort, and a particular skill set. One way to improve this is to construct the models automatically. Doing this would enable the design teams to carry out the reliability assessment without acquiring expertise in reliability methods. Automation of the reliability models also reduces the time spent on the model construction and removes human construction errors. The automatic construction of Fault trees has received the most attention, using a variety of methods. The most commonly used approaches to generate Fault Trees include diagraphs[1], decision tables[2], transition tables[3] and mini fault trees[4]. All these approaches have some form of restriction on their application and so no one method can be applied to all systems. Apart from Fault Tree Analysis, automation processes for other modelling techniques have received little attention. The aim of the work presented here is to outline an approach to automate the generation of a reliability model for a system undertaking a phased mission. In order to describe the process an example of a pressure tank system has been described in detail. Phased missions are a collection of consecutive time periods, or phases, where a system must meet different requirements in order to complete the phase successfully. For the mission to be a success each phase within the mission must be successful, therefore any failures within any phase will result in mission failure. Techniques currently used to assess phased-missions include Fault Tree Analysis, Markov Analysis and simulation. Fault Tree Analysis and Markov Analysis both suffer from increasingly large models as the system grows and/or the number of phases within a mission increases. Simulation techniques suit such situations better due to their computational nature allowing for complex systems and scenarios to be considered. A simulation technique which is designed for ease of representation, with significant modelling power is the Petri net (PN). Hence in this paper the automatic generation of a PN model for a system undertaking a phased mission is described. Initially a brief description of PNs is given in section 2, then the general automated process is outlined in section 3. The process applied to an example is described in detail in section 4 and validated for a single mission in section 5. Results for multiple missions for the example are given in section 6 and then some general conclusions are drawn in the final section.

2 Petri nets

A Petri net is a bipartite directed graph with two types of nodes; places and transitions. Places are represented by circles and transitions are represented by either hollow or solid bars. A solid bar is an immediate transition, i.e. the time to transition is zero. A hollow bar represents a time to transition which is greater than zero. Directed arcs create links between the places and the transitions; but places can

Manuscript received date; revised date

only link to transitions and vice versa. Multiple links can occur between places and transitions. This can be shown by a dash on the arc with a number placed next to it to show the multiplicity. If there is no dash then the multiplicity of the arc is one. Another element of a Petri net is tokens, or marks, which reside within the places. These are passed between different places by the switching of transitions. The dynamic behaviour of the Petri net model is represented by this switching of the transitions. At any given time, the distribution of tokens, or net marking, within the Petri net represents a different system state. This is of great interest to the analyst. An example of transition switching is given in Fig. 1.

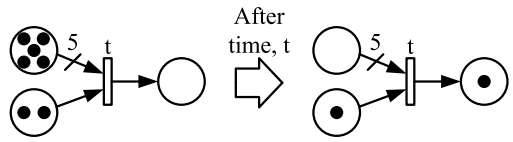

Fig. 1 Petri net showing the positions of the tokens before and after the transition occurs

Fig. 1 shows a transition with two input places; the first linked to the transition with an arc of weight 5 and another linked to the transition with a single arc of weight 1. When all the input places contain at least the weight number of tokens, or marks, the transition becomes enabled. Therefore for the transition to be enabled the first place must contain at least 5 tokens and the second must contain at least 1. This transition has a delay of time, t, associated with it. The switching of the transition cannot occur until this delay of t has lapsed whilst the transition is enabled. If t is zero, then a solid bar would be present, to represent an immediate transition. Once the delay has passed whilst the transition has remained enabled, the switching can occur. The switching process removes 5 tokens from the top place and 1 token from the bottom place, and a single token is placed in the output place connected to the transition.

3 Model Generation

3.1 Description requirements

Before a reliability model can be created, details of the system and the mission are necessary. These details fall into the following categories;

- **Component description** in the form of component models including the component failure modes.
- **System structure** in the form of a system topology diagram.
- **Mission Description** in the form of phase models and, initial and starting conditions.
- **System failure conditions**
- **Failure and repair data**

To model the components two tables are used; decision tables and operational mode tables. Decision tables describe how the component reacts to inputs from other components in the system, depending on the current state of the component. Operational mode tables, similar to state transition tables[5], are only used for components that have more than one mode of operation. These tables describe how the mode of operation can be changed, when a command to the component is introduced. For example, if a switch, which is currently open is commanded by an operator to close, as long as the switch is in a working condition, the switch would change mode from open to closed. The system topology diagram describes how the components are linked together. The phase models describe the different phases the mission can enter with the condition of the system needed to transition from one phase to another. The initial conditions are the conditions the components must satisfy in order for the mission to commence. The system failure conditions are the system failure modes. The failure and repair (if applicable) data is necessary for each component in the system to determine a reliability estimate.

3.2 Model Construction

The model is comprised of four distinctive Petri nets; component nets (CPN), system nets (SPN), phase nets (PPN) and circuit nets (CiPN). Each is necessary to model a different aspect of either the system or the mission. This approach of using distinctive PNs can be seen in the work of [6] and [7]. In this work the CPNs model the components' failure and repair. All circuits identified in the system are represented by the CiPNs. These are used to identify at any given time whether there is current (C) or no current (NC) present in the circuit. Circuits are identified by the software created, by locating all components identified as able to pass current, and which exist within a loop. This process uses the component descriptions in conjunction with the system topology to obtain the list of circuits present. These are only necessary for systems that contain electrical circuits. The SPN is the collection of all CPN instances and the connections, or links, between them in the system. The SPN is generated using the information stored in the topology diagram. The PPN describes the mission the system is undertaking. This is created from the phase transition table; each row of the table represents a transition within the PPN.

3.3 Software Structure

To demonstrate the overall structure of the software Fig. 2 shows how the information described in the previous section about the system, including the components and the mission, is used to create the different Petri net types discussed above.

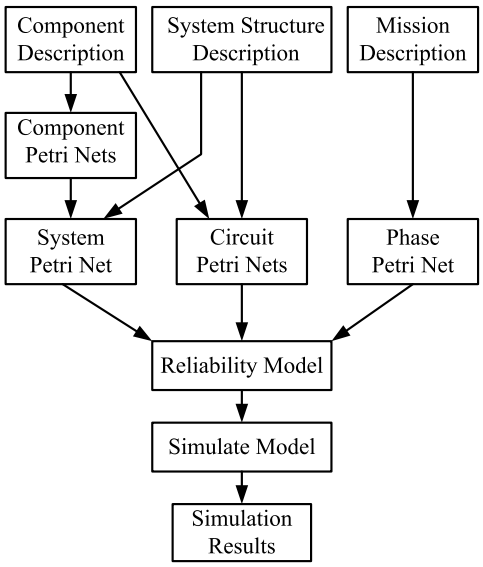

Fig. 2 Software structure diagram

4 Pressure tank system

To demonstrate the construction procedure created, a pressure tank system has been used as a case study. The pressure tank system used here is undertaking a four phase mission.

4.1 System description

The aim of the system shown in Fig. 3 is to control the filling and emptying of the tank (T). The initial state of the system is that it is dormant and therefore de-energised. The push-switch (S1), the timer contact (TC) and the relay contact (RC) are open. The toggle switch (S2) and valve (V) are closed. The tank is empty. All components are working at the start of each mission.

The system is initially started by the operator depressing switch S1, momentarily applying power to the timer relay (TIM), whose contacts close and start the timer. Switch S1's contacts open. Power is applied to relay (R) whose contacts close and start the pump motor. The tank starts to fill. After a time t_1 the timer relay's contacts open, relay R de-energies and its contacts open thus removing power from the pump motor. When T is de-energised the timer clock resets. The operator will notice the tank pressure by the pressure gauge and will open the valve to empty the tank. After a time t_2 the tank will have emptied sufficiently for filling to start again by the operator pressing switch S1 and closing the valve. Switch S2 is a safety mechanism built into the system so that in the event that a failure occurs and the tank overfills the operator, who will be alerted by the pressure gauge, can stop the pump by opening that switch, hence denying power to R.

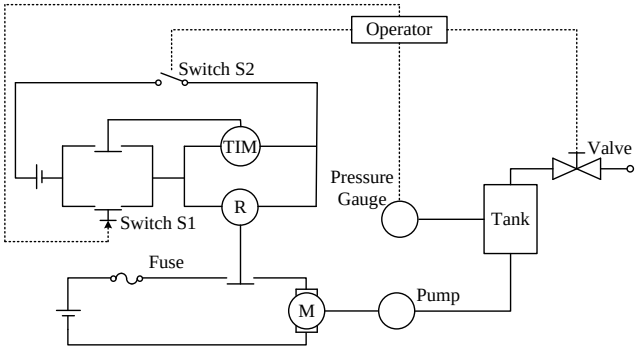

Fig. 3 Schematic of pressure tank system

4.2 Mission Description

The mission is described in the phase transition table seen in Table 1. The four main phases of the mission are detailed below:

- Phase 1: System start-up, discrete phase at $t = 0$.
- Phase 2: Filling of the tank, duration t_1.
- Phase 3: Opening the valve on the tank, discrete phase at $t = t_1$.
- Phase 4: Emptying the tank, duration t_2.

The four main phases represent the system in a fully working condition; therefore other phases are required to signify when the system enters a failure phase, these are listed as follows:

- Phase 5: System failure due to overfill
- Phase 6: System overfill with system shutdown
- Phase 7: System overfill without system shutdown
- Phase 8: System failure not due to overfill
- Phase 9: Mission success

Table 1 Phase transition table for four phase mission

	Time	From Phase	To Phase	Condition
1	0	1	2	TC Mode=Closed
2	δ	1	8	TC Mode=Open
3	t_1	2	3	T Out1=CONST
4	-	2	8	T Out1=CONST
5	-	3	4	V Mode=Open
6	δ	3	5	V Mode=Closed
7	$t_1 + t_2$	4	9	T Out1=DEC
8	-	4	8	T Out1=CONST
9	δ	5	6	RC Mode=Open
10	δ	5	7	RC Mode=Closed

The condition to enter each of these phases is seen under the heading of condition in the phase transition table. The transitions between phases are all based on the condition of a component within the system. For example, the transition between phase 1 and 2 can only occur if the condition

that component TC is in the mode closed at time $t = 0$. When time is designated as δ, this refers to a small amount of time and is used in situations when a single component's output or mode of operation could change the system state. In this example $t_1 = 1$hr and $t_2 = 2$hrs.

4.3 Software inputs

The failure modes of each component type are listed in Table 2. The failure data of each component is required to create the delayed transition within the component PNs. Each component is considered separately and this information is entered by the system designers through a text file that specifies the identifier of the component with either the distribution type, or types, by which the component fails or a definitive value. A definitive value can be used to test how the system would react to a specific component failure at a given time. Using the distribution type and a random number generator a time to failure is generated for each component instance. For this system all components are assumed to fail by the exponential distribution. The failure rate of each component failure mode is listed in Table 2. The system topology diagram of the pressure tank system is shown in Fig. 4. The decision and operational mode tables for the components within the system can be found from Table 3-16. These are generated from component information and the topology diagram. In the tables FL and NFL denote flow and no flow, EN and DE denote energised and de-energised, CL, OP and NA denote close, open and no action and LPR, HPR and VHPR denote low pressure, high pressure and very high pressure, respectively. Decision tables are time dependent when a time, t, column is included, which dictates when that transition can occur.

Table 2 Component failure mode descriptions and failure data

Failure Mode	Failure Rate	Description
S1_FCL	0.1	Switch failed closed
S1_FOP	0.1	Switch failed open
S2_FCL	0.001	Switch failed closed
S2_FOP	0.8698	Switch failed open
PS1_F	0.001	No power
PS2_F	0.001	No power
RC_FCL	0.00023	Contacts failed closed
RC_FOP	0.00023	Contacts failed open
TC_FCL	0.1	Contacts failed closed
TC_FOP	0.1	Contacts failed open
TIM_F	0.001	Relay failed de-energised
R_F	0.1	Relay failed de-energised
FS_F	0.01	Fuse broken
M_F	0.001	Motor broken
P_F	0.1	Pump broken
V_FCL	0.03	Valve failed closed
V_FOP	0.03	Valve failed open
OP_F	0.1	Operator fails to take action
T_F	0.0001	Tank leaks significantly
PG_FLOW	0.01	Gauge stuck at low pressure
PG_FHIGH	0.01	Gauge stuck at high pressure
PG_FVHIGH	0.01	Gauge stuck at very high pressure

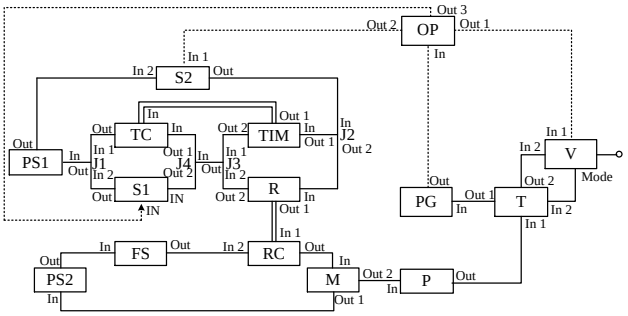

Fig. 4 System topology diagram for the pressure tank system

Table 3 Operational mode table for push switch, S1

	Mode 1	Command (In1)	State	Mode 2
1	Closed	–	FCL	Closed
2	Closed	CL	–	Closed
3	Closed	NA	W	Open
4	Open	–	FOP	Open
5	Open	NA	–	Open
6	Open	CL	W	Closed

Table 4 Operational mode table for toggle switch, S2 and Valve, V

	Mode 1	Command (In1)	State	Mode 2
1	Closed	–	FCL	Closed
2	Closed	CL	–	Closed
3	Closed	OP	W	Open
4	Closed	NA	–	Closed
5	Open	–	FOP	Open
6	Open	OP	–	Open
7	Open	CL	W	Closed
8	Open	NA	–	Open

Table 5 Operational mode table for contacts, RC and TC

	Mode 1	Command (In1)	State	Mode 2
1	Closed	–	FCL	Closed
2	Closed	EN	–	Closed
3	Closed	DE	W	Open
4	Open	–	FOP	Open
5	Open	DE	–	Open
6	Open	EN	W	Closed

Table 6 Decision table for switches, S1 and S2, and contacts, RC and TC

	In 2	Mode	Out
1	–	Open	NC
2	NC	–	NC
3	C	Closed	C

Table 7 Decision table for power supplies, PS1 and PS2, and fuse, FS

	In	State	Out
1	C	W	C
2	–	F	NC
3	NC	–	NC

Table 8 Decision table for timer relay, TIM

	t	In	State	Out 1	Out 2
1	$t < t_1$	C	W	EN	C
2	$t \geq t_1$	C	W	DE	NC
3	–	–	F	DE	NC
4	–	NC	–	DE	NC

Table 9 Decision table for relay, R

	In	State	Out 1	Out 2
1	C	W	EN	C
2	–	F	DE	NC
3	NC	–	DE	NC

Table 10 Decision table for junctions, J1, J2, J3 and J4

	In 1	In 2	Out 1	Out 2
1	C	–	C	
2	–	C	C	
3	NC	NC	NC	
4	C		C	C
5	NC		NC	NC

Table 11 Decision table for motor, M

	In	State	Out 1	Out 2
1	C	W	C	ON
2	–	F	NC	OFF
3	NC	–	NC	OFF

Table 12 Decision table for pump, P

	In	State	Out
1	ON	W	FL
2	–	F	NFL
3	OFF	–	NFL

Table 13 Decision table for pressure gauge, PG

	t	In	State	Out 1
1	$t < t_1$	CONST	W	LPR
2	$t < t_1$	INC	W	LPR
3	t_1	CONST	W	LPR
4	t_1	INC	W	HPR
5	–	DEC	W	LPR
6	$t_1 < t \leq t_1 + t_2$	CONST	W	HPR
7	$t_1 < t \leq t_1 + t_2$	INC	W	VHPR
8	–	–	FLOW	LPR
9	–	–	FHIGH	HPR
10	–	–	FVHIGH	VHPR

Table 14 Decision table for valve, V

	In 2	Mode	Out
1	–	Closed	NFL
2	NFL	–	NFL
3	FL	Open	FL

Table 15 Decision table for tank, T

	t	In 1	In 2	State	Out 1	Out 2
1	–	FL	Open	W	CONST	FL
2	–	FL	Closed	W	INC	NFL
3	–	NFL	Closed	W	CONST	NFL
4	$t \leq t_1$	NFL	Open	W	CONST	NFL
5	$t_1 < t \leq t_1 + t_2$	NFL	Open	W	DEC	FL
6	$t \leq t_1$	–	–	F	CONST	NFL
7	$t_1 < t \leq t_1 + t_2$	–	–	F	DEC	NFL

Table 16 Decision table for operator, OP

	t	In 1	State	Out 1	Out 2	Out 3
1	0	LPR	W	CL	CL	CL
2	$0 < t < t_1 + t_2$	LPR	W	NA	NA	NA
3	–	HPR	W	OP	NA	NA
4	–	VHPR	W	NA	OP	NA
5	–	–	F	NA	NA	NA

4.4 Petri net models

4.4.1 Component and system Petri nets

The CPNs are constructed directly from the decision and operational mode tables, where each row of the tables is treated as a separate transition. For decision tables, the conditions under the headings In, State, and Mode are used as input places to the transitions, and the conditions under the heading Out are treated as the output places. In the case of multiple inputs/outputs,In i and Out i are used, where i is an arbitrary index. Within the operational mode tables the headings Mode 1, In i and State are treated as the input places of the transition and the conditions under the heading Mode 2 are treated as the output place of the transition. An example of the push switch, S1, CPN using both Table 3 and Table 6 can be seen in Fig. 5. The working to failed state PN for each component is created separately from the table using the failure data. The current state of the component feeds into the CPN, as shown for S1 by the dotted arrows in Fig. 5. If the component only has one failure mode then a single transition will exist between the working and failed state places. If however the component has multiple failure modes and the distributions by which they fail are different, then a delayed transition will exist for each failure mode. If all failure modes fail by the same distribution then a single delayed transition is required between the working and a designated failed state. From the failed state, the failure mode can be determined from the current mode of the component. During the construction process of the CPNs, a time to failure for each component is generated using the information given in Table 2. Each of these values follow an exponential distribution shown in (1).

$$Q(t) = 1 - e^{-\lambda t} \tag{1}$$

where $Q(t)$ is the unreliability at time t and λ is the failure rate.

By rearranging (1) for t and substituting $Q(t)$ for a random number between 0 and 1, random times to failure for the components can be generated based on their failure rate. Components with other distributions, namely normal and

weibull distribution are also accepted by the software. The time to failures are generated from these distributions in similar ways. The SPN is generated by merging the output place of one component to the input place of the connected component, until all components are connected together. An example of this connection can be seen in Fig. 6. This is completed using the information in the topology diagram.

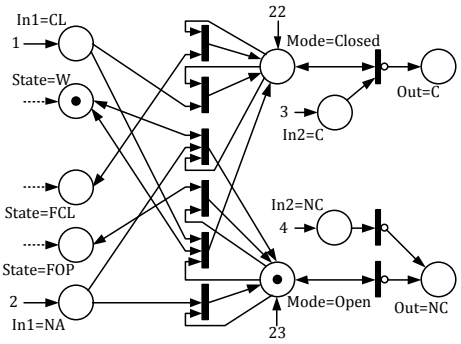

Fig. 5 Component Petri net of S1

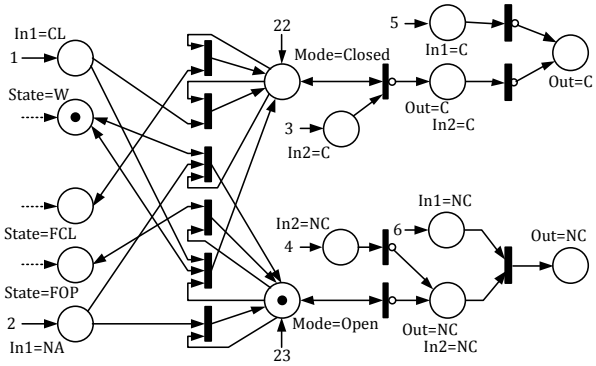

Fig. 6 Example section of the system Petri net of S1 and J1 connected

4.4.2 Circuit Petri nets

The CiPNs are automatically generated using the list of circuits identified. The circuits identified within this system are as follows:

- C1 ={PS1, S2, J2, TIM, J3, J4, TC, J1, PS1}
- C2={ PS1, S2, J2, TIM, J3, J4, S1, J1, PS1}
- C3={PS1, S2, J2, R, J3, J4, TC, J1, PS1}
- C4={ PS1, S2, J2, R, J3, J4, S1, J1, PS1}
- C5={PS2, F, RC, M, PS2}

An example of a constructed CiPN can be seen in Fig. 7 for the circuit list C3. The CiPNs connect to the SPN through a component within the circuit list. The CiPN acts as a way of initiating the flow of tokens within the SPN, showing either the flow or no flow of current in the circuits of the system. One half of the CiPN describes the condition for current within a circuit; all components must either be in a working condition, or completing the circuit through the mode of the component. Using the example of the CiPN in Fig. 7, for current to flow within the circuit, the components PS1 and R must be in a working state, and S2 and TC must be in an operational mode of closed, to create a complete circuit connection. The second half of the CiPN describes the condition(s) required for no current within the circuit. In the example, PS1 or R could be in a failed state, or S2 or TC could be in an operational mode of open. One of these conditions would cause a state of no current flowing within that circuit.

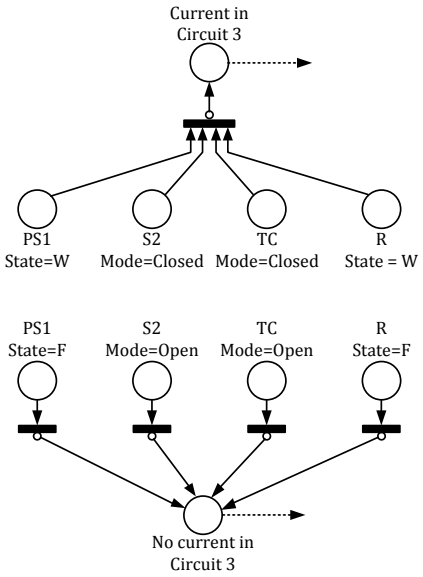

Fig. 7 Circuit Petri net for circuit list C3

4.4.3 Phase Petri nets

This phase transition table, Table 1, shows all the connections between the different phases the system can enter. The software generated automatically detects the main phases of operation from the table, and identifies them as such. The software also identifies the duration of each main phase. All other phases are then created. A transition is created for each row of the table and the switching conditions are based on the information given. The PPN controls all time aspects found within the phase transition table and component decision tables. An example of a section of the PPN can be seen in Fig. 8. The numbers on the figure show the connection to other parts of the PN model.

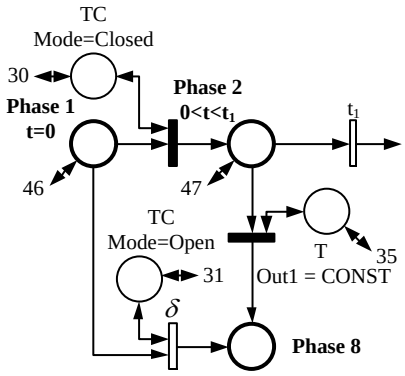

Fig. 8 Section of the phase Petri net showing the transition between phases

5 Validation

5.1 Phase fault trees

To validate the model a single four phase mission has been considered. The phase fault trees, as defined in [8], are used. Each of the four phases of the mission can be seen from Fig. 9-12. In the phase fault trees the subscript numbers identify which of the phases the component can fail in, to contribute to the failure of the current phase. It should be noted that this generally only affects components with multiple modes of operation, as the time in which the component fails, dictates the mode in which the component fails within. The phase unreliability was calculated using the minimal cut set upper bound approximation, seen in (2).

$$Q(t) = 1 - \prod_{i=1}^{N}(1 - P(C_i)) \tag{2}$$

where, $P(C_i)$ is the probability of cut set C_i occurring and N is the total number of cut sets.

Applying (2) to each of the phase fault trees in Fig. 9-12, the equations obtained can be seen in (3), (5), (7) and (9), respectively.

$$Q(Phase1) = 1 - (1 - P(S1_{OP_1})(1 - P(PS1_1)) \\ (1 - P(TC_{OP_1}))(1 - P(TIM_1)) \tag{3}$$

A common term within the phase 2 unreliability equation is represented by X and is seen in (4).

$$X = \overline{S1_{OP_1}} \cdot \overline{PS1_1} \cdot \overline{TC_{OP_1}} \cdot \overline{TIM_1} \tag{4}$$

$$Q(Phase2) = 1 - (1 - P(X \cdot R_{12}))(1 - P(X \cdot RC_{OP_1})) \\ (1 - P(X \cdot M_{12}))(1 - P(X \cdot FS_{12}))(1 - P(X \cdot PS2_{12})) \\ (1 - P(X \cdot T_{12}))(1 - P(\overline{S1_{OP_1}} \cdot \overline{PS1_1} \cdot \overline{TC_{OP_1}} \cdot TIM_2)) \\ (1 - P(X \cdot P_{12}))(1 - P(\overline{S1_{OP_1}} \cdot \overline{TC_{OP_1}} \cdot \overline{TIM_1} \cdot PS1_2)) \tag{5}$$

A common term within the phase 3 unreliability equation is represented by Y and is seen in (6).

$$Y = \overline{S1_{OP_1}} \cdot \overline{PS1_{12}} \cdot \overline{TC_{OP_1}} \cdot \overline{TIM_{12}} \cdot \overline{R_{12}}.\overline{M_{12}} \cdot \overline{FS_{12}} \cdot \overline{P_{12}} \cdot \\ \overline{T_{12}} \cdot \overline{RC_{OP_1}} \cdot \overline{PS2_{12}} \tag{6}$$

$$Q(Phase3) = 1 - (1 - P(Y \cdot V_{CL_{13}})(1 - P(Y \cdot OP_{13})) \\ (1 - P(Y \cdot PG_{L_{13}})) \tag{7}$$

A common term within the phase 4 unreliability equation is represented by Z and is seen in (8).

$$Z = \overline{S1_{OP_1}} \cdot \overline{PS1_{12}} \cdot \overline{TC_{OP_1}} \cdot \overline{TIM_{12}} \cdot \overline{R_{12}} \cdot \overline{M_{12}} \cdot \overline{FS_{12}} \cdot \overline{P_{12}} \cdot \\ \overline{T_{12}} \cdot \overline{RC_{OP_1}} \cdot \overline{PS2_{12}} \cdot \overline{V_{CL_{13}}} \cdot \overline{OP_{13}} \cdot \overline{PG_{L_{13}}} \tag{8}$$

$$Q(Phase4) = 1 - (1 - P(Z \cdot RC_{CL_{12}})(1 - P(Z \cdot S1_{CL_1})) \\ (1 - P(Z \cdot TC_{CL_{12}})) \tag{9}$$

The analytical values calculated from (3), (5), (7) and (9) can be seen in the first row of Table 17.

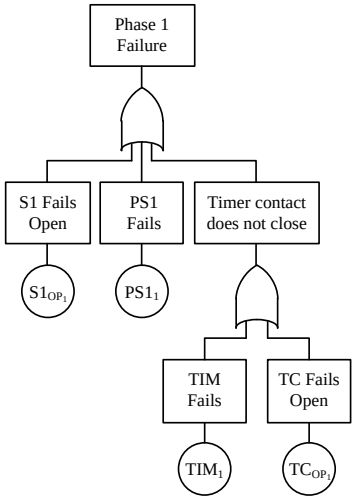

Fig. 9 Phase 1 fault tree

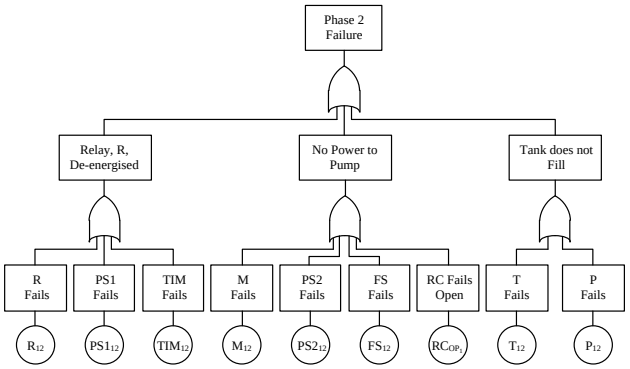

Fig. 10 Phase 2 fault tree

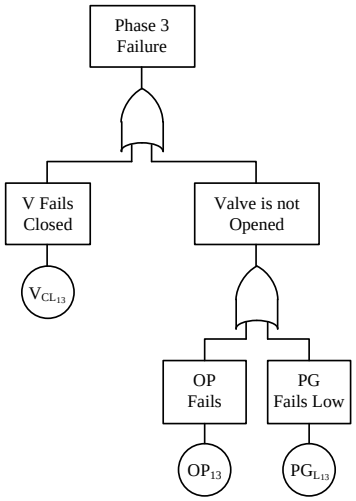

Fig. 11 Phase 3 fault tree

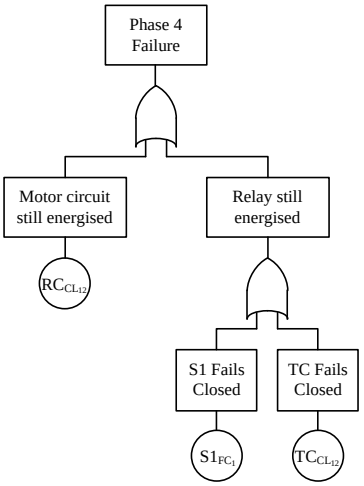

Fig. 12 Phase 4 fault tree

Table 17 Comparison of analytical and simulated data

Phase Number	1	2	3	4	Mission
Analytical	0	0.1927	0.1061	0.0670	0.3658
Simulation	0	0.1990	0.1080	0.0645	0.3715
Difference (%)	0	3.25	1.80	3.74	1.56

5.2 Convergence study

A convergence study on the simulation model was carried out for each phase of the mission and the overall mission unreliability. The first phase, a discrete phase, was completed successfully each time, i.e. no failures occurred in this phase. The second, third and fourth phase all start to converge within ± 5% of the values given in the second row of Table 17 at approximately 2100 simulations. Phases 2, 3 and 4 can be seen in Fig. 13-15, respectively. The overall mission unreliability is constant within ±5% from 1250 simulations. The convergence can be seen in Fig. 16. From this it can be seen that there is great potential in the method used to generate the reliability models.

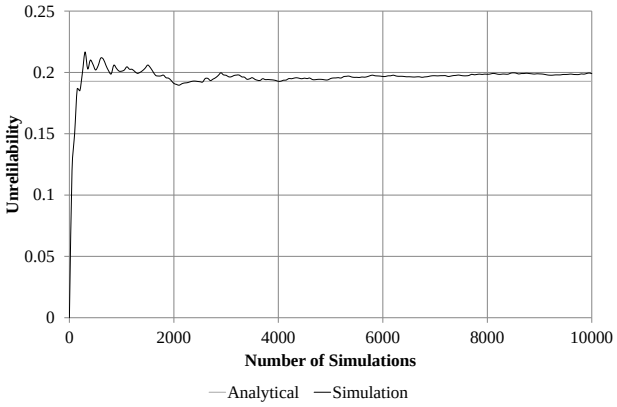

Fig. 13 Comparison of phase 2 unreliability over the 10,000 simulations and the analytical unreliability

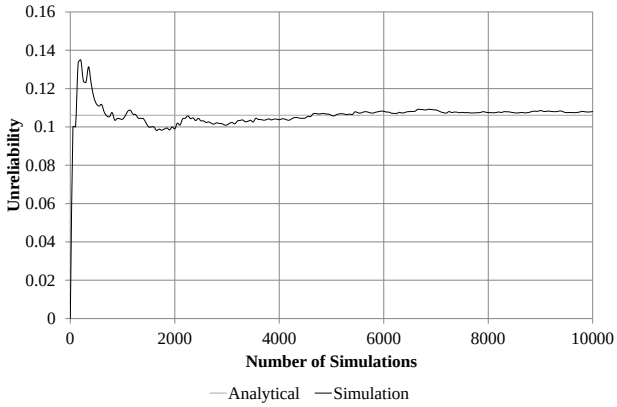

Fig. 14 Comparison of phase 3 unreliability over the 10,000 simulations and the analytical unreliability

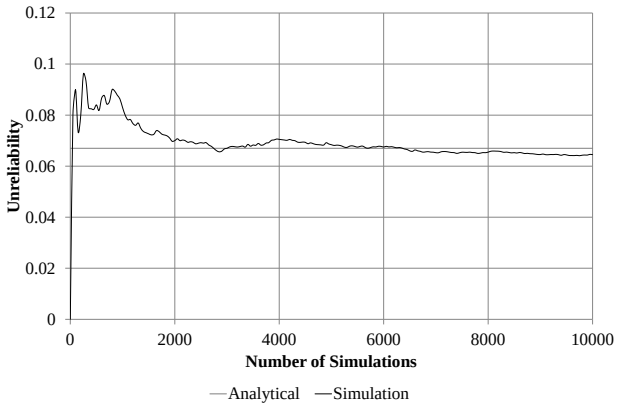

Fig. 15 Comparison of phase 4 unreliability over the 10,000 simulations and the analytical unreliability

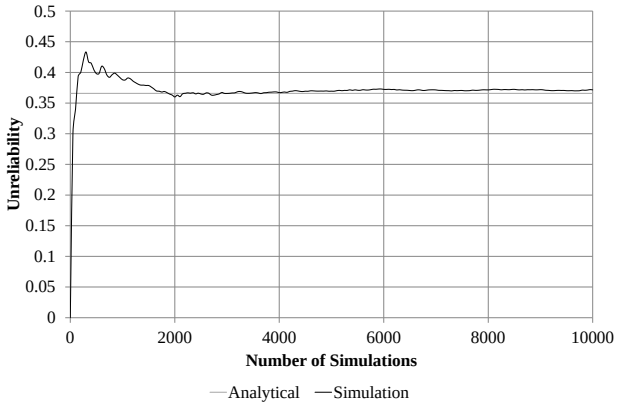

Fig. 16 Comparison of mission unreliability over the 10,000 simulations and the analytical unreliability

6 Analysis of multiple missions

The previous section studied the results for a single mission undertaken by the pressure tank system. The calculation of a single mission for this example can be carried out using the fault tree method by individuals with the correct skill set, but can take a considerable time to complete. To calculate the reliability of a system that will undertake a mission numerous times is more complex. The software has been designed to show how the system will behave over multiple missions. It is assumed that the system is continuously carrying out the mission; it is never shutdown in-between missions. For this study the failure data in Table 2 was divided by 10^3 to represent components designed to last for multiple missions. For non-repairable systems, the reliability of a system is the same as the availability of the system over a given time. In the previous study for a single mission it was seen that all phase calculations converged within the given tolerance of the analytical value at approximately 2100 simulations. Therefore, 2500 simulations were used for the study of multiple missions. The simulations carried out 5000 consecutive missions. The individual phases of the mission can be found in Fig. 17. Fig. 17 shows that the system is more likely to fail within the first two phases of the mission, rather than the later phases. Fig. 18 shows the mission unavailability of the system over the 5000 consecutive missions. As expected the data showed that as the number of missions demanded on the system increased, the likelihood that the system could carry out the required number of missions decreased. At approximately 3000 consecutive missions, the likelihood that the system would complete this number of missions was close to zero. Fig. 18, showing the mission unavailability, follows that of an exponential distribution, which would be expected of a non-repairable system.

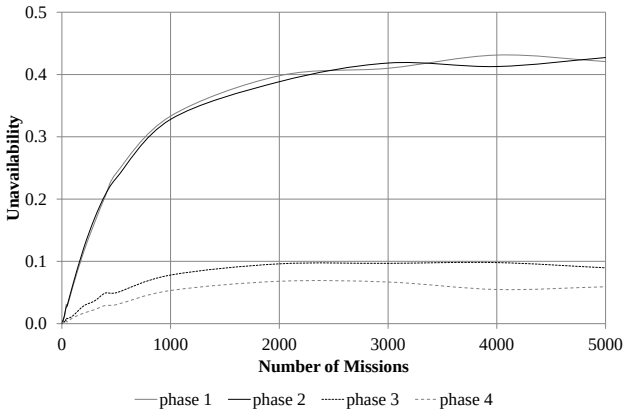

Fig. 17 Individual phase unavailability over 5000 missions

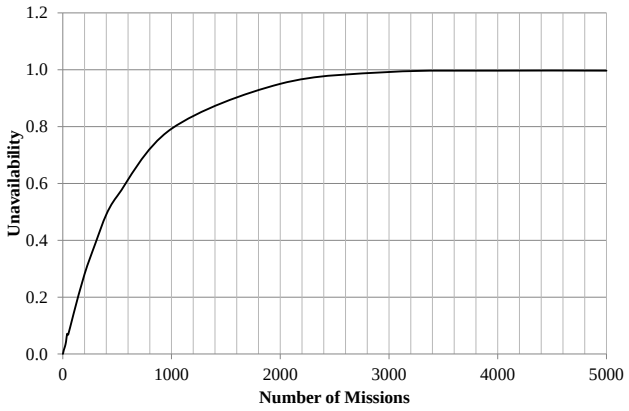

Fig. 18 Missions unavailability over 5000 missions

7 Conclusion

In this paper a procedure for automatically generating a reliability model based on Petri nets and simulation, from a description of a system and its operation has been described. The procedure has been demonstrated by applying it to a simple non-repairable example. The work presented here demonstrated that the model generation process was successful in generating the phase failure probabilities for this system considered within a given tolerance of $\pm 5\%$. The software developed can also be used to determine the availability of a non-repairable system over a given number of missions. For the non-repairable example considered it was seen that the unavailability in this case, as expected followed an exponential distribution.

The procedure is currently been applied to a more complex repairable system in order to validate its general use. Also to improve this method of automation a technique taking the system description in the form of a CAD (Computer Aided Design) diagram or a Piping and Instrumentation Diagram (P&ID) to generate the component tables is currently being investigated.

References

[1] S. A. Lapp and G. J. Powers. Computer-aided synthesis of fault trees. In *IEEE Transactions on Reliability*, vol. R-26, no. 1 pp. 2-13, 1977.

[2] S. L. Salem, G. E. Apostolakis, D. Okrent. A new methodology for the computer-aided construction of fault trees. *Annals of Nuclear Energy*, vol. 4, pp. 417-433, 1977.

[3] J. R. Taylor. An algorithm for fault tree construction. In *IEEE Transactions on Reliability*, vol. R-31, no. 2, pp. 137-146, 1982.

[4] B.B. Kelly and F.P. Lees. The propagation of faults in process plants. *Reliability Engineering & System Safety*, vol. 16, pp. 1-108, 1986.

[5] A. Majadera and T Wakabayashi. Component based-modelling of systems for automated fault tree generation. In *IEEE Transactions on Reliability*, vol. R-26, pp. 2-13, 2009.

[6] I. Mura and A Bondavalli. Markov regernarative stochastic petri nets to model and evaluate phased mission systems dependability. In *IEEE Transactions on Computers*, vol. 50, no. 12, pp. 1337-1351, 2001.

[7] S. P. Chew, S. J. Dunnett and J. D. Andrews. Phased-mission modelling of systems with maintenance-free operating periods using simulated Petri nets. *Reliability Engineering & System Safety*, vol. 93, pp. 980-994, 2008.

[8] R. La Band and J. D. Andrews. Phased mission modelling using fault tree analysis. In *Proceedings of the Institute of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*, vol. 218, pp. 83-91, 2004.

Kathryn Stockwell received her MEng in Aeronautical Engineering from Loughborough University, UK in 2009 and is currently carrying out research towards a Ph.D. degree.

Research interests include phased-mission modelling and and automatic construction of reliability models.

E-mail: k.s.stockwell@lboro.ac.uk

Sarah Dunnett received her BSc and MSc degrees in Mathematics from London University, U.K. and her PhD from Leeds University U.K. She has worked as a lecturer at Leeds University and as a senior research officer at the Health and Safety Laboratories in the UK and is currently a lecturer in the Department of Aeronautical and Automotive Engineering at Loughborough University.

Her current research interests include several aspects of system reliability modelling. Much of this work concentrates on the modelling of phase missions, and the automation of reliability techniques.

E-mail: s.j.dunnett@lboro.ac.uk (Corresponding author)